Springer Proceedings in Mathematics & Statistics

Leszek Plaskota
Henryk Woźniakowski *Editors*

Monte Carlo and Quasi-Monte Carlo Methods 2010

Springer

Springer Proceedings in Mathematics and Statistics

Volume 23

For further volumes:
http://www.springer.com/series/10533

Springer Proceedings in Mathematics and Statistics

This book series features volumes composed of selected contributions from workshops and conferences in all areas of current research in mathematics and statistics, including OR and optimization. In addition to an overall evaluation of the interest, scientific quality, and timeliness of each proposal at the hands of the publisher, individual contributions are all refereed to the high quality standards of leading journals in the field. Thus, this series provides the research community with well-edited, authoritative reports on developments in the most exciting areas of mathematical and statistical research today.

Leszek Plaskota • Henryk Woźniakowski
Editors

Monte Carlo and Quasi-Monte Carlo Methods 2010

Springer

Editors
Leszek Plaskota
University of Warsaw
Institute of Applied Mathematics
and Mechanics
Warsaw
Poland

Henryk Woźniakowski
University of Warsaw
Institute of Applied Mathematics
and Mechanics
Warsaw
Poland

and

Columbia University
Department of Computer Science
New York, NY
USA

ISSN 2194-1009 ISSN 2194-1017 (electronic)
ISBN 978-3-642-27439-8 ISBN 978-3-642-27440-4 (eBook)
DOI 10.1007/978-3-642-27440-4
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012938963

Mathematical Subject Classification (2010): Primary: 11K45, 65-06, 65C05, 65C10
Secondary: 11K38, 65D18, 65D30, 65D32, 65R20, 91B25

Printed on acid-free paper

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

Preface

We are pleased to present the refereed proceedings of the Ninth International Conference on Monte Carlo and Quasi Monte Carlo Methods in Scientific Computing (MCQMC 2010). The conference was held on the main campus of the University of Warsaw, Poland, from August 15–20 of 2010.

The program of the conference was arranged and the refereeing of papers was done with the help of an international committee consisting of: William Chen (Macquarie University, Australia), Ronald Cools (Katholieke Universiteit Leuven, Belgium), Josef Dick (University of New South Wales, Australia), Henri Faure (CNRS Marseille, France), Alan Genz (Washington State University, USA), Paul Glasserman (Columbia University, USA), Stefan Heinrich (University of Kaiserslautern, Germany), Fred J. Hickernell (Illinois Institute of Technology, USA), Stephen Joe (University of Waikato, New Zealand), Aneta Karaivanova (Bulgarian Academy of Sciences, Bulgaria), Aleksander Keller (NVIDIA Research, Berlin, Germany), Frances Y. Kuo (University of New South Wales, Australia), Pierre L'Ecuyer (University of Montreal, Canada), Christiane Lemieux (University of Waterloo, Canada), Gerhard Larcher (University of Linz, Austria), Peter Mathé (Weierstrass Institute, Berlin, Germany), Makoto Matsumoto (Hiroshima University, Japan), Thomas Müller-Gronbach (University of Passau, Germany), Harald Niederreiter (RICAM Linz and University of Salzburg, Austria), Erich Novak (University of Jena, Germany), Art. B. Owen (Stanford University, USA), Friedrich Pillichshammer (University of Linz, Austria), Leszek Plaskota (University of Warsaw, Poland), Klaus Ritter (University of Kaiserslautern, Germany), Wolfgang Ch. Schmid (University of Salzburg, Austria), Nikolai Simonov (Russian Academy of Sciences, Russia), Ian H. Sloan (University of New South Wales, Australia), Ilya M. Sobol' (Russian Academy of Sciences, Russia), Jerome Spanier (Claremont, California, USA), Shu Tezuka (Kyushu University, Japan), Xiaoqun Wang (Tsinghua University, China), Grzegorz W. Wasilkowski (University of Kentucky, USA), Henryk Woźniakowski (chair, Columbia University, USA, and University of Warsaw, Poland).

The local organizing committee of MCQMC 2010 consisted of Piotr Krzyżanowski (University of Warsaw), Marek Kwas (Warsaw School of Economics), Leszek Plaskota and Henryk Woźniakowski.

The MCQMC conferences were created by Harald Niederreiter and have become the world's major event on both Monte Carlo and Quasi-Monte Carlo methods. The meeting in Warsaw followed the successful meetings in Las Vegas, USA (1994), Salzburg, Austria (1996), Claremont, USA (1998), Hong Kong (2000), Singapore (2002), Juan-Les-Pins, France (2004), Ulm, Germany (2006), and Montreal, Canada (2008). The next MCQMC conference has already been held in Sydney, Australia, in February 2012.

The proceedings of the previous MCQMC conferences were all published by Springer-Verlag under the following titles: *Monte Carlo and Quasi-Monte Carlo Methods in 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, S. Heinrich and H. Niederreiter, eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2008* (P. L'Ecuyer and A.B. Owen, eds.).

The program of the conference in Warsaw consisted of ten 1-h plenary talks and 111 regular talks presented in 17 special sessions and 13 technical sessions. The invited speakers were: Søren Asmussen (Aarhus University, Denmark), William Chen (Macquarie University, Australia), Michael Gnewuch (Columbia University, USA), Emmanuel Gobet (Grenoble Institute of Technology, France), Stefan Heinrich (University of Kaiserslautern, Germany), Pierre L'Ecuyer (University of Montreal, Canada), Friedrich Pillichshammer (University of Linz, Austria), Gareth Roberts (University of Warwick, United Kingdom), Ian H. Sloan (University of New South Wales, Australia), Grzegorz W. Wasilkowski (University of Kentucky, USA),

The proceedings contain a limited selection of papers based on presentations given at the conference. The papers were carefully screened and they cover both the recent advances in the theory of MC and QMC methods as well as their numerous applications in different areas of computing.

We thank all the people who participated in the MCQMC conference in Warsaw and presented excellent talks, as well as all who contributed to the organization of the conference and to its proceedings. We appreciate the help of students during the conference: Piotr Gońda, Mateusz Łącki, Mateusz Obidziński, Kasia Pękalska, Jakub Pękalski, Klaudia Plaskota, Ola Plaskota and Marta Stupnicka. Our special thanks go to Piotr Krzyżanowski who, with the help of Paweł Bechler, Piotr Gońda, Piotr Kowalczyk, Mateusz Łącki and Leszek Marcinkowski, provided the invaluable support in editing the proceedings.

We gratefully acknowledge the generous financial support of the University of Warsaw, the Department of Mathematics, Informatics and Mechanics of the University of Warsaw, and the Committee of the Polish Academy of Sciences. We are especially thankful for warm hospitality of the Rector of the University of Warsaw, Professor Katarzyna Chałasińska-Macukow, and the Vice-Rector, Professor Marcin Pałys, during the conference.

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

Warsaw,
April 2012

Leszek Plaskota
Henryk Woźniakowski

Contents

Part I Invited Articles

Part II Contributed Articles

Markov Bridges, Bisection and Variance Reduction

Søren Asmussen and Asger Hobolth

Abstract Time-continuous Markov jump processes are popular modeling tools in disciplines ranging from computational finance and operations research to human genetics and genomics. The data is often sampled at discrete points in time, and it can be useful to simulate sample paths between the datapoints. In this paper we firstly consider the problem of generating sample paths from a continuous-time Markov chain conditioned on the endpoints using a new algorithm based on the idea of bisection. Secondly we study the potentials of the bisection algorithm for variance reduction. In particular, examples are presented where the methods of stratification, importance sampling and quasi Monte Carlo are investigated.

1 Introduction

Let $X = \{X(t) : t \geq 0\}$ be a Markov process in continuous time with discrete or general state space E. A *Markov bridge* with parameters T, a, b is then a stochastic process with time parameter $t \in [0, T]$ and having the distribution of $\{X(t) : 0 \leq t \leq T\}$ conditioned on $X(0) = a$ and $X(T) = b$.

Markov bridges occur in a number of disciplines ranging from computational finance and operations research to human genetics and genomics. In many applications, it is of relevance to simulate sample paths of such bridges. In particular, the case of diffusions has received extensive attention. An early reference is [31]

S. Asmussen
Department of Mathematical Sciences, Aarhus University, Aarhus, Denmark
e-mail: asmus@imf.au.dk

A. Hobolth
Bioinformatics Research Center, Aarhus University, Arhus, Denmark
e-mail: asger@birc.au.dk

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_1, © Springer-Verlag Berlin Heidelberg 2012

and later selected ones [5–7], and [9]. Also some special Lévy processes have been considered, see [4, 24, 29], and [30]. A more theoretical discussion of Markov bridges can be found in [18].

The present paper is concerned with the CTMC (continuous time Markov chain) case where the state space E of X is finite. This case is of course much simpler than diffusions or Lévy processes, but nevertheless, it occurs in some important applications. With E finite, there is a simple description of the process in terms of the rate (intensity) matrix $Q = (q_{ij})_{i,j\in E}$: a jump $i \to j \neq i$ occurs at rate q_{ij}. Equivalently, state i has an exponential holding time with mean $1/q_i$ where $q_i = -q_{ii} = \sum_{j\neq i} q_{ij}$, and upon exit, the new state equals $j \neq i$ with probability $\theta_{ij} = q_{ij}/q_i$. Cf. [1, Chap. 2]. Note that the assumption of recurrence needs not be imposed (i.e. absorbing states are allowed).

We focus here on a bisection algorithm first presented in [3]. The details are surveyed in Sect. 4, but the key is two fundamental observations. Firstly, if the endpoints are the same and the process does not experience any jumps, the sample path generation is finished. Secondly, if the endpoints are different and the process experiences exactly one jump, sample path generation is easy; we must basically simulate a waiting time before the jump from a truncated exponential distribution. These two fundamental observations are described in more detail in Sect. 4.1, but once they are in place they immediately suggest a recursive procedure for sample path generation: continue splitting the large time interval into smaller time intervals, and keep splitting until all intervals contain either no jumps or one jump only.

Previous algorithms for bridge sampling from CTMCs include *rejection sampling*, *uniformization* and *direct simulation* and are briefly surveyed in Sect. 3 (also Markov chain Monte Carlo methods have been used, e.g. [31] and [6], but we do not discuss this aspect here). Reference [20] compares these algorithms, and a comparison with bisection can be found in [3]. The overall picture is that no algorithm is universally superior in terms of fast generation of sample paths. In particular, we do not insist that the bisection idea is a major jump forward in this respect. Rather, it is our intention to advocate the use of bisection for variance reduction by looking for some 'most important' random variables on which to concentrate variance reduction ideas. We implement this in examples, and in addition, we give a detailed description of the bisection algorithm and a survey of alternative methods.

The idea of using bisection for variance reduction is familiar from the Brownian bridge, see for example [10, 11, 27] and [2, p. 277–280]. Here the 'most important' r.v.'s are first $X(0)$, $X(T)$, next $X(T/2)$, then $X(T/4)$, $X(3T/4)$ and so on. However, in our implementation of CTMC bridges a new aspect occurs since also the number of jumps in $[0, T]$, $[0, T/2]$, $[T/2, T]$ and so on play a role. The variance reduction techniques we study are stratification, importance sampling and quasi Monte Carlo.

2 Examples

2.1 Statistical Inference in Finite State CTMC Models

For statistical purposes, the relevant parameters to estimate are often the elements q_{ij} of the rate matrix Q, or, equivalently, the q_i and the $\theta_{ij} = q_{ij}/q_i$, $j \neq i$. In the case of complete observations in $[0, T]$ (that is, the whole trajectory $\{X(t) : 0 \leq t \leq T\}$ is observed), there is a simple solution: the maximum likelihood estimators $\widehat{q_i}$, $\widehat{\theta}_{ij}$ of q_i and $\theta_{ij} = q_{ij}/q_i$ are just the empirical counterparts. That is, the sufficient statistics are

$$\begin{aligned}
T_i &= \text{time in state } i \;=\; \int_0^T I\big(X(t) = i\big)\,\mathrm{d}t\,,\\
N_{ij} &= \#\,(\text{jumps } i \to j) \;=\; \sum_{0 \leq t \leq T} I\big(X(t-) = i,\, X(t) = j\big)\,,\\
N_i &= \sum_{j \neq i} N_{ij}\,,
\end{aligned}$$

and the maximum likelihood estimators are

$$\widehat{q_i} \;=\; \frac{N_i}{T_i}\,, \quad \widehat{\theta}_{ij} \;=\; \frac{N_{ij}}{N_i}\,. \tag{1}$$

In many applications of continuous time Markov chains, the stochastic process $\{X(t) : t \geq 0\}$ is, however, sampled at equidistant discrete points

$$t_0 = 0 < t_1 = h < t_2 = 2h < \cdots < t_{n-1} = (n-1)h < t_n = nh = T$$

in time, while the process itself is a continuous-time process. This situation is a missing data problem, for which the EM (Expectation-Maximization) algorithm is a classical tool. This algorithm is iterative, i.e. in step k it has a trial $q_i^{(k)}$, $\theta_{ij}^{(k)}$ for the parameters. To update to $k+1$, one then in (1) replaces the sufficient statistics by their conditional expectation with parameters $q_i^{(k)}$, $\theta_{ij}^{(k)}$ given the data $X(0), X(h), \ldots, X((n-1)h), X(T)$. That is,

$$\widehat{q}_i^{(k+1)} \;=\; \frac{N_i^{(k)}}{T_i^{(k)}}\,, \quad \widehat{\theta}_{ij}^{(k+1)} \;=\; \frac{N_{ij}^{(k)}}{N_i^{(k)}}\,, \tag{2}$$

where

$$\begin{aligned}
T_i^{(k)} &= \mathbb{E}_{q_i^{(k)},\theta_{ij}^{(k)}}\Big[\int_0^T I\big(X(t)=i\big)\,\mathrm{d}t \,\Big|\, X(0),X(h),\dots,X((n-1)h),X(T)\Big],\\
N_{ij}^{(k)} &= \mathbb{E}_{q_i^{(k)},\theta_{ij}^{(k)}}\Big[\sum_{0\le t\le T} I\big(X(t-)=i,\,X(t)=j\big)\,\Big|\\
&\qquad \times X(0),X(h),\dots,X((n-1)h),X(T)\Big],\\
N_i^{(k)} &= \sum_{j\neq i} N_{ij}^{(k)}\,.
\end{aligned}$$

The computation of these conditional expectations is the *E-step* of the algorithm, whereas (2) is the *M-step*. The E-step is the computationally demanding one. As a consequence of the Markov assumption, knowledge of the data partitions the problem into $n = T/h$ independent problems. For example,

$$T_i^{(k)} = \sum_{m=1}^{n} \mathbb{E}_{q_i^{(k)},\theta_{ij}^{(k)}}\Big[\int_{(m-1)h}^{mh} I\big(X(t)=i\big)\,\mathrm{d}t \,\Big|\, X\big((m-1)h\big),X(mh)\Big].$$

The computations are in principle feasible via deterministic numerical analysis, but the implementation is somewhat tedious, so it is popular to use simulation instead. Then independent sample paths $\{X(t) : (m-1)h \le t < mh\}$ must be generated between the timepoints $(m-1)h$ and mh, conditional on the datapoints $X\big((m-1)h\big)$ and $X(mh)$. This is how the problem of simulating Markov bridges arises in the statistical context.

For more information on rate matrix estimation in partially observed finite state CTMC models we refer to [25] and references therein.

2.2 Applications in Genetics

A DNA string is a word from the alphabet A, G, C, T. When observing two closely related species like e.g. human and mouse, letters are equal at most sites (more than 80%; see [13]), but differ at a few as in Fig. 1 where the two strings are identical except at the third site. The lines in the figure are ancestral lineages back to the common ancestor. At each site, mutations occur, changing for example an A to a G. One is often interested in the (unobservable) complete history along the ancestral lines.

For a fixed single site, the common model assumes Markovian mutations at known exponential holding rates $q_{\mathtt{A}}, q_{\mathtt{C}}, q_{\mathtt{G}}, q_{\mathtt{T}}$ and known transition probabilities (e.g. $\theta_{\mathtt{AG}} = q_{\mathtt{AG}}/q_{\mathtt{A}}$ for A→G). One further assumes time reversibility and that the ancestral lines are so long that stationarity has been reached. One can then reverse

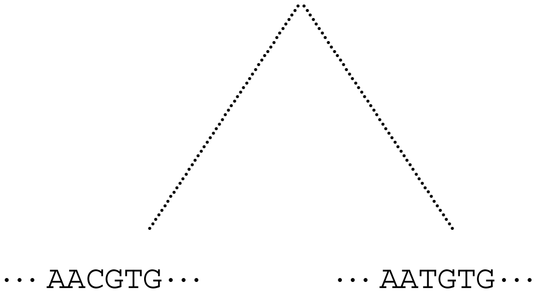


Fig. 1 Related sites of DNA from human and mouse are identical at most positions. The possible states are from the DNA alphabet {A, G, C, T }.

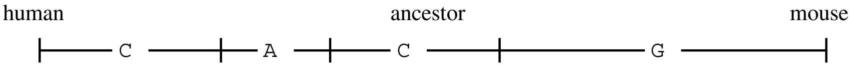


Fig. 2 Example of evolution from human to mouse at a specific position. Time is reversed at the human lineage when compared to the previous figure.

time along one of the ancestral lines, say the one starting from the human, to get a Markov process running from e.g. human to mouse and having known endpoints, see Fig. 2.

An early popular model is that of [23] where the Q-matrix takes the form

$$\begin{array}{c|cccc} & \mathtt{A} & \mathtt{G} & \mathtt{C} & \mathtt{T} \\ \hline \mathtt{A} & -\alpha-2\beta & \alpha & \beta & \beta \\ \mathtt{G} & \alpha & -\alpha-2\beta & \beta & \beta \\ \mathtt{C} & \beta & \beta & -\alpha-2\beta & \alpha \\ \mathtt{T} & \beta & \beta & \alpha & -\alpha-2\beta \end{array}$$

One readily computes the stationary distribution $\pi = (1, 1, 1, 1)/4$ and checks the conditions of detailed balance ($\pi_{\mathtt{G}} q_{\mathtt{GT}} = \pi_{\mathtt{T}} q_{\mathtt{TG}}$ etc.) so that the model is indeed time reversible. Plugging in specific values of α, β and the time horizon T, one can then simulate the Markov bridge from human to mouse to obtain information on the ancestral history. One possible application is to put a prior on the length $T/2$ of the ancestral lines and use the simulations to compute a posterior.

Endpoint conditioned CTMC's are thus a crucial modelling tool for the evolution of DNA sequences. At the nucleotide level the states for the DNA substitution process state space is 4 as described above. At the amino acid level the state space size is 20 and at the codon level the size is 61. The ancestry is usually represented by

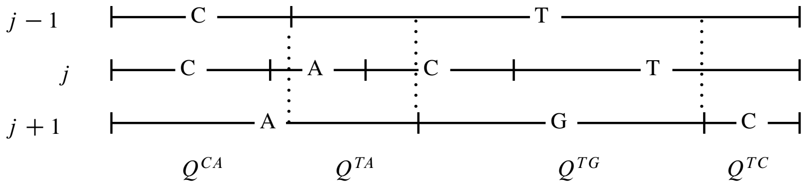


Fig. 3 Illustration of neighbour dependence. The Q-matrix at site j depends on the states at the neighbouring sites. In the figure the neighbouring states at site $(j-1, j+1)$ are $(\mathtt{C}, \mathtt{A})$; $(\mathtt{T}, \mathtt{A})$; $(\mathtt{T}, \mathtt{G})$ and $(\mathtt{T}, \mathtt{C})$. When simulating site j conditioned on the endpoints one must take the states of the neighbouring sites into account.

a binary tree where a node corresponds to two DNA sequences finding a common ancestor.

For a given set of DNA sequences observed at the leaves of a given tree we can determine the probabilities of the states in the inner nodes using Felsenstein's tree peeling algorithm [16], and therefore the basic setup is very similar to an endpoint conditioned CTMC. For more information on the use of CTMC methodology in evolutionary models of DNA sequences we refer to [14, 17, Chaps. 13 and 14] and references therein. Extremely large comparative genomic data sets consisting of hundreds of sequences of length thousands of sites are currently being generated. Interestingly, in order to analyze such large trees, approximative methods based on bisection and uniformization techniques are being devoloped; see [12] for more information on this line of research.

Single site analysis is not completely satisfactory because there is dependence among neighboring sites. A simple and popular model assumes that the Q-matrix at site j only depends on the states at sites $j-1$ and $j+1$ as in Fig. 3.

Simulation of multiple sites can then be performed by Gibbs sampling, where one site at a time is updated. For updating of site j, one first simulates X at change points, i.e. times of state change of either site $j-1$ or $j+1$. These values form an inhomogeneous end-point conditioned discrete time Markov chain with easily computed transition probabilities. Once they are known, the evolution between change points are Markov bridges. See [19, 21] and [22] for more information on neighbour-dependent substitution models in molecular evolution.

3 Previous Algorithms

Reference [20] describes and analyses 3 previously suggested algorithms for end-point conditional simulation from continuous time Markov chains. The algorithms are called *rejection sampling, uniformization* and *direct simulation*. We will only briefly describe the algorithms here. For a detailed description of the algorithms we refer to [20].

Recall that our aim is to simulate a sample path $\{X(t) : 0 \leq t \leq T\}$ from a continuous time Markov chain conditional on the end-points $X(0) = a$ and $X(T) = b$. In *rejection sampling*, a sample path is simulated forward in time from $X(T) = a$, and the path is accepted if $X(T) = b$. Reference [28] describes an improvement of the naive rejection sampling approach where it is taken into account that if $a \neq b$, at least one jump must occur. Nielsens improvement is particularly important when the time interval is short and the beginning and ending states are different. Rejection sampling is inefficient if the acceptance probability is low, i.e. if it is unlikely for the forward simulated Markov chain to end up in the desired ending state.

In *uniformization* (e.g. [15]), the number of state changes within an interval is Poisson distributed. The state changes themselves constitute a Markov chain. The price for the simple description of the number of state transitions is that virtual state changes (in which the state does not change) are permitted. Sampling from this related process is equivalent to sampling from the target continuous time Markov chain when the virtual changes are ignored. When simulating an endpoint conditioned sample path using uniformization, the number of state transitions is firstly simulated. This number follows a slightly modified Poisson distribution (the modification comes from the conditioning on the endpoints). When the number of jumps is simulated, the Markovian structure of the state transitions is utilized to simulate the types of changes that occur. Uniformization is usually very efficient, but can be slow if many virtual state changes are needed in the simulation procedure.

Finally, *direct simulation* [19] is based on analytical expressions for simulating the next state and the waiting time before the state change occurs. The expression for the waiting time distribution and corresponding cumulative distribution function are analytically available, but unfortunately not very tractable. Therefore the recursive steps of simulating the new state and corresponding waiting time is a rather time-consuming process.

All three algorithms for simulating an end-point conditioned CTMC can be divided into a (1) initialization, (2) recursion and (3) termination step. Denoting the computational cost for initialization α and a single step in the recursion β, the full cost of each of the three algorithms can, for moderately large T, be well approximated by

Rejection sampling	Direct sampling	Uniformization
$(\alpha_R + \beta_R T)\,/\,p_{acc}$	$\alpha_D + \beta_D T$	$\alpha_U + \beta_U T\mu.$

In these formulas Q is scaled such that one jump is expected per time unit $(\sum_c \pi_c Q_c = 1)$, p_{acc} is the probability of accepting a forward sample, and $\mu = \max_c q_c$ is the rate of state changes (including the virtual) in the uniformized process. The computational costs α and β depend on the size of the rate matrix and the software.

4 Bisection Algorithm

The algorithm involves an initialization step and a recursive step. The recursive step is easy once the initialization step is explained. We divide the discussion of the initialization into two parts. In the first part, the end-points are the same, and in the second part the end-points are different.

4.1 The Basic Idea

The bisection algorithm is based on two fundamental observations:

1. If $X(0) = X(T) = a$ and there are no jumps we are done: $X(t) = a,\ 0 \le t \le T$.
2. If $X(0) = a$ and $X(T) = b \neq a$ and there is precisely one jump we are basically done: $X(t) = a,\ 0 \le t < \tau$, and $X(t) = b,\ \tau \le t \le T$.

In 2, the jump time τ is determined by Lemma 3 and Remark 1 below, which show that intervals with precisely one jump are easy to handle.

The basic idea of the bisection algorithm is to formulate a recursive procedure where we finish off intervals with zero or one jumps according to the two fundamental observations above, and keep bisecting intervals with two or more jumps. The recursion ends when no intervals with two or more jumps are present.

We recall the notation $Q = \{q_{ab}\}$ for the instantaneous rate matrix with off-diagonal entries $q_{ab} \ge 0$ and diagonal entries $q_{aa} = -\sum_{b\neq a} q_{ab} = -q_a < 0$. We make the assumption that the process is irreducible, i.e. it is possible to get from any state to any state in the jump chain. The algorithm (as well as uniformization and direct simulation, cf. Sect. 3) require the transition probabilities $P_{ab}(t)$, i.e. the elements of the transition matrix $P(t) = \mathrm{e}^{Qt}$. These can easily be computed, for example, if Q can be written in diagonal form UDU^{-1} with $D = \mathrm{diag}(\lambda_i)$; then $P(t) = U\mathrm{diag}(\mathrm{e}^{\lambda_i t})U^{-1}$. For different methods, see the classical paper by Moler and van Loan [26].

Lemma 1. *Consider an interval of length T with $X(0) = a$, and let $b \neq a$. The probability that $X(T) = b$ and there is only one single jump (necessarily from a to b) in the interval is given by*

$$R_{ab}(T) = q_{ab} \begin{cases} \dfrac{\mathrm{e}^{-q_a T} - \mathrm{e}^{-q_b T}}{q_b - q_a} & q_a \neq q_b \\ T\mathrm{e}^{-q_a T} & q_a = q_b. \end{cases} \tag{3}$$

The density of the time of state change is

$$f_{ab}(t;T) = \frac{q_{ab}\mathrm{e}^{-q_b T}}{R_{ab}(T)}\mathrm{e}^{-(q_a - q_b)t}, \quad 0 \le t \le T.$$

Furthermore, the probability that $X(T) = b$ and there are at least two jumps in the interval is $P_{ab}(T) - R_{ab}(T)$.

Proof. Let $N(T)$ denote the number of jumps in the interval $[0, T]$. The first two parts of the Lemma follow from

$$R_{ab}(T) = \mathbb{P}\big(X(T) = b, N(T) = 1 \,\big|\, X(0) = a\big)$$
$$= \int_0^T q_a \mathrm{e}^{-q_a t} \frac{q_{ab}}{q_a} \mathrm{e}^{-q_b(T-t)}\,\mathrm{d}t \;=\; q_{ab}\mathrm{e}^{-q_b T} \int_0^T \mathrm{e}^{-(q_a - q_b)t}\,\mathrm{d}t, \quad a \neq b.$$

The last part is clear since the case of zero jumps is excluded by $a \neq b$. □

Remark 1. If $q_a > q_b$, the time of the state change is an exponentially distributed random variable with rate $q_a - q_b$ truncated to $[0, T]$. Such a random variable V is easily simulated by inversion (e.g. [2, p. 39]). If $q_a < q_b$, we have by symmetry that $f_{ab}(t)$ is the density of the random variable $T - V$, where V is an exponentially distributed random variable with rate $q_b - q_a$ truncated to $[0, T]$. Finally, if $q_a = q_b$, the time of the state change is simply uniform on $[0, T]$. □

4.2 Initialization When the Endpoints Are Equal

Consider the case $X(0) = X(T) = a$. We may write

$$P_{aa}(T) = P_{aa}(T/2)P_{aa}(T/2) + \sum_{c \neq a} P_{ac}(T/2)P_{ca}(T/2). \tag{4}$$

Here $P_{aa}(T/2)$ can be further dissected into

$$\begin{aligned} P_{aa}(T/2) &= \mathbb{P}(X(T/2) = a | X(0) = a) \\ &= \mathbb{P}(X(T/2) = a, N(T/2) = 0 | X(0) = a) \\ &\quad + \mathbb{P}(X(T/2) = a, N(T/2) \geq 2 | X(0) = a) \\ &= \mathrm{e}^{-q_a T/2} + [P_{aa}(T/2) - \mathrm{e}^{-q_a T/2}], \end{aligned} \tag{5}$$

and similarly $P_{ac}(T/2)$ can be written as

$$P_{ac}(T/2) = R_{ac}(T/2) + [P_{ac}(T/2) - R_{ac}(T/2)]. \tag{6}$$

With the abbreviation $e_a = \mathrm{e}^{-q_a T/2}$, $r_{ab} = R_{ab}(T/2)$, $p_{ab} = P_{ab}(T/2)$ we obtain Table 1 when substituting (5) and (6) into (4).

Note that in case 1–4 we have $X(T/2) = a$, and in case 5–8 we have $X(T/2) = c \neq a$. Of course we have

Table 1 Possible scenarios when the endpoints $X(0) = a$ and $X(T) = a$ are the same.

Case	Number of jumps in first interval	Number of jumps in second interval	(Unconditional) Probability	Notation
1	0	0	$e_a e_a$	α_1
2	0	≥ 2	$e_a(p_{aa} - e_a)$	α_2
3	≥ 2	0	$(p_{aa} - e_a)e_a$	α_3
4	≥ 2	≥ 2	$(p_{aa} - e_a)(p_{aa} - e_a)$	α_4
5	1	1	$r_{ac} r_{ca}$	$\alpha_{5,c}$
6	1	≥ 2	$r_{ac}(p_{ca} - r_{ca})$	$\alpha_{6,c}$
7	≥ 2	1	$(p_{ac} - r_{ac})r_{ca}$	$\alpha_{7,c}$
8	≥ 2	≥ 2	$(p_{ac} - r_{ac})(p_{ca} - r_{ca})$	$\alpha_{8,c}$

$$P_{aa}(T) = \sum_{i=1}^{4} \alpha_i + \sum_{i=5}^{8} \sum_{c \neq a} \alpha_{i,c}.$$

In the initialization step, we select one of the cases with probabilities proportional to the corresponding α-value. In case the algorithm enters case 1 we are done. In case the algorithm enters case 5 we are almost done; we just need to simulate two waiting times according to Remark 1: one waiting time in the interval $[0, T/2]$ with beginning state a and ending state c, and another in the interval $[T/2, T]$ with beginning state c and ending state a.

In case the algorithm enters one or more intervals where the number of jumps are ≥ 2, further simulation is needed (case 2, 3, 4, 6, 7, 8), and we move on to the recursion step explained below. However, we only need to pass intervals to the next level of the algorithm if the number of jumps are larger or equal to two. If the selected case is case 2, for example, we only need to pass the second interval $[T/2, T]$ and the endpoints $X(T/2) = a$ and $X(T) = a$. Similarly, if the selected case is case 6 we use Remark 1 to simulate the waiting time to state c in the first interval (and keep the type and time of the state change in the memory), but we only pass on the second interval $[T/2, T]$ and the endpoints $X(T/2) = c$ and $X(T) = a$ to the next level.

4.3 Initialization When the Endpoints Are Different

Now consider the case when the end-points $X(0) = a$ and $X(T) = b \neq a$ are different. This time we get

$$P_{ab}(T) = P_{aa}(T/2)P_{ab}(T/2) + P_{ab}(T/2)P_{bb}(T/2) + \sum_{c \notin \{a,b\}} P_{ac}(T/2)P_{cb}(T/2).$$

Using the same notation as previously, we get the 12 cases in Table 2.

Table 2 Possible scenarios when the endpoints $X(0) = a$ and $X(T) = b \neq a$ are different.

Case	Number of jumps in first interval	Number of jumps in second interval	(Unconditional) Probability	Notation
1	0	1	$e_a r_{ab}$	β_1
2	0	≥ 2	$e_a(p_{ab} - r_{ab})$	β_2
3	≥ 2	1	$(p_{aa} - e_a)r_{ab}$	β_3
4	≥ 2	≥ 2	$(p_{aa} - e_a)(p_{ab} - r_{ab})$	β_4
5	1	0	$r_{ab}e_b$	β_5
6	1	≥ 2	$r_{ab}(p_{bb} - e_b)$	β_6
7	≥ 2	0	$(p_{ab} - r_{ab})e_b$	β_7
8	≥ 2	≥ 2	$(p_{ab} - r_{ab})(p_{bb} - e_b)$	β_8
9	1	1	$r_{ac}r_{cb}$	$\beta_{9,c}$
10	1	≥ 2	$r_{ac}(p_{cb} - r_{cb})$	$\beta_{10,c}$
11	≥ 2	1	$(p_{ac} - r_{ac})r_{cb}$	$\beta_{11,c}$
12	≥ 2	≥ 2	$(p_{ac} - r_{ac})(p_{cb} - r_{cb})$	$\beta_{12,c}$

Note that we can merge case 1 and case 5 (corresponding to one jump):

$$e_a r_{ab} + r_{ab} e_b = R_{ab}(T).$$

It clearly holds that

$$P_{ab}(T) = \sum_{i=1}^{8} \beta_i + \sum_{i=9}^{12} \sum_{c \neq (a,b)} \beta_{i,c}.$$

In case 1–4 we have $X(T/2) = a$, in case 5–8 we have $X(T/2) = b \neq a$, and in case 9–12 we have $X(T/2) = c \notin \{a, b\}$.

In the initialization step, we select one of the cases with probabilities proportional to the corresponding β-value. If the algorithm enters one or more intervals where the number of jumps are larger than two, further simulation is needed (case 2,3,4,6,7,8,10,11,12). If the algorithm enters a case where the number of jumps is at most one in both intervals (case 1,5,9), then the construction of the path on the current subinterval is finished.

Entering an interval with $\geq$ 2 jumps means that further simulation is needed. In the next subsection, we discuss this recursive part of the bisection algorithm.

4.4 Recursion and Termination

When an interval with $\geq$ 2 jumps is entered, further simulation is needed. However, it is straightforward to calculate the probabilities for the various scenarios; the (unconditional) probabilities are given by Table 1 with case 1 removed if the endpoints of the interval are the same, and by Table 2 with case 1 and 5 removed if the end-points of the interval are different. (The values occurring in Tables 1 and 2 should also be calculated for half as long a time interval.) The algorithm terminates when no intervals with ≥ 2 jumps are present.

5 Numerical Examples

We now present a collection of results from three experiments where bisection ideas are used for variance reduction in time-continuous Markov jump processes. The three experiments are (1) Stratification, (2) Importance sampling and (3) Quasi Monte Carlo. We consider a $N \times N$ rate matrix Q where the rates are given by $q_{1,2} = \lambda$, $q_{n,n+1} = \mu$, $n = 2, \ldots, N-1$, $q_{N,1} = \mu$, and all other rates are zero, cf. Fig. 4. We let $\mu = N$.

Our target is to determine the probability p_λ of exactly one cycle in the time interval [0, 1] conditioned on the initial state $X(0) = 1$ and final state $X(1) = 1$. We stress that neither the model nor the problem of estimating p_λ are chosen because of their intrinsic interest but in order to investigate the potential of the bisection algorithm for variance reduction in a simple example. Indeed, the value of p_λ is the probability of making exactly N jumps conditional on the end-points $X(0) = X(1) = 1$. This probability can be calculated using the algorithm of [32] which for convenience is reproduced in the Appendix. In Table 3 we provide the exact probabilities of p_λ in our experiments.

One possibility of estimating p_λ is of course the crude Monte Carlo method, to just generate sample paths $\{X(t) : 0 \leq t \leq 1\}$ conditional on $X(0) = 1$ and $X(1) = 1$ (we have previously discussed several algorithms, including bisection, for obtaining such samples). Let Z_r, $r = 1, \ldots, R$ indicate if exactly one cycle

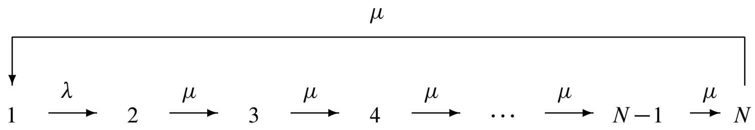


Fig. 4 Transition diagram for the cyclic example.

Table 3 Exact probability p_λ for one cycle for various state space sizes N and various ratios λ/N.

	λ/N						
N	0.10	0.45	0.80	1.00	1.20	3.10	5.00
4	0.138405	0.567199	0.766990	0.800363	0.803456	0.639064	0.562409
7	0.191982	0.818845	0.946474	0.948949	0.941157	0.857991	0.818518
10	0.253733	0.940944	0.987026	0.984950	0.981193	0.950914	0.935423
15	0.373870	0.992560	0.998395	0.997843	0.997230	0.992588	0.990111
20	0.506338	0.999121	0.999775	0.999687	0.999597	0.998920	0.998555
30	0.745579	0.999988	0.999995	0.999993	0.999992	0.999977	0.999970

is obtained in the rth sample path where R is the number of replications. Clearly $Z_r \sim \text{Bin}(1, p_\lambda)$ and so the crude Monte Carlo estimator is $\bar{Z} = \sum_{r=1}^{R} Z_r/R$ with variance

$$\sigma^2 = \text{Var}(\bar{Z}) = \frac{p_\lambda(1-p_\lambda)}{R}.$$

5.1 Stratification

First consider a proportional stratification procedure [2, V.7] where the R replicates are allocated according to the probability of the midpoint s, $s = 1, \dots, N$ of the process. More precisely, let $p_s = \mathbb{P}(X(1/2) = s | X(0) = 1, X(1) = 1)$ be the probability of the midpoint being s and allocate $R_s = Rp_s$ (or rather the rounded value) replicates to this stratum. We use $\sum_{s=1}^{N} p_s \bar{Z}_s$ as an estimator of p_λ, where $\bar{Z}_s = \sum_{i=1}^{R_s} Z_{s,i}/R_s$ and $Z_{s,i}$ indicates if the ith sample in the sth stratum contains exactly one cycle.

Letting

$$p_{\lambda,s} = \mathbb{P}\big(\text{exactly one cycle} \,\big|\, X(0) = 1, X(1/2) = s, X(1) = 0\big),$$

we obtain the stratum variance

$$\sigma^2_{\text{Str}} = \sum_{s=1}^{N} p_s^2 \frac{p_{\lambda,s}(1-p_{\lambda,s})}{R_s}.$$

We now see that the ratio between the two variances is given by

$$\frac{\sigma^2_{\text{Str}}}{\sigma^2} = \sum_{s=1}^{N} p_s \frac{p_{\lambda,s}(1-p_{\lambda,s})}{p_\lambda(1-p_\lambda)}, \tag{7}$$

where we have used $R_s = Rp_s$.

In Fig. 5 left we show (using exact calculations) the values of the ratios between the two variances for several values of λ and size of state space N. We see that when $\lambda \ll N$ we obtain a major reduction in the variance when stratification is applied. In the cases $\lambda \sim N$ and $\lambda \gg N$, a variance reduction is mainly obtained for large state spaces.

Instead of only stratifying according to the midpoint, we can include information about the number of jumps according to Table 1. We thus include not only the midpoint but also if 0,1 or at least 2 jumps are present. We again apply a proportional stratification procedure. The variance ratio between the two stratification procedures are shown in Fig. 5 right. We see that a major variance reduction is obtained for small values of λ and small state spaces.

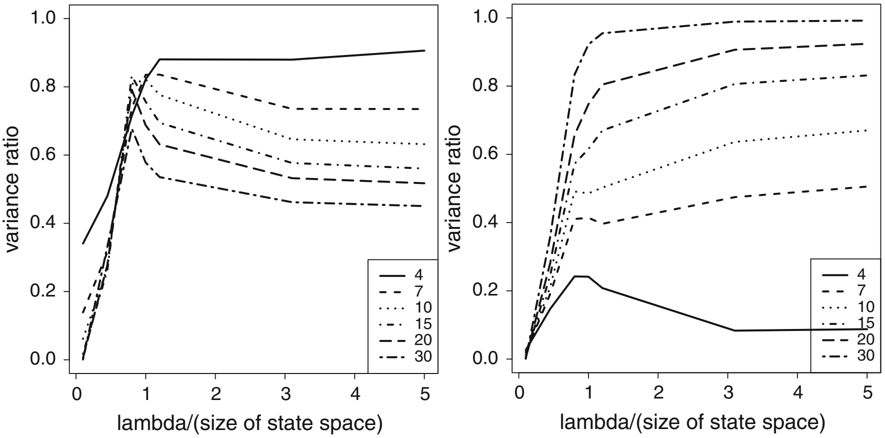


Fig. 5 Variance reduction using stratification strategies. *Left*: Variance ratio (7) between naive sampling and stratification according to midpoint. The variance ratio is shown as a function of λ/N. The variance reduction is large when λ is small and otherwise the variance reduction is moderate. *Right*: Variance ratio between stratification according to midpoint and stratification according to midpoint *and* information on number of jumps (0,1 or $\geq$ 2). Again the variance reduction is large when λ is small.

5.2 *Importance Sampling*

Another variance reduction mechanism is importance sampling. We choose to do importance sampling on the midpoint and include information that (a) the chain can only jump from n to $(n + 1)$ (modulo N) and (b) one cycle corresponds to exactly N jumps. Having sampled the midpoint and number of jumps in the two intervals (0,1 or $\geq$ 2), we proceed according to the bisection algorithm. In our proposal mechanism, the N jumps are distributed in the two intervals according to a multinomial distribution with probability vector $(1/2, 1/2)$ and number of trials N, i.e. the number of jumps in the first interval follows a binomial distribution $\text{Bin}(N, 1/2)$ with parameter $1/2$ and N trials. We have outlined the proposal mechanism in Table 4 (compare to Table 1). The importance sampling weight is the ratio between the bisection probability (the true distribution) and the importance sampling probability.

In Fig. 6 we show the ratio between the importance sampling variance (the variance of the importance weights) and the 'naive' sampling scheme. Even though the importance sampling scheme takes information about the type of CTMC into account it appears that we only obtain a variance reduction when the state space is smaller than 15.

Table 4 Possible number of jumps in the two intervals in the importance sampling scheme. In case 8, the last case, the value of the number of jumps k is between 2 and $N-2$. The importance sampling weight is the ratio between the bisection probability and sampling probability.

Case	Number of jumps in first interval	Number of jumps in second interval	Sampling probability	Bisection probability
1	0	0	0	Irrelevant
2	0	≥ 2	Bin(0; N, 1/2)	$\alpha_2/P_{aa}(T)$
3	≥ 2	0	Bin(N; N, 1/2)	$\alpha_3/P_{aa}(T)$
4	≥ 2	≥ 2	0	Irrelevant
5	1	1	0	Irrelevant
6	1	≥ 2	Bin(1; N, 1/2)	$\alpha_6/P_{aa}(T)$
7	≥ 2	1	Bin($N-1$; N, 1/2)	$\alpha_{7,N-1}/P_{aa}(T)$
8	≥ 2	≥ 2	Bin(k; N, 1/2)	$\alpha_{8,k}/P_{aa}(T)$

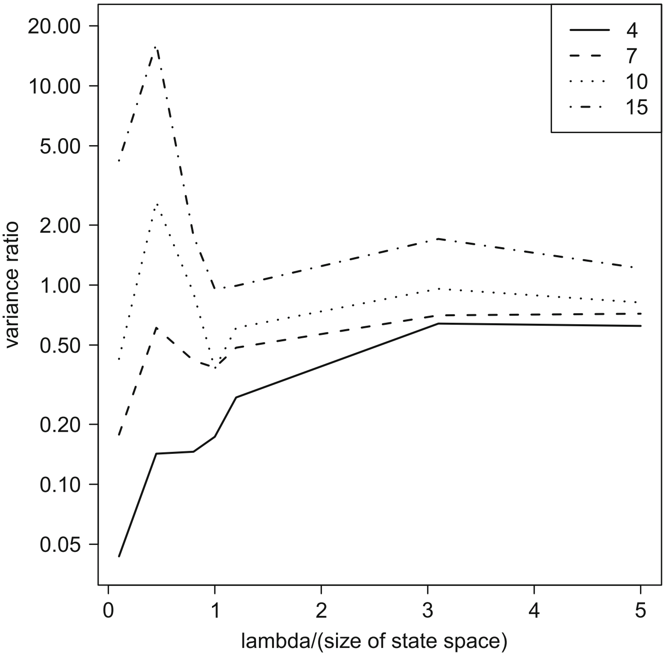


Fig. 6 Variance reduction using importance sampling. The variance ratio is the ratio between the variance from importance sampling and the variance from bisection. The variance ratio is shown as a function of λ/N.

5.3 *Quasi Monte Carlo*

We finally consider a quasi Monte Carlo approach, cf., e.g. [2, IX.3] . Here quasi-random numbers replace the pseudo-random numbers in ordinary Monte Carlo. A difficulty in the implementation is that for the bisection algorithm, the number

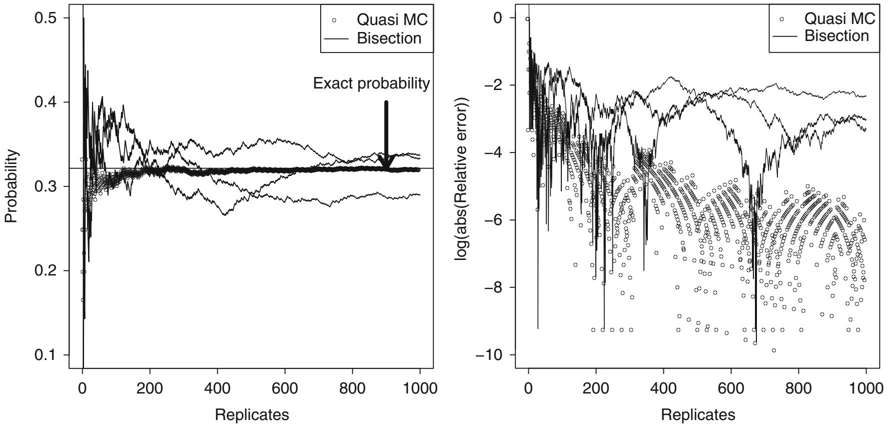


Fig. 7 Quasi Monte Carlo. *Left*: Raw estimation of the probability for one cycle as a function of the number of replicates. *Right*: Log of the absolute value of the relative error (defined as (true-obs)/true) as a function of the number of replicates.

of random numbers that need to be generated is random rather than fixed, which may lead to serious problems, cf. [2, item (ii) p. 271]. To avoid this problem, we choose a hybrid implementation, where only the midpoint and the number of jumps in the two intervals are generated from quasi-random numbers (for simplicity of implementation, we used the three-dimensional Halton sequence). The remaining part of the bisection algorithm is as before. In Fig. 7 we compare the two sampling schemes. It is quite clear that QMC is a very efficient strategy to improve the convergence rate for the algorithm.

6 Conclusions and Extensions

1. As mentioned in the Introduction, we do not believe that bisection in itself is a major improvement of existing methods for simulating CTMC bridges, but that the justification of the method rather is its potential for variance reduction. We find this potential well illustrated via the numerical examples, stressing that these are rather crude by only using variance reduction methods for the midpoint $T/2$. A substantial improvement is to be expected if in addition one incorporates $T/4, 3T/4$ and so on. For stratification, this is unfeasible for even moderate state spaces, since the order of strata increases from $4N$ to $(4N)^3$ by just going from $T/2$ to $T/2, T/4, 3T/4$. However, the situation is much better for importance sampling and quasi Monte Carlo, and in particular, such an extension could well dramatically change the somewhat disappointing behavior of importance sampling.

2. Another extension not implemented here is hybrid algorithms where the bisection is only used to generate say $X(T/2)$, $X(1/4)$, $X(3T/4)$ (and possibly the number of jumps in each of the four intervals), to apply variance reduction techniques ideas to these r.v.'s only and generate the rest of the sample path by some other algorithm, say rejection sampling which is much faster (see [3, 20]) when the endpoint conditioning is not rare.
3. A phase-type distribution is the distribution of the absorption time τ of a Markov process X^* on $\{0, 1, \ldots, M\}$, where 0 is absorbing and the states in $1, \ldots, M$ non-absorbing, and having some specified initial probabilities ξ_a, $a = 1, \ldots, M$. In simulation-based statistical estimation, one needs to generate a sample path of X^* conditioned on $\tau = T$. An algorithm is suggested in [8] and uses Gibbs sampling.

   The problem can, however, be translated to endpoint conditioned simulation. To this end, one simply computes the probability η_b that $X^*(\tau-) = b$ (this reduces to simple matrix algebra but we omit the details). One then draws a, b according to the ξ_a and η_b, and simulates X^* conditioned to have endpoints a, b and no transitions to state 0 in $[0, T]$.
4. Another potential application of the bisection algorithm is in combination with the uniformization algorithm. To this end, one first notes that since it is not essential to split intervals into two of exactly equal size, our algorithm applies with minor changes to discrete time Markov chains, in this case the chain at Poisson times. Doing so has the potential advantage that a segment of length K where the Markov chain is constant can be simulated in a single step instead of K steps. This is appealing in situations where the q_i are of different orders of magnitudes, since then segments with large K are likely to show up in the sample path.

Appendix

Consider a CTMC $\{X(t) : 0 \leq t \leq T\}$ with rate matrix Q and endpoints $X(0) = a$ and $X(T) = b$. In this Appendix we provide a recursion for the number of substitutions using the approach suggested by Siepel et al. [32].

Consider a uniformization of the process. Let

$$R = I + \frac{1}{\mu} Q,$$

where $\mu = \max_c q_c$. Furthermore, let J denote the (stochastic) number of jumps (including virtual) and N the (stochastic) number of substitutions (excluding the virtual jumps). Siepel, Pollard and Haussler's formula for the number of substitutions is based on the following fundamental observation

$$
\begin{aligned}
&P(N(T) = n, X(T) = b \mid X(0) = a) = \\
&\sum_{j=n}^{\infty} P(N(T) = n, X(T) = b, J(T) = j|X(0) = a) = \\
&\sum_{j=n}^{\infty} P(N(T) = n, X(T) = b|J(T) = j, X(0) = a)P(J(T) = j|X(0) = a) = \\
&\sum_{j=n}^{\infty} P(n, b|j, a)\text{Pois}(j|\mu T).
\end{aligned} \tag{8}
$$

Here $\text{Pois}(\cdot|\mu T)$ is the Poisson distribution with rate μT. Note that $P(n, b|j, a)$ does not depend on the time interval. Also note that we can find the transition probabilities from

$$P_{ab}(T) = P(b|a, T) = \sum_{n=0}^{\infty} P(N(T) = n, X(T) = b|X(0) = a).$$

This formula provides a way of calculating the transition probability without using a diagonalization of the rate matrix.

Having calculated $P(N(T) = n, X(T) = b|X(0) = a)$ we can easily find the distribution for the number of endpoint-conditioned substitutions

$$P(N(T) = n|X(0) = a, X(T) = b) = \frac{P(N(T) = n, X(T) = b|X(0) = a)}{P(X(T) = b|X(0) = a)}.$$

The crucial step for (8) to be useful is a fast way of calculating the quantities $P(n, b|j, a)$, and [32] provide a recursion for accomplishing this task.

First note that $P(n, b|j, a) = 0$ if $n > j$.
For $j = 0$ we have

$$P(n, b|j = 0, a) = \begin{cases} 1 \text{ if } a = b \text{ and } n = 0 \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

This provides the basis of the recursion.

For $j \geq 1$ we find $P(n, b|j, a)$ for $0 \leq n \leq j$ using the recursion

$$
\begin{aligned}
P(n, b|j, a) &= P(N = n, Y(j) = b|J = j, Y(0) = a) \\
&= P(N = n, Y(j) = b, Y(j-1) = b|J = j, Y(0) = a) + \\
&\quad \sum_{c \neq b} P(N = n, Y(j) = b, Y(j-1) = c|J = j, Y(0) = a) \\
&= R_{bb} P(n, b|j-1, a) + R_{cb} P(n-1, c|j-1, a),
\end{aligned} \tag{10}
$$

where Y is the uniformized (auxiliary) process that includes the virtual jumps.

The actual implementation of the recursion is described in the following algorithm:

1. **Initialization** Fix a to the desired value and calculate basis of recursion using (9). Set $j = 1$.
2. **Recursion** Define matrix $M_j(b, n)$ with number of rows equal to the size of the state space and $(j + 1)$ columns. Calculate entries $M_j(b, n) = P(n, b|a, j)$ using (10).
3. **Stopping Criteria** If $\sum_{n,b} M_j(b, n) = 1$ to machine precision, then stop. Otherwise set $j = j + 1$ and go to 2.

References

1. Asmussen, S. (2003). *Applied Probability and Queues* (2nd ed.). Springer-Verlag.
2. Asmussen, S. and Glynn, P.W. (2007). *Stochastic Simulation. Algorithms and Analysis* Springer-Verlag.
3. Asmussen, S. and Hobolth, A. (2008). Bisection ideas in end-point conditioned Markov process simulation. In *Proceedings of the 7th International Workshop on Rare Event Simulation* (G. Rubino and B. Tuffin, eds.), 121–130. Rennes, France.
4. Avramidis, A.N., L'Ecuyer, P. and Tremblay, P.-A. (2003) Efficient simulation of gamma and variance-gamma processes. In *Proceedings of the 2003 Winter Simulation Conference* (S. Chick *et al.*, eds.).
5. Beskos, A., Papaspiliopoulos,O. and Roberts, G.O. (2006) Retrospective exact simulation of diffusion sample paths with applications. *Bernoulli* **12**, 1077–1098.
6. Beskos, A., Papaspiliopoulos, O. and Roberts, G.O. (2008a). A factorisation of diffusion measure and finite sample path constructions. *Methodol. Comput. Appl. Probab.* **10**, 85–104
7. Beskos, A., Roberts, G.O., Stuart, A. and Voss, J. (2008b) MCMC methods for diffusion bridges. *Stoch. Dyn.* **8**, 319–350.
8. Bladt, M., Gonzales, A. and Lauritzen, S.L. (2003). The estimation of phase-type related functionals using Markov chain Monte Carlo. *Scand. Act. J.* **4**, 280–300.
9. Bladt, M. and Sørensen, M. (2009) Simple simulation of diffusion bridges with application to likelihood inference for diffusions.
   http://www.math.ku.dk/michael/diffusionbridge0809.pdf
10. Caflish, R.E. and Moskowitz, B. (1995) Modified Monte Carlo methods using quasi-random sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P.J.-S. Shine, eds.). Lecture Notes in Statistics **106**, 1–16. Springer-Verlag
11. Caflish, R.E., Morokoff, W. and Owen, A.B. (1997). Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. *J. Comput. Finance* **1**, 27–46.
12. de Koning, A.P.J., Gu, J. and Pollock, D.D. (2010). Rapid Likelihood Analysis on Large Phylogenies Using Partial Sampling of Substitution Histories. *Mol Biol Evol* **27**, 249–265.
13. Duret, L. (2008). Neutral theory: The null hypothesis of molecular evolution. *Nature Education* 1(1).
14. Ewens, W.J. and Grant, G.R. (2001). *Statistical methods in Bioinformatics.* Springer-Verlag.
15. Fearnhead, P. and Sherlock, C. (2006). An exact Gibbs sampler for the Markov-modulated Poisson process. *J.R. Statist. Soc. B* **68**, 767–784.
16. Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* **17**, 368–376.
17. Felsenstein, J. (2004). *Inferring Phylogenies.* Sinauer Associates, Inc.

18. Fitzsimmons, P., Pitman, J. and Yor, M. (1992) Markovian bridges: construction, Palm interpretation and splicing. In *Seminar on Stochastic Processes*. Progress in Probability **32**, 101–134.
19. Hobolth, A. (2008) A Markov chain Monte Carlo expectation maximization algorithm for statistical analysis of DNA sequence evolution with neighbour-dependent substitution rates. *Journal of Computational and Graphical Statistics* **17**, 138–164.
20. Hobolth, A. and Stone, E.A. (2009). Efficient simulation from finite-state, continuous-time Markov chains with incomplete observations. *Ann. Appl. Statist.* **3**, 1204–1231.
21. Hwang, D.G. and Green, P. (2004). Bayesian Markov chain Monte Carlo sequence analysis reveals varying neutral substitution patterns in mammalian evolution. *PNAS*, **101**, 13994–14001.
22. Jensen, J. L. and Pedersen, A.-M. K. (2000). Probabilistic models of DNA sequence evolution with context dependent rates of substitution. *Adv. Appl. Prob.*, **32**, 499–517.
23. Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* **16**, 111–120.
24. Leobacher, G. (2006) Stratified sampling and quasi-Monte Carlo simulation of Lévy processes. *Monte Carlo Methods and Applications* **12**, 231–238.
25. Metzner, P., Dittmer, E., Jahnke, T. and Schútte, C. (2007). Generator estimation of Markov jump processes. *Journal of Computational Physics* **227**, 353–375.
26. Moler, C. and van Loan, C. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* **45**, 3–49.
27. Moskowitz, B. and Caflish, R.E. (1996) Smoothness and dimension reduction in quasi-Monte Carlo methods. *Journal of Mathematical and Computer Modeling* **23**, 37–54.
28. Nielsen, R. (2002). Mapping mutations on phylogenies. *Syst. Biol.*, **51**, 729–739.
29. Ribeiro, C. and Webber, N. (2003) Valuing path-dependent options in the variance-gamma model by Monte Carlo with a gamma bridge. *J. Comput. Finance* **7**, 81–100.
30. Ribeiro, C. and Webber, N. (2006) Correction for simulation bias in Monte Carlo methods to value exotic options in models driven by Lévy processes. *Appl. Math. Finance* **13**, 333–352.
31. Roberts, G.O. and Stramer, O. (2001) On inference for partially observed nonlinear diffusion processes using Metropolis-Hastings algorithms. *Biometrika* **88**, 603–621.
32. Siepel, A., Pollard, K.S. and Haussler, D. (2006). New methods for detecting lineage-specific selection. *Proceedings of the 10th International Conference on Research in Computational Molecular Biology (RECOMB)*, 190–205.

Upper Bounds in Discrepancy Theory

William W.L. Chen

Abstract Through the use of a few examples, we shall illustrate the use of probability theory, or otherwise, in the study of upper bound questions in the theory of irregularities of point distribution. Such uses may be Monte Carlo in nature but the most efficient ones appear to be quasi Monte Carlo in nature. Furthermore, we shall compare the relative merits of probabilistic and non-probabilistic techniques, as well as try to understand the actual role that the probability theory plays in some of these arguments.

1 Introduction

Discrepancy theory concerns the comparison of the discrete, namely an actual point count, with the continuous, namely the corresponding expectation. Since the former is always an integer while the latter can take a range of real values, such comparisons inevitably lead to discrepancies. Lower bound results in discrepancy theory support the notion that no point set can, in some sense, be too evenly distributed, while upper bound results give rise to point sets that are as evenly distributed as possible under such constraints.

Let us look at the problem from a practical viewpoint. Consider an integral

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

where $f:[0,1]^K \to \mathbf{R}$ is a real valued function in K real variables. Of course, this integral simply represents the average value of the function f in $[0,1]^K$. If we are unable to evaluate this integral analytically, we may elect to select a large number of points $\mathbf{p}_1,\ldots,\mathbf{p}_N \in [0,1]^K$, and use the discrete average

W.W.L. Chen
Macquarie University, Sydney, NSW 2109, Australia
e-mail: william.chen@mq.edu.au

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_2, © Springer-Verlag Berlin Heidelberg 2012

$$\frac{1}{N}\sum_{j=1}^{N} f(\mathbf{p}_j)$$

as an approximation, resulting in an error

$$\frac{1}{N}\sum_{j=1}^{N} f(\mathbf{p}_j) - \int_{[0,1]^K} f(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Suppose next that $f = \chi_A$, the characteristic function of some measurable set $A \subseteq [0,1]^K$. Then the above error, without the normalization factor N^{-1}, becomes

$$\sum_{j=1}^{N} \chi_A(\mathbf{p}_j) - N\int_{[0,1]^K} \chi_A(\mathbf{x})\,\mathrm{d}\mathbf{x} = \#(\mathscr{P} \cap A) - N\mu(A),$$

the discrepancy of the set $\mathscr{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ in A. Often we consider a collection $\mathscr{A}$ of such measurable sets $A \subseteq [0,1]^K$; an often considered example of $\mathscr{A}$ is the collection of all aligned rectangular boxes in $[0,1]^K$ which are anchored at the origin. Upper bound problems in discrepancy theory involve finding point sets that are good, in some sense, with respect to all the sets in $\mathscr{A}$.

Naturally, we try if possible to construct explicitly a good point set. However, when this is not possible, then the next best alternative is to show nevertheless that a good point set exists, by the use of probabilistic techniques. Thus, in upper bound arguments, we may use probability with great abandon, use probability with careful control, or not use probability at all. These correspond respectively to the three approaches, namely Monte Carlo, quasi Monte Carlo or deterministic. We remark here that the experts may find the use of the term quasi Monte Carlo here a little unusual.

There are a number of outcomes and questions associated with a probabilistic approach. First of all, we may end up with a very poor point distribution or a very good point distribution. It is almost certain that we lose explicitness. However, it is important to ask whether the probability is necessary, and if so, what it really does.

This brief survey is organized as follows. In Sect. 2, we discuss some basic ideas by considering a large discrepancy example. In Sect. 3, we take this example a little further and compare the merits of the three different approaches. We then discuss in Sect. 4 the classical problem, an example of small discrepancy. We continue with this example in Sect. 5 to give some insight into what the probability really does.

Notation: Throughout, $\mathscr{P}$ denotes a distribution of N points in $[0,1]^K$. For any measurable subset $B \subseteq [0,1]^K$, we let $Z[\mathscr{P}; B] = \#(\mathscr{P} \cap B)$ denote the number of points of $\mathscr{P}$ that fall into B, with corresponding expected point count $N\mu(B)$. We then denote the discrepancy by $D[\mathscr{P}; B] = Z[\mathscr{P}; B] - N\mu(B)$.

Also, $\mathscr{Q}$ denotes a distribution of points in $[0,1)^k \times [0,\infty)$ with a density of one point per unit volume. For any measurable subset $B \subseteq [0,1)^k \times [0,\infty)$, we consider the corresponding discrepancy $E[\mathscr{Q}; B] = \#(\mathscr{Q} \cap B) - \mu(B)$.

For any real valued function f and non-negative function g, we write $f = O(g)$ or $f \ll g$ to indicate that there exists a positive constant c such that $|f| < cg$. For any non-negative functions f and g, we write $f \gg g$ to indicate that there exists a positive constant c such that $f > cg$, and write $f \asymp g$ to denote that $f \ll g$ and $f \gg g$. The symbols $\ll$ and $\gg$ may be endowed with subscripts, and this means that the implicit constant c may depend on these subscripts.

Remark 1. The author has taken the liberty of omitting unnecessary details and concentrate mainly on the ideas, occasionally at the expense of accuracy. The reader will therefore find that some definitions and details in this survey will not stand up to closer scrutiny.

2 A Large Discrepancy Example

Let $\mathscr{A}$ denote the collection of all discs in the unit torus $[0,1]^2$ of diameter less than 1.

A special case of a result of Beck [3] states that for every distribution $\mathscr{P}$ of N points in $[0,1]^2$, we have the lower bound

$$\sup_{A \in \mathscr{A}} |D[\mathscr{P}; A]| \gg N^{\frac{1}{4}}. \tag{1}$$

An alternative proof of this result can be found in Montgomery [23].

The lower bound (1) is almost sharp, since for every natural number $N \geq 2$, there exists a distribution $\mathscr{P}$ of N points in $[0,1]^2$ such that

$$\sup_{A \in \mathscr{A}} |D[\mathscr{P}; A]| \ll N^{\frac{1}{4}} (\log N)^{\frac{1}{2}}, \tag{2}$$

similar to a special case of an earlier result of Beck [2]. We shall indicate some of the ideas behind this upper bound.

Let us assume, for simplicity, that $N = M^2$, where M is a natural number, and partition $[0,1]^2$ into $N = M^2$ little squares in the obvious and natural way to create the collection $\mathscr{S}$ of all the little squares S. We then place one point anywhere in each little square $S \in \mathscr{S}$, and let $\mathscr{P}$ denote the collection of all these points.

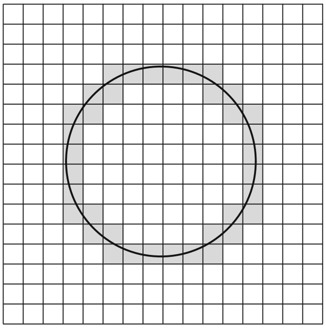
Now take any disc $A \in \mathscr{A}$, and try to bound the term $|D[\mathscr{P}; A]|$ from above. Since discrepancy is additive with respect to disjoint unions, we have

$$D[\mathscr{P}; A] = \sum_{S \in \mathscr{S}} D[\mathscr{P}; S \cap A].$$

It is easy to see that for any little square $S \in \mathscr{S}$ such that $S \cap A = \emptyset$ or $S \subseteq A$, we have $D[\mathscr{P}; S \cap A] = 0$. Hence

$$D[\mathscr{P}; A] = \sum_{\substack{S \in \mathscr{S} \\ S \cap \partial A \neq \emptyset}} D[\mathscr{P}; S \cap A],$$

where ∂A denotes the boundary of A.



It then follows easily that

$$|D[\mathscr{P}; A]| \leq \sum_{\substack{S \in \mathscr{S} \\ S \cap \partial A \neq \emptyset}} |D[\mathscr{P}; S \cap A]| \ll M = N^{\frac{1}{2}},$$

rather weak in comparison to what we hope to obtain.

In order to improve on this rather trivial upper bound, we next adopt a quasi Monte Carlo approach.

For every little square $S \in \mathscr{S}$, let the point $\mathbf{p}_S$ be uniformly distributed within S, and independently from those points in the other little squares. In other words, we have a random point $\widetilde{\mathbf{p}_S} \in S$. Furthermore, we introduce the random variable

$$\xi_S = \begin{cases} 1, \text{ if } \widetilde{\mathbf{p}_S} \in A, \\ 0, \text{ if } \widetilde{\mathbf{p}_S} \notin A, \end{cases}$$

with discrepancy $\eta_S = \xi_S - \mathbf{E}\xi_S$. Clearly $\widetilde{\mathscr{P}} = \{\widetilde{\mathbf{p}_S} : S \in \mathscr{S}\}$ is a random point set, $\{\eta_S : S \in \mathscr{S}\}$ is a collection of independent random variables, and we have

$$D[\widetilde{\mathscr{P}}; A] = \sum_{S \in \mathscr{S}} \eta_S = \sum_{\substack{S \in \mathscr{S} \\ S \cap \partial A \neq \emptyset}} \eta_S. \tag{3}$$

To obtain the desired result, we now simply invoke a large deviation type result in probability theory, for instance due to Hoeffding; see Pollard [24, Appendix B]. In summary, the probability theory enables us to obtain the square root of the trivial estimate, as is clear from the upper bound (2). Perhaps, we can think of the extra factor $(\log N)^{\frac{1}{2}}$ in (2) as the price of using probability.

In fact, for every distribution $\mathscr{P}$ of N points in $[0,1]^2$, the lower bound (1) follows from the stronger lower bound

$$\int_{\mathscr{A}} |D[\mathscr{P};A]|^2 \,\mathrm{d}A \gg N^{\frac{1}{2}},$$

also due to Beck [3]. We next proceed to show that this bound is best possible.

Let us choose $A \in \mathscr{A}$ and keep it fixed. It then follows from (3) that

$$|D[\widetilde{\mathscr{P}};A]|^2 = \sum_{\substack{S_1,S_2\in\mathscr{S}\\ S_1\cap\partial A\neq\emptyset\\ S_2\cap\partial A\neq\emptyset}} \eta_{S_1}\eta_{S_2}.$$

Taking expectation over all N random points, we obtain

$$\mathbf{E}\left(|D[\widetilde{\mathscr{P}};A]|^2\right) = \sum_{\substack{S_1,S_2\in\mathscr{S}\\ S_1\cap\partial A\neq\emptyset\\ S_2\cap\partial A\neq\emptyset}} \mathbf{E}(\eta_{S_1}\eta_{S_2}). \tag{4}$$

If $S_1 \neq S_2$, then η_{S_1} and η_{S_2} are independent, and so

$$\mathbf{E}(\eta_{S_1}\eta_{S_2}) = \mathbf{E}(\eta_{S_1})\mathbf{E}(\eta_{S_2}) = 0.$$

It follows that the only non-zero contributions to the sum in (4) come from those terms where $S_1 = S_2$, and so

$$\mathbf{E}\left(|D[\widetilde{\mathscr{P}};A]|^2\right) \leq \sum_{\substack{S\in\mathscr{S}\\ S\cap\partial A\neq\emptyset}} 1 \ll N^{\frac{1}{2}}.$$

We now integrate over all $A \in \mathscr{A}$ to obtain

$$\mathbf{E}\left(\int_{\mathscr{A}} |D[\widetilde{\mathscr{P}};A]|^2 \,\mathrm{d}A\right) \ll N^{\frac{1}{2}},$$

and the desired result follows immediately.

3 Monte Carlo, Quasi Monte Carlo, or Not

Let $\mathscr{A}$ denote the collection of all discs in the unit torus $[0,1]^2$ of diameter equal to $\frac{1}{2}$. Consider a distribution $\mathscr{P}$ of $N = M^2$ points in $[0,1]^2$, with one point in each little square $S \in \mathscr{S}$. We now randomize these points, or otherwise, in one of the following ways: (1) The point in each S is uniformly distributed in $[0,1]^2$, and independently of other points. This is the Monte Carlo case. (2) The point in each S is uniformly distributed in S, and independently of other points. This is the quasi Monte Carlo case. (3) The point in each S is fixed in the centre of S, so that there is absolutely no probabilistic machinery. This is the deterministic case.

We can take a different viewpoint, and let ν denote a probabilistic measure on $U = [0,1]^2$. Taking the origin as the reference point for ν, for every $S \in \mathscr{S}$, we let ν_S denote the translation of ν to the centre of S, and let $\widetilde{\mathbf{p}_S}$ denote the random point associated to ν_S. Repeating this for every $S \in \mathscr{S}$, we obtain a random point set $\widetilde{\mathscr{P}} = \{\widetilde{\mathbf{p}_S} : S \in \mathscr{S}\}$. Now write

$$D_\nu^2(N) = \int_U \cdots \int_U \left(\int_{\mathscr{A}} |D[\widetilde{\mathscr{P}}; A]|^2 \, \mathrm{d}A \right) \prod_{S \in \mathscr{S}} \mathrm{d}\nu_S.$$

We now choose ν in one of the following ways, corresponding to cases above: (1) We take ν to be the uniform measure supported by $[-\frac{1}{2}, \frac{1}{2}]^2$. (2) We take ν to be the uniform measure supported by $[-\frac{1}{2M}, \frac{1}{2M}]^2$. (3) We take ν to be the Dirac measure δ_0 concentrated at the origin.

Since $\mathscr{A}$ is the collection of all discs in the unit torus $[0,1]^2$ of diameter equal to $\frac{1}{2}$, each $A \in \mathscr{A}$ is a translate of any other, and so

$$\int_{\mathscr{A}} \mathrm{d}A \quad \text{is essentially} \quad \int_U \mathrm{d}\mathbf{x}$$

and this enables us to use Fourier transform techniques, first used in this area by Kendall [21].

Let χ denote the characteristic function of the disc centred at the origin. Then one can show that

$$D_\nu^2(N) = N \sum_{\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2} |\widehat{\chi}(\mathbf{t})|^2 (1 - |\widehat{\nu}(\mathbf{t})|^2) + N^2 \sum_{\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2} |\widehat{\chi}(M\mathbf{t})|^2 |\widehat{\nu}(M\mathbf{t})|^2; \qquad (5)$$

see Chen and Travaglini [13].

Consider first the Monte Carlo case, where the probabilistic measure ν is the uniform measure supported by $[-\frac{1}{2}, \frac{1}{2}]^2$. Then the Fourier transform $\widehat{\nu}$ satisfies $\widehat{\nu}(\mathbf{0}) = 1$ and $\widehat{\nu}(\mathbf{t}) = 0$ whenever $\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2$. In this case, the identity (5) becomes

$$D_\nu^2(N) = N \sum_{\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2} |\widehat{\chi}(\mathbf{t})|^2 \asymp N,$$

a very poor outcome.

Consider next the quasi Monte Carlo case, where the probabilistic measure ν is the uniform measure supported by $[-\frac{1}{2M}, \frac{1}{2M}]^2$. Then

$$\widehat{\nu}(\mathbf{t}) = N \, \frac{\sin(\pi M^{-1} t_1)}{\pi t_1} \, \frac{\sin(\pi M^{-1} t_2)}{\pi t_2},$$

so that $\widehat{\nu}(M\mathbf{t}) = 0$ whenever $\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2$. In this case, the identity (5) becomes

$$D_\nu^2(N) = N \sum_{\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2} |\widehat{\chi}(\mathbf{t})|^2 (1 - |\widehat{\nu}(\mathbf{t})|^2).$$

Consider finally the deterministic case, where the probabilistic measure ν is the Dirac measure concentrated at the origin. Then $\widehat{\nu}(\mathbf{t}) = 1$ identically. In this case, the identity (5) becomes

$$D_\nu^2(N) = N^2 \sum_{\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^2} |\widehat{\chi}(M\mathbf{t})|^2.$$

Which of these two latter cases is superior?

To answer this question fully, it is necessary to consider all higher dimensional analogues of this question. Accordingly, in the K-dimensional unit torus $[0, 1]^K$, where $K \geq 2$, we consider $N = M^K$ little cubes, where M is a natural number. All the definitions in dimension 2 are extended in the natural way to higher dimensions. In the quasi Monte Carlo case, the probabilistic measure ν is the uniform measure λ supported by $[-\frac{1}{2M}, \frac{1}{2M}]^K$, whereas in the deterministic case, the probabilistic measure ν is the Dirac measure δ_0 at the origin.

We now compare the quantities $D_{\delta_0}^2(M^K)$ and $D_\lambda^2(M^K)$, and have the following intriguing results due to Chen and Travaglini [13]:

- For dimension $K = 2$, $D_{\delta_0}^2(M^K) < D_\lambda^2(M^K)$ for all sufficiently large natural numbers M. Hence the deterministic model is superior.
- For all sufficiently large dimensions $K \not\equiv 1 \bmod 4$, $D_\lambda^2(M^K) < D_{\delta_0}^2(M^K)$ for all sufficiently large natural numbers M. Hence the quasi Monte Carlo model is superior.
- For all sufficiently large dimensions $K \equiv 1 \bmod 4$, $D_\lambda^2(M^K) < D_{\delta_0}^2(M^K)$ for infinitely many natural numbers M, and $D_{\delta_0}^2(M^K) < D_\lambda^2(M^K)$ for infinitely many natural numbers M. Hence neither model is superior.

We comment here that the last case is due to the unusual nature of lattices with respect to balls in these dimensions. A closer look at the Bessel functions that arise from the Fourier transforms of their characteristic functions will ultimately remove any intrigue.

4 The Classical Problem

The most studied example of small discrepancy concerns the classical problem of the subject.

Let $\mathscr{P}$ be distribution of N points in the unit cube $[0,1)^K$, where the dimension $K \geq 2$ is fixed. For every $\mathbf{x} = (x_1, \ldots, x_K) \in [0,1]^K$, we consider the rectangular box $B(\mathbf{x}) = [0, x_1) \times \ldots \times [0, x_K)$ anchored at the origin, with discrepancy

$$D[\mathscr{P}; B(\mathbf{x})] = Z[\mathscr{P}; B(\mathbf{x})] - N x_1 \ldots x_k.$$

We are interested in the extreme discrepancy

$$\|D[\mathscr{P}]\|_\infty = \sup_{\mathbf{x}\in[0,1]^K} |D[\mathscr{P}; B(\mathbf{x})]|,$$

as well as average discrepancies

$$\|D[\mathscr{P}]\|_W = \left(\int_{[0,1]^K} |D[\mathscr{P}; B(\mathbf{x})]|^W \, \mathrm{d}\mathbf{x}\right)^{\frac{1}{W}},$$

where W is a positive real number.

The extreme discrepancy gives rise to the most famous open problem in the subject. First of all, an upper bound result of Halton [19] says that for every natural number $N \geq 2$, there exists a distribution $\mathscr{P}$ of N points in $[0,1]^K$ such that

$$\|D[\mathscr{P}]\|_\infty \ll_K (\log N)^{K-1}. \tag{6}$$

Also, it is well known that for every $K \geq 2$, there exists a real number $\eta(K) > 0$ such that for every distribution $\mathscr{P}$ of N points in $[0,1]^K$, we have the lower bound

$$\|D[\mathscr{P}]\|_\infty \gg_K (\log N)^{\frac{K-1}{2}+\eta(K)}. \tag{7}$$

In dimension $K = 2$, the inequality (7) holds with $\eta(2) = \frac{1}{2}$, and this goes back to the famous result of Schmidt [27]. The case $K \geq 3$ is the subject of very recent groundbreaking work of Bilyk et al. [6]. However, the constant $\eta(K)$ is subject to the restriction $\eta(K) \leq \frac{1}{2}$, so there remains a huge gap between the lower bound (7) and the upper bound (6). This is known as the *Great Open Problem*. In particular, there has been no real improvement on the upper bound (6) for over 50 years.

On the other hand, the average discrepancies $\|D[\mathscr{P}]\|_W$ are completely resolved for every real number $W > 1$ in all dimensions $K \geq 2$. The amazing breakthrough result is due to Roth [25] and says that for every distribution $\mathscr{P}$ of N points in $[0,1]^K$, we have the lower bound

$$\|D[\mathscr{P}]\|_2 \gg_K (\log N)^{\frac{K-1}{2}}.$$

The generalization to the stronger lower bound

$$\|D[\mathscr{P}]\|_W \gg_{K,W} (\log N)^{\frac{K-1}{2}}$$

for all real numbers $W > 1$ is due to Schmidt [28], using an extension of Roth's technique. These lower bounds are complemented by the upper bound, established using quasi Monte Carlo techniques, that for every real number $W > 0$ and every natural number $N \geq 2$, there exists a distribution $\mathscr{P}$ of N points such that

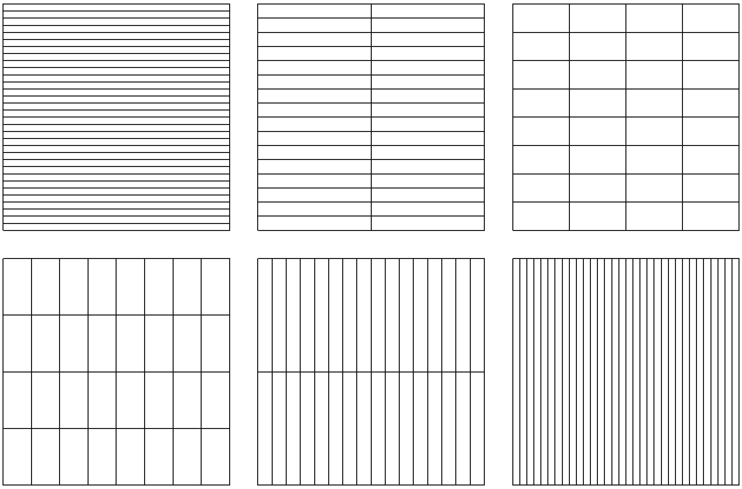
$$\|D[\mathscr{P}]\|_W \ll_{K,W} (\log N)^{\frac{K-1}{2}}. \tag{8}$$

The case $W = 2$ is due to Roth [26], the father of probabilistic techniques in the study of discrepancy theory. The general case is due to Chen [7].

4.1 Two Dimensions

We shall discuss some of the ideas behind the upper bounds (6) and (8) by first concentrating on the special case when the dimension $K = 2$.

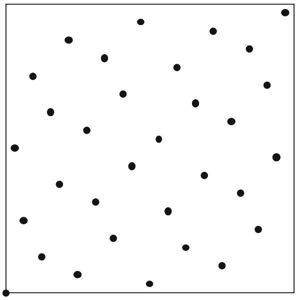
The van der Corput set $\mathscr{P}_h$ of 2^h points must satisfy the following requirement: Suppose that we partition $[0,1]^2$ in the natural way into 2^h congruent rectangles of size $2^{-h_1} \times 2^{-h_2}$, where $0 \leq h_1, h_2 \leq h$ and $h_1 + h_2 = h$. Whatever choice of h_1 and h_2 we make, any rectangle that arises from any such partition must contain precisely one point of $\mathscr{P}_h$. For instance, the van der Corput set $\mathscr{P}_5$ has 32 points, one in each rectangle below.

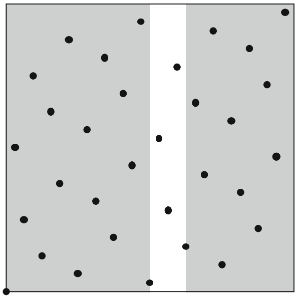


The 2^h points of $\mathscr{P}_h$ are best given in dyadic expansion. We have

$$\mathscr{P}_h = \{(0.a_1 \dots a_h, 0.a_h \dots a_1) : a_1, \dots, a_h \in \{0,1\}\}. \tag{9}$$

Note that the digits of the second coordinates are in reverse order from the digits of the first coordinates. For instance, the 32 points of $\mathscr{P}_5$ are shown in the picture below on the left.





To describe the periodicity properties of the van der Corput set $\mathscr{P}_h$, we again look at $\mathscr{P}_5$. The picture above on the right shows that for those points with first coordinates in the dyadic interval $[4 \times 2^{-3}, 5 \times 2^{-3})$, the second coordinates have period 2^{-2}. Periodicity normally suggests the use of classical Fourier series.

Let us choose a real number $x_1 \in [0,1)$ and keep it fixed. For simplicity, let us assume that x_1 is an integer multiple of 2^{-h}, so that $x_1 = 0.a_1 \dots a_h$ for some digits $a_1, \dots, a_h \in \{0,1\}$. Then

$$[0, x_1) = \bigcup_{\substack{i=1 \\ a_i=1}}^{h} [0.a_1 \dots a_{i-1}, 0.a_1 \dots a_i).$$

Consider now a rectangle of the form $B(x_1, x_2) = [0, x_1) \times [0, x_2)$. Then one can show without too much difficulty that

$$\begin{aligned} D[\mathscr{P}_h; B(x_1,x_2)] &= \sum_{\substack{i=1 \\ a_i=1}}^{h} D[\mathscr{P}_h; [0.a_1 \dots a_{i-1}, 0.a_1 \dots a_i) \times [0, x_2)] \\ &= \sum_{\substack{i=1 \\ a_i=1}}^{h} \left(\alpha_i - \psi\left(\frac{x_2 + \beta_i}{2^{i-h}}\right)\right), \end{aligned} \tag{10}$$

where $\psi(z) = z - [z] - \frac{1}{2}$ is the sawtooth function and the numbers α_i and β_i are constants. Note that the summand is periodic in the variable x_2 with period 2^{i-h}.

Since the summands are bounded, the inequality $|D[\mathscr{P}_h; B(x_1,x_2)]| \ll h$ follows immediately, and we can go on to show that $\|D[\mathscr{P}_h]\|_\infty \ll h$. This is essentially inequality (6) in the case $K = 2$ and $N = 2^h$. A little elaboration of the argument will lead to the inequality (6) in the case $K = 2$ for all $N \geq 2$.

Next, let us investigate $\|D[\mathscr{P}_h]\|_2$. Squaring the expression (10) and expanding, we see clearly that $|D[\mathscr{P}_h; B(x_1, x_2)]|^2$ contains a term of the form

$$\sum_{\substack{i,j=1\\ a_i=a_j=1}}^{h} \alpha_i \alpha_j .$$

This ultimately leads to the estimate

$$\int_{[0,1]^2} |D[\mathscr{P}_h; B(\mathbf{x})]|^2 \, \mathrm{d}\mathbf{x} = 2^{-6}h^2 + O(h),$$

as first observed by Halton and Zaremba [20]. Thus the van der Corput point sets $\mathscr{P}_h$ will not lead to the estimate (8) in the special case $K = W = 2$.

The periodicity in the x_2-direction suggests a quasi Monte Carlo approach. In Roth [26], we consider translating the set $\mathscr{P}_h$ in the x_2-direction modulo 1 by a quantity t to obtain the translated set $\mathscr{P}_h(t)$. Now keep x_2 as well as x_1 fixed. Then one can show without too much difficulty that

$$D[\mathscr{P}_h(t); B(x_1, x_2)] = \sum_{\substack{i=1\\ a_i=1}}^{h} \left(\psi\left(\frac{z_i + t}{2^{i-h}}\right) - \psi\left(\frac{w_i + t}{2^{i-h}}\right)\right), \tag{11}$$

where the numbers z_i and w_i are constants. This is a sum of quasi-orthogonal functions in the probabilistic variable t, and one can show that

$$\int_0^1 |D[\mathscr{P}_h(t); B(x_1, x_2)]|^2 \, \mathrm{d}t \ll h. \tag{12}$$

Integrating trivially over $\mathbf{x} = (x_1, x_2) \in [0, 1]^2$, we finally conclude that there exists $t^* \in [0, 1]$ such that

$$\int_{[0,1]^2} |D[\mathscr{P}_h(t^*); B(x_1, x_2)]|^2 \, \mathrm{d}\mathbf{x} \ll h.$$

We remark that an explicit value for t^* can be found; see the recent paper of Bilyk [5]. This represents an example of derandomization.

Note that the probabilistic technique eschews the effect of the constants α_i in the expression (10). This leads us to wonder whether we can superimpose another van der Corput like point set on the set $\mathscr{P}_h$ in order to remove the constants α_i. If this is possible, then it will give rise to a non-probabilistic approach and an explicit point set. Consider the point set

$$\mathscr{P}_h^* = \{(p_1, 1 - p_2) : (p_1, p_2) \in \mathscr{P}_h\},$$

obtained from $\mathscr{P}_h$ by a reflection across the horizontal line $x_2 = \frac{1}{2}$. Then one can show without too much difficulty that

$$D[\mathscr{P}_h^*; B(x_1, x_2)] = \sum_{\substack{i=1\\a_i=1}}^{h} \left(-\alpha_i - \psi\left(\frac{x_2 + \gamma_i}{2^{i-h}}\right)\right),$$

where the numbers γ_i are constants. Combining this with (11), we conclude that

$$D[\mathscr{P}_h \cup \mathscr{P}_h^*; B(x_1, x_2)] = -\sum_{\substack{i=1\\a_i=1}}^{h} \left(\psi\left(\frac{x_2 + \beta_i}{2^{i-h}}\right) + \psi\left(\frac{x_2 + \gamma_i}{2^{i-h}}\right)\right).$$

This is a sum of quasi-orthogonal functions in the variable x_2, and one can show that for the set $\mathscr{P}_h \cup \mathscr{P}_h^*$ of 2^{h+1} points in $[0, 1]^2$,

$$\int_{[0,1]} |D[\mathscr{P}_h \cup \mathscr{P}_h^*; B(x_1, x_2)]|^2 \, \mathrm{d}x_2 \ll h.$$

This argument is an example of a reflection principle introduced by Davenport [15]. See also Chen and Skriganov [10].

To summarize, if (10) were a sum of quasi-orthogonal functions with respect to the variable x_2, then we would be able to derive the inequality

$$\int_{[0,1]} |D[\mathscr{P}_h; B(x_1, x_2)]|^2 \, \mathrm{d}x_2 \ll h. \tag{13}$$

However, there is no quasi-orthogonality. By introducing the probabilistic variable t, we are able to replace the expression (10) with the expression (11) which is a sum of quasi-orthogonal functions in the probabilistic variable t, and this leads to the inequality (12) which has the same strength as the inequality (13). In other words, the probability leads to crucial quasi-orthogonality. On the other hand, some crucial quasi-orthogonality can also be brought in by the Davenport reflection principle.

Remark 2. The Davenport reflection principle is only valid in dimension $K = 2$. The absence of such a principle in higher dimensions contributes greatly to the difficulty of finding explicit point sets that satisfy the inequality (8), a problem eventually solved by Chen and Skriganov [11] for the case $W = 2$ and later by Skriganov [30] for all positive real numbers W.

4.2 Higher Dimensions

Many new ideas in the study of upper bounds only come in when we consider the problem in higher dimensions.

Our first task is to generalize the van der Corput sets. To do this, we first rescale the second coordinate of every point in the van der Corput set $\mathscr{P}_h$ given by (9) by a factor 2^h to obtain the set

$$\mathscr{Q}_h = \{(0.a_1 \ldots a_h, a_h \ldots a_1) : a_1, \ldots, a_h \in \{0,1\}\}.$$

Clearly $0 \le a_h \ldots a_1 < 2^h$, and so $\mathscr{Q}_h \subseteq [0,1) \times [0,2^h)$. We next extend $\mathscr{Q}_h$ to an infinite set as follows. Every non-negative integer n can be written in the form

$$n = \sum_{i=1}^{\infty} 2^{i-1} a_i = \ldots a_3 a_2 a_1, \quad a_i \in \{0,1\}.$$

Writing the digits in reverse order and placing them behind the decimal point, we then arrive at the expression

$$x_2(n) = \sum_{i=1}^{\infty} 2^{-i} a_i = 0.a_1 a_2 a_3 \ldots .$$

We now consider the set

$$\mathscr{Q} = \{(x_2(n), n) : n = 0,1,2,\ldots\} \subseteq [0,1) \times [0,\infty).$$

Clearly $\mathscr{Q}_h \subseteq \mathscr{Q}$. It is not difficult to show that every rectangle of the form

$$[\ell 2^{-s}, (\ell+1)2^{-s}) \times [m2^s, (m+1)2^s)$$

in $[0,1) \times [0,\infty)$, where ℓ and m are integers, has unit area and contains precisely one point of $\mathscr{Q}$.

Next we consider van der Corput sets in higher dimensions. We follow the ideas of Halton [19]. Let p be a prime number. Similar to our earlier considerations, every non-negative integer n can be written in the form

$$n = \sum_{i=1}^{\infty} p^{i-1} a_i = \ldots a_3 a_2 a_1, \quad a_i \in \{0,1,\ldots,p-1\}.$$

Writing the digits in reverse order and placing them behind the decimal point, we then arrive at the expression

$$x_p(n) = \sum_{i=1}^{\infty} p^{-i} a_i = 0.a_1 a_2 a_3 \ldots .$$

Now let $p_1, \ldots, p_k$ be prime numbers, and consider the set

$$\mathscr{Q} = \{(x_{p_1}(n), \ldots, x_{p_k}(n), n) : n = 0, 1, 2, \ldots\} \subseteq [0, 1)^k \times [0, \infty).$$

It can then be shown, using the Chinese remainder theorem, that every rectangular box of the form

$$\begin{aligned} &[\ell_1 p_1^{-s_1}, (\ell_1 + 1) p_1^{-s_1}) \times \ldots \times [\ell_k p_k^{-s_k}, (\ell_k + 1) p_k^{-s_k}) \\ &\times [m p_1^{s_1} \ldots p_k^{s_k}, (m + 1) p_1^{s_1} \ldots p_k^{s_k}) \end{aligned} \qquad (14)$$

in $[0, 1)^k \times [0, \infty)$, where $\ell_1, \ldots, \ell_k$ and m are integers, has unit volume and contains precisely one point of $\mathscr{Q}$, provided that $p_1, \ldots, p_k$ are distinct.

The inequality (8) for $W = 2$ can now be established by quasi Monte Carlo techniques if we consider translations

$$\mathscr{Q}(t) = \{(x_{p_1}(n), \ldots, x_{p_k}(n), n + t) : n = 0, 1, 2, \ldots\}$$

of the set $\mathscr{Q}$ using a probabilistic parameter t. We omit the rather messy details.

Remark 3. Strictly speaking, before we consider the translation by t, we should extend the set $\mathscr{Q}$ further to one in $[0, 1)^k \times (-\infty, \infty)$ in a suitable way.

4.3 Good Distributions

The important condition above is that the primes $p_1, \ldots, p_k$ are distinct. We now ask the more general question of whether there exist primes $p_1, \ldots, p_k$, not necessarily distinct, and a point set $\mathscr{Q} \subseteq [0, 1)^k \times [0, \infty)$ such that every rectangular box of the form (14), of unit volume and where $\ell_1, \ldots, \ell_k$ and m are integers, contains precisely one point of $\mathscr{Q}$. For any such instance, we shall say that $\mathscr{Q}$ is good with respect to the primes $p_1, \ldots, p_k$.

Halton's argument shows that good sets $\mathscr{Q}$ exist with respect to distinct primes $p_1, \ldots, p_k$. A construction of Faure [18] shows that good sets $\mathscr{Q}$ exist with respect to primes $p_1, \ldots, p_k$, provided that $p_1 = \ldots = p_k \geq k$. No other good sets $\mathscr{Q}$ are currently known.

The good sets constructed by Halton have good periodicity properties, and thus permit a quasi Monte Carlo technique using a translation parameter t. However, the good sets constructed by Faure do not have such periodicity properties, and so do not permit a similar quasi Monte Carlo technique. The challenge now is to find a quasi Monte Carlo technique that works in both instances as well as for any other good point sets that may arise. The answer lies in digit shifts introduced by Chen [8].

Let us first restrict ourselves to two dimensions, and consider a good set

$$\mathscr{Q} = \{(x_p(n), n) : n = 0, 1, 2, \ldots\} \subseteq [0,1) \times [0,\infty);$$

note that here $x_p(n)$ may not be obtained from n by the digit-reversing process we have described earlier for Halton sets. The number of digits that we shift depends on the natural number $N \geq 2$, the cardinality of the finite point set $\mathscr{P}$ we wish to find. Normally, we choose a non-negative integer h determined uniquely by the inequalities $2^{h-1} < N \leq 2^h$, so that $h \asymp \log N$. Suppose that

$$x_p(n) = \sum_{i=1}^{\infty} p^{-i} a_i = 0.a_1 a_2 a_3 \ldots .$$

For every $\mathbf{b} = (b_1, \ldots, b_h)$, where $b_1, \ldots, b_h \in \{0, 1, \ldots, p-1\}$, let

$$x_p^{\mathbf{b}}(n) = 0.a_1 a_2 a_3 \ldots \oplus 0.b_1 \ldots b_h 000 \ldots,$$

where $\oplus$ denotes digit-wise addition modulo p, and write

$$\mathscr{Q}^{\mathbf{b}} = \{(x_p^{\mathbf{b}}(n), n) : n = 0, 1, 2, \ldots\}.$$

Analogous to (12), we can show that

$$\frac{1}{p^h} \sum_{\mathbf{b} \in \{0,1,\ldots,p-1\}^h} |E[\mathscr{Q}^{\mathbf{b}}; B(x,y)]|^2 \ll_p h.$$

In higher dimensions, we consider a good set

$$\mathscr{Q} = \{(x_{p_1}(n), \ldots, x_{p_k}(n), n) : n = 0, 1, 2, \ldots\} \subseteq [0,1)^k \times [0,\infty),$$

and choose h as above. For every $j = 1, \ldots, k$ and $\mathbf{b}_j \in \{0, 1, \ldots, p_j - 1\}^h$, we define $x_{p_j}^{\mathbf{b}_j}(n)$ in terms of $x_{p_j}(n)$ as before for every $n = 0, 1, 2, \ldots$, and write

$$\mathscr{Q}^{\mathbf{b}_1,\ldots,\mathbf{b}_k} = \{(x_{p_1}^{\mathbf{b}_1}(n), \ldots, x_{p_k}^{\mathbf{b}_k}(n), n) : n = 0, 1, 2, \ldots\}.$$

We can then show that

$$\frac{1}{(p_1 \ldots p_k)^h} \sum_{\substack{j=1,\ldots,k \\ \mathbf{b}_j \in \{0,1,\ldots,p_j-1\}^h}} |E[\mathscr{Q}^{\mathbf{b}_1,\ldots,\mathbf{b}_k}; B(x_1, \ldots, x_k, y)]|^2 \ll_{p_1,\ldots,p_k} h^k.$$

We emphasize that this quasi Monte Carlo approach is independent of choice of $p_1, \ldots, p_k$, so long as $\mathscr{Q}$ is good with respect to the primes $p_1, \ldots, p_k$.

5 Fourier–Walsh Analysis

Much greater insight on the role of probability theory has been gained recently through the study of the classical problem via Fourier–Walsh analysis.

The van der Corput set (9) of 2^h points, together with coordinate-wise and digit-wise addition modulo 2, forms a group which is isomorphic to $\mathbf{Z}_2^h$. The characters of these groups are the classical Walsh functions with values ± 1. To study the discrepancy of these sets, it is therefore natural to appeal to Fourier–Walsh analysis, in particular Fourier–Walsh series.

The more general van der Corput set

$$\mathscr{P}_h = \{(0.a_1 \ldots a_h, 0.a_h \ldots a_1) : 0 \leq a_1, \ldots, a_k < p\}$$

of p^h points, together with coordinate-wise and digit-wise addition modulo p, forms a group which is isomorphic to $\mathbf{Z}_p^h$. The characters of these groups are the base p Walsh functions, or Chrestenson–Levy functions, with values p-th roots of unity. To study the discrepancy of these sets, it is therefore natural to appeal to base p Fourier–Walsh analysis, in particular base p Fourier–Walsh series.

Suppose that a point set $\mathscr{P}$ possesses the structure of vector spaces over $\mathbf{Z}_p$. The work of Skriganov [29] shows that $\mathscr{P}$ is a good point distribution with respect to the norm $\|D[\mathscr{P}]\|_\infty$ provided that the corresponding vector spaces have large weights relative to a special metric. Furthermore, the work of Chen and Skriganov [11] shows that $\mathscr{P}$ is a good point distribution with respect to the norm $\|D[\mathscr{P}]\|_2$ provided that the corresponding vector spaces have large weights simultaneously relative to two special metrics, a Hamming metric and a non-Hamming metric arising from coding theory. Indeed, these large weights are guaranteed by taking $p \geq 2K^2$ if we consider the classical problem in $[0,1]^K$. This is sufficient for dispensing with the quasi Monte Carlo approach.

Suppose now that a distribution $\mathscr{P}$ possesses the structure of vector spaces over $\mathbf{Z}_p$, and suppose that $\mathscr{P}$ contains $N = p^h$ points. Then it can be shown that a good approximation of the discrepancy function $D[\mathscr{P}; B(\mathbf{x})]$ is given by

$$F[\mathscr{P}; B(\mathbf{x})] = N \sum_{\mathbf{l} \in \mathscr{L}} \phi_{\mathbf{l}}(\mathbf{x}),$$

where $\mathscr{L}$ is a finite set depending on $\mathscr{P}$ and $\phi_{\mathbf{l}}(\mathbf{x})$ is a product of certain coefficients of the Fourier–Walsh series of the characteristic functions $\chi_{[0,x_i)}$ of the intervals forming the rectangular box $B(\mathbf{x})$.

If $p \geq 2K^2$, then the functions $\phi_{\mathbf{l}}(\mathbf{x})$ are orthogonal, and so

$$\int_{[0,1]^K} |F[\mathscr{P}; B(\mathbf{x})]|^2 \,\mathrm{d}\mathbf{x} = N^2 \sum_{\mathbf{l}\in\mathscr{L}} \int_{[0,1]^K} |\phi_{\mathbf{l}}(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x}.$$

On ther other hand, if $p < 2K^2$, so that we do not know whether the functions $\phi_{\mathbf{l}}(\mathbf{x})$ are orthogonal, then we consider a suitable group $\mathscr{T}$ of digit shifts $\mathbf{t}$, so that

$$F[\mathscr{P} \oplus \mathbf{t}; B(\mathbf{x})] = N \sum_{\mathbf{l}\in\mathscr{L}} \overline{W_{\mathbf{l}}(\mathbf{t})}\phi_{\mathbf{l}}(\mathbf{x}),$$

where $W_{\mathbf{l}}(\mathbf{t})$ are K-dimensional base p Walsh functions. This quasi Monte Carlo argument then leads to

$$\sum_{\mathbf{t}\in\mathscr{T}} |F[\mathscr{P} \oplus \mathbf{t}; B(\mathbf{x})]|^2 = N^2 \sum_{\mathbf{l}',\mathbf{l}''\in\mathscr{L}} \left(\sum_{\mathbf{t}\in\mathscr{T}} \overline{W_{\mathbf{l}'}(\mathbf{t})} W_{\mathbf{l}''}(\mathbf{t})\right) \phi_{\mathbf{l}'}(\mathbf{x})\overline{\phi_{\mathbf{l}''}(\mathbf{x})}.$$

Using the orthogonality property

$$\sum_{\mathbf{t}\in\mathscr{T}} \overline{W_{\mathbf{l}'}(\mathbf{t})} W_{\mathbf{l}''}(\mathbf{t}) = \begin{cases} \#\mathscr{T}, & \text{if } \mathbf{l}' = \mathbf{l}'', \\ 0, & \text{otherwise,} \end{cases}$$

we conclude immediately that

$$\frac{1}{\#\mathscr{T}} \sum_{\mathbf{t}\in\mathscr{T}} |F[\mathscr{P} \oplus \mathbf{t}; B(\mathbf{x})]|^2 = N^2 \sum_{\mathbf{l}\in\mathscr{L}} |\phi_{\mathbf{l}}(\mathbf{x})|^2.$$

Integrating with respect to $\mathbf{x}$ trivially over $[0,1]^K$, we conclude that

$$\frac{1}{\#\mathscr{T}} \sum_{\mathbf{t}\in\mathscr{T}} \int_{[0,1]^K} |F[\mathscr{P} \oplus \mathbf{t}; B(\mathbf{x})]|^2 \,\mathrm{d}\mathbf{x} = N^2 \sum_{\mathbf{l}\in\mathscr{L}} \int_{[0,1]^K} |\phi_{\mathbf{l}}(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x}.$$

Hence the quasi Monte Carlo methods gives rise to orthogonality via the back door.

For more details, see Chen and Skriganov [11, 12].

6 Further Reading

The oldest monograph on discrepancy theory is due to Beck and Chen [4], and covers the subject from its infancy up to the mid-1980s, with fairly detailed proofs, but is naturally very out of date. A more recent attempt is the beautifully written monograph of Matoušek [22].

The comprehensive volume by Drmota and Tichy [17] contains many results and a very long list of precisely 2,000 references, whereas the recent volume by

Dick and Pillichshammer [16] concentrates on quasi Monte Carlo methods in both discrepancy theory and numerical integration.

The survey by Alexander et al. [1] covers the majority of the main results in discrepancy theory up to the turn of the century, and provides references for the major developments. Shorter surveys, on selected aspects of the subject, are given by Chen [9] and by Chen and Travaglini [14].

References

1. Alexander, J.R., Beck, J., Chen, W.W.L.: Geometric discrepancy theory and uniform distribution. In: Goodman, J.E., O'Rourke, J. (eds.) Handbook of Discrete and Computational Geometry (2nd edition), pp. 279–304. CRC Press (2004)
2. Beck, J.: Balanced two-colourings of finite sets in the square I. Combinatorica **1**, 327–335 (1981)
3. Beck, J.: Irregularities of distribution I. Acta Math. **159**, 1–49 (1987)
4. Beck, J., Chen, W.W.L.: Irregularities of Distribution. Cambridge Tracts in Mathematics **89**, Cambridge University Press (1987)
5. Bilyk, D.: Cyclic shifts of the van der Corput set. Proc. American Math. Soc. **137**, 2591–2600 (2009)
6. Bilyk, D., Lacey, M.T., Vagharshakyan, A.: On the small ball inequality in all dimensions. J. Funct. Anal. **254**, 2470–2502 (2008)
7. Chen, W.W.L.: On irregularities of distribution. Mathematika **27**, 153–170 (1980)
8. Chen, W.W.L.: On irregularities of distribution II. Quart. J. Math. Oxford **34**, 257–279 (1983)
9. Chen, W.W.L.: Fourier techniques in the theory of irregularities of point distribution. In: Brandolini, L., Colzani, L., Iosevich, A., Travaglini, G. (eds.) Fourier Analysis and Convexity, pp. 59–82. Birkhäuser Verlag (2004)
10. Chen, W.W.L., Skriganov, M.M.: Davenport's theorem in the theory of irregularities of point distribution. Zapiski Nauch. Sem. POMI **269**, 339–353 (2000); J. Math. Sci. **115**, 2076–2084 (2003)
11. Chen, W.W.L., Skriganov, M.M.: Explicit constructions in the classical mean squares problem in irregularities of point distribution. J. reine angew. Math. **545**, 67–95 (2002)
12. Chen, W.W.L., Skriganov, M.M.: Orthogonality and digit shifts in the classical mean squares problem in irregularities of point distribution. In: Schlickewei, H.P., Schmidt, K., Tichy, R.F. (eds.) Diophantine Approximation: Festschrift for Wolfgang Schmidt, pp. 141–159. Developments in Mathematics **16**, Springer Verlag (2008)
13. Chen, W.W.L., Travaglini, G.: Deterministic and probabilistic discrepancies. Ark. Mat. **47**, 273–293 (2009)
14. Chen, W.W.L., Travaglini, G.: Some of Roth's ideas in discrepancy theory. In: Chen, W.W.L., Gowers, W.T., Halberstam, H., Schmidt, W.M., Vaughan, R.C. (eds.) Analytic Number Theory: Essays in Honour of Klaus Roth, pp. 150–163. Cambridge University Press (2009)
15. Davenport, H.: Note on irregularities of distribution. Mathematika **3**, 131–135 (1956)
16. Dick, J., Pillichshammer, F.: Digital Nets and Sequences. Cambridge University Press (2010)
17. Drmota, M., Tichy, R.F.: Sequences, Discrepancies and Applications. Lecture Notes in Mathematics **1651**, Springer Verlag (1997)
18. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. **41**, 337–351 (1982)
19. Halton, J.H.: On the efficiency of certain quasirandom sequences of points in evaluating multidimensional integrals. Num. Math. **2**, 84–90 (1960)
20. Halton, J.H., Zaremba, S.K.: The extreme and L^2 discrepancies of some plane sets. Monatsh. Math. **73**, 316–328 (1969)

21. Kendall, D.G.: On the number of lattice points in a random oval. Quart. J. Math. **19**, 1–26 (1948)
22. Matoušek, J.: Geometric Discrepancy. Algorithms and Combinatorics **18**, Springer Verlag (1999, 2010)
23. Montgomery, H.L.: Ten Lectures on the Interface between Analytic Number Theory and Harmonic Analysis. CBMS Regional Conference Series in Mathematics **84**, American Mathematical Society (1994)
24. Pollard, D.: Convergence of Stochastic Processes. Springer Verlag (1984)
25. Roth, K.F.: On irregularities of distribution. Mathematika **1**, 73–79 (1954)
26. Roth, K.F.: On irregularities of distribution IV. Acta Arith. **37**, 67–75 (1980)
27. Schmidt, W.M.: Irregularities of distribution VII. Acta Arith. **21**, 45–50 (1972)
28. Schmidt, W.M.: Irregularities of distribution X. In: Zassenhaus, H. (ed.) Number Theory and Algebra, pp. 311–329. Academic Press (1977)
29. Skriganov, M.M.: Coding theory and uniform distributions. Algebra i Analiz **13** (2), 191–239 (2001). English translation: St. Petersburg Math. J. **13**, 301–337 (2002)
30. Skriganov, M.M.: Harmonic analysis on totally disconnected groups and irregularities of point distributions. J. reine angew. Math. **600**, 25–49 (2006)

Entropy, Randomization, Derandomization, and Discrepancy

Michael Gnewuch

Abstract The star discrepancy is a measure of how uniformly distributed a finite point set is in the d-dimensional unit cube. It is related to high-dimensional numerical integration of certain function classes as expressed by the Koksma-Hlawka inequality. A sharp version of this inequality states that the worst-case error of approximating the integral of functions from the unit ball of some Sobolev space by an equal-weight cubature is exactly the star discrepancy of the set of sample points. In many applications, as, e.g., in physics, quantum chemistry or finance, it is essential to approximate high-dimensional integrals. Thus with regard to the Koksma-Hlawka inequality the following three questions are very important:

1. What are good bounds with explicitly given dependence on the dimension d for the smallest possible discrepancy of any n-point set for moderate n?
2. How can we construct point sets efficiently that satisfy such bounds?
3. How can we calculate the discrepancy of given point sets efficiently?

We want to discuss these questions and survey and explain some approaches to tackle them relying on metric entropy, randomization, and derandomization.

1 Introduction

Geometric discrepancy theory studies the uniformity of distribution of finite point sets. There are many different notions of discrepancies to measure quantitatively different aspects of "uniformity", see, e.g., [5, 16, 25, 58, 62, 68].

M. Gnewuch
Institut für Informatik, Christian-Albrechts-Universität zu Kiel, Christian-Albrechts-Platz 4, 24098, Kiel, Germany
e-mail: mig@informatik.uni-kiel.de

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_3, © Springer-Verlag Berlin Heidelberg 2012

1.1 The Star Discrepancy

A particularly relevant measure is the star discrepancy, which is defined in the following way: Let $P \subset [0,1]^d$ be an n-point set. (We always want to understand an "n-point set" as a "multi-set": It consists of n points, but these points are not necessarily pairwise different.) For $x = (x_1, \ldots, x_d) \in [0,1]^d$ the *local discrepancy* $\Delta(x, P)$ of P in the axis-parallel box anchored at zero $[0,x) := [0,x_1) \times \cdots \times [0,x_d)$ (which we, likewise, simply want to call *test box*) is given by

$$\Delta(x, P) := \lambda_d([0,x)) - \frac{1}{n}|P \cap [0,x)|;$$

here λ_d denotes the d-dimensional Lebesgue measure and $|A|$ denotes the cardinality of a multi-set A. The *star discrepancy* of P is defined as

$$d_\infty^*(P) := \sup_{x \in [0,1]^d} |\Delta(x, P)|.$$

Further quantities of interest are the *smallest possible star discrepancy of any n-point set in* $[0,1]^d$

$$d_\infty^*(n, d) = \inf_{P \subset [0,1]^d \,;\, |P| = n} d_\infty^*(P),$$

and, for $\varepsilon \in (0,1)$, the *inverse of the star discrepancy*

$$n_\infty^*(\varepsilon, d) = \min\{n \in \mathbb{N} \mid d_\infty^*(n, d) \le \varepsilon\}.$$

Although we mainly focus on the star discrepancy, we will also mention from time to time the L_p-star discrepancy of P for $1 \le p < \infty$, which is defined by

$$d_p^*(P) := \left(\int_{[0,1]^d} |\Delta(x, P)|^p \, \mathrm{d}x \right)^{1/p}.$$

1.2 Relation to Numerical Integration

Discrepancy notions are related to multivariate numerical integration. Such relations are put in a quantitative form by inequalities of Koksma-Hlawka- or Zaremba-type. Here we want to state a sharp version of the classical Koksma-Hlawka inequality [51, 56], which relates the star discrepancy to the worst-case error of quasi-Monte Carlo integration on certain function spaces. For other relations of discrepancy notions to numerical integration we refer the reader to the original papers [15, 33, 46–48, 67, 82, 93, 94] or the survey in [68, Chap. 9].

To state a sharp version of the Koksma-Hlawka inequality, let us first define the normed function spaces we want to consider:

Let $H^{1,1}$ be the space of absolutely continuous functions f on $[0, 1]$ whose derivatives f' are integrable. A norm on $H^{1,1}$ is given by $\|f\|_{1,1} := |f(1)| + \|f'\|_{L_1([0,1])}$. The (algebraic) tensor product $\otimes_{i=1}^d H^{1,1}$ consists of linear combinations of functions f of product form $f(x) = f_1(x_1) \cdots f_d(x_d)$, $f_1, \dots, f_d \in H^{1,1}$. The space $H^{1,d}$ is then defined as the closure of $\otimes_{i=1}^d H^{1,1}$ with respect to the norm

$$\|f\|_{1,d} := |f(\mathbf{1})| + \sum_{\emptyset \neq u \subseteq \{1,\dots,d\}} \|f_u'\|_{L_1([0,1]^{|u|})}, \tag{1}$$

where $\mathbf{1}$ denotes the vector $(1, \dots, 1)$ and f_u' is defined by

$$f_u'(x_u) = \frac{\partial^{|u|}}{\prod_{k \in u} \partial x_k} f(x_u, \mathbf{1}), \tag{2}$$

with $(x_u, \mathbf{1})_k = x_k$ if $k \in u$, and $(x_u, \mathbf{1})_k = 1$ otherwise. Then the following theorem holds:

Theorem 1. *Let $t^{(1)}, \dots, t^{(n)} \in [0, 1)^d$, and let I_d be the integration functional and $Q_{d,n}$ be the quasi-Monte Carlo cubature defined by*

$$I_d(f) := \int_{[0,1]^d} f(t)\, \mathrm{d}t \quad \textit{and} \quad Q_{d,n}(f) := \frac{1}{n} \sum_{i=1}^n f(t^{(i)}).$$

Then the worst-case error $e^{\text{wor}}(Q_{n,d})$ *of $Q_{n,d}$ satisfies*

$$e^{\text{wor}}(Q_{n,d}) := \sup_{f \in H^{1,d}\,;\, \|f\|_{1,d}=1} |I_d(f) - Q_{d,n}(f)| = d_\infty^*(t^{(1)}, \dots, t^{(n)}). \tag{3}$$

In particular, we obtain for all $f \in H^{1,d}$

$$|I_d(f) - Q_{d,n}(f)| \le \|f\|_{1,d}\, d_\infty^*(t^{(1)}, \dots, t^{(n)}). \tag{4}$$

Theorem 1 is a corollary of a more general theorem proved by Hickernell et al. in [47]. There the so-called L_∞*-same-quadrant discrepancy*, which covers the star discrepancy as a special case, is related to the worst-case error of quasi-Monte Carlo approximation of multivariate integrals on *anchored* L_1-Sobolev spaces. In the special case of the star discrepancy the anchor is the point $\mathbf{1}$.

Particularly with regard to Theorem 1 the following three questions are very important.

Questions:

(*i*) *What are good bounds with explicitly given dependence on the dimension* d *for the smallest possible discrepancy of any* n*-point set for moderate* n*?*
(*ii*) *How can we construct point sets efficiently that satisfy such bounds?*
(*iii*) *How can we calculate the discrepancy of given point sets efficiently?*

Let us discuss the relevance of these questions for the star discrepancy. If we intend to approximate high-dimensional integrals of functions from $H^{1,d}$ by a quasi-Monte Carlo cubature $Q_{n,d}$, and if we wish to minimize the corresponding worst-case error $e^{\mathrm{wor}}(Q_{n,d})$, then Theorem 1 tells us that we have to minimize the star discrepancy of the set of integration points we want to use. For this purpose it is certainly helpful to have upper bounds for the smallest star discrepancy that we can achieve with n points. In high dimensions cubatures whose number of integration points n are exponential in the dimension are not feasible. That is why we ask in question (i) for good bounds for the smallest possible discrepancy of sample sets of moderate size n. By "moderate" we mean that n does not grow faster than a polynomial of small degree in the dimension d.

Bounds for the smallest discrepancy achievable are certainly useful, but for quasi-Monte Carlo integration we need to have explicit integration points. Therefore question (ii) is essential.

In practice we may have some point sets that are reasonable candidates to use for quasi-Monte Carlo integration. This may be due to several reasons as, e.g., that in those points we can easily evaluate the functions we want to integrate or that those points are in some sense uniformly distributed. Therefore it would be desirable to be able to calculate the star discrepancy of a given set efficiently.

In fact question (iii) is directly related to question (ii) by the *concentration of measure phenomenon*:

Let us assume that we have a class of n-point sets endowed with some probability measure and the expected discrepancy of a random set is small enough for our needs. Under suitable conditions the measure of the discrepancy distribution is sharply concentrated around the expected discrepancy and a large deviation bound ensures that a randomly chosen set has a sufficiently small discrepancy with high probability. In this situation we may consider the following randomized algorithm, which is a *semi-construction* in the sense of Novak and Woźniakowski [66]:

We choose a point set randomly and calculate its actual discrepancy. If it serves our needs, we accept the point set and stop; otherwise we make a new random choice. The large deviation bound guarantees that with high probability we only have to perform a few random trials to receive an acceptable point set.

Apart from the practical problem of choosing the point set according to the law induced by the probability measure, we have to think of ways to calculate the discrepancy of a chosen set efficiently.

In this bookchapter our main goal is to study the bracketing entropy of axis-parallel boxes anchored at zero and use the results, in particular upper bounds for the bracketing number and explicit constructions of bracketing covers of small size, to tackle question (i), (ii), and (iii).

Before we do so, we want to survey known bounds for the smallest possible star discrepancy, the problem of constructing small low-discrepancy samples, and known algorithms to calculate or approximate the star discrepancy of given point sets.

1.3 Known Bounds for the Star Discrepancy

We may distinguish two kinds of bounds for the smallest possible star discrepancy $d^*_\infty(n,d)$: *Asymptotic bounds* which describe the behavior of $d^*_\infty(n,d)$ well in the *asymptotic range*, i.e., for fixed dimension d and a large number of points n (which usually has to be exponential in d, see the discussion in Sect. 1.3.1), and *pre-asymptotic bounds* which describe its behavior well in the *pre-asymptotic range*, i.e., for moderate values of n (which depend at most polynomially on d).

Usually asymptotic bounds do not reveal the explicit dependence of $d^*_\infty(n,d)$ on d, while pre-asymptotic bounds exhibit the dependence of $d^*_\infty(n,d)$ on both parameters n and d. (Thus an alternative terminology might be "dimension-insensitive bounds" and "dimension-sensitive bounds".)

1.3.1 Asymptotic Bounds

For fixed dimension d the asymptotically best upper bounds for $d^*_\infty(n,d)$ that have been proved so far are of the form

$$d^*_\infty(n,d) \le C_d \ln(n)^{d-1} n^{-1}, \quad n \ge 2, \tag{5}$$

see, e.g., the original papers [28,40,65] or the monographs [5,16,25,58,62]. These bounds have been proved constructively, i.e., there are explicit constructions known that satisfy (5) for suitable constants C_d.

For $d = 1$ the set $T = \{1/2n, 3/2n, \ldots, (2n-1)/2n\}$ establishes (5) with $C_1 = 1/2$. For $d = 2$ the bound (5) can be derived from the results of Hardy and Littlewood [41] and of Ostrowski [72,73] (the essential ideas can already be found in Lerch's paper [57]). For $d \ge 3$ the bound (5) was established by Halton, who showed in [40] that the Hammersley points exhibit this asymptotic behavior. The Hammersley points can be seen as a generalization of the two-dimensional point sets obtained in a canonical way from the one-dimensional infinite sequence of van der Corput from [11, 12]. (In general, if one has an infinite $(d-1)$-dimensional low-discrepancy sequence $(t^{(k)})_{k\in\mathbb{N}}$, one canonically gets a d-dimensional low-discrepancy point set $\{p^{(1)}, \ldots, p^{(n)}\}$ for every n by putting $p^{(k)} = ((k-1)/n, t^{(k)})$, see also [58, Sect. 1.1, 2.1].)

Looking at the asymptotic bound (5) it is natural to ask whether it is sharp or not. That it is optimal up to logarithmic factors is clear from the trivial lower bound $1/2n$. A better lower bound was shown by Roth in [76]:

$$d_\infty^*(n,d) \geq c_d \ln(n)^{\frac{d-1}{2}} n^{-1}\,, \quad n \geq 2\,. \tag{6}$$

In fact, Roth proved that the right hand side of (6) is a lower bound for the smallest possible L_2-star discrepancy $d_2^*(n,d)$, and this bound is best possible as was shown for $d = 2$ by Davenport [13], and for $d \geq 3$ by Roth himself [77, 78] and independently by Frolov [30]. Although Roth's lower bound is sharp for the L_2-star discrepancy, it is not optimal for the L_∞-star discrepancy. This was shown by Schmidt in [79]. He established in dimension $d = 2$ the lower bound

$$d_\infty^*(n,2) \geq c_2 \ln(n) n^{-1}\,, \quad n \geq 2\,, \tag{7}$$

and proved in this way that the upper bound (5) is optimal in dimension 2. In dimension $d \geq 3$ improvements of (6) were achieved by Beck [4], and later by Bilyk et al. [6, 7]; but although those improvements are deep mathematical results, their quantitative gain is rather modest. The remaining gap, baptized the "great open problem" by Beck and Chen in [5], has still not been bridged so far.

Nonetheless, the solution of this intricate problem is not overly significant for numerical integration in high dimensions. In particular, bounds of the form (5) give us no helpful information for moderate values of n, since $\ln(n)^{d-1}n^{-1}$ is an increasing function in n as long as $n \leq e^{d-1}$. This means that with respect to d we have to use at least exponentially many integration points to perceive any rate of decay of the right hand side of inequality (5). Additionally it is instructive to compare the convergence rate $n^{-1}\ln(n)^{d-1}$ and the Monte Carlo convergence rate $n^{-1/2}$: For example, in dimension $d = 3$ we have $n^{-1}\ln(n)^{d-1} < n^{-1/2}$ if $n \geq 5504$, but for $d = 10$ we already have $n^{-1}\ln(n)^{d-1} > n^{-1/2}$ for all $n \leq 1.295 \cdot 10^{34}$. Furthermore, point configurations satisfying (5) may lead to constants C_d that depend critically on d. (Actually, it is known for some constructions that the constant C'_d in the representation

$$d_\infty^*(n,d) \leq \left(C'_d \ln(n)^{d-1} + o_d(\ln(n)^{d-1})\right) n^{-1}$$

of (5) tends to zero as d approaches infinity, see, e.g., [2, 62, 65]. Here the o-notation with index d should emphasize that the implicit constant may depend on d; so far no good bounds for the implicit constant or, respectively, the constant C_d in (5), have been published.)

1.3.2 Pre-Asymptotic Bounds

A bound more suitable for high-dimensional integration was established by Heinrich et al. [45], who proved

$$d_\infty^*(n,d) \leq c d^{1/2} n^{-1/2} \quad \text{and} \quad n_\infty^*(d,\varepsilon) \leq \lceil c^2 d \varepsilon^{-2} \rceil\,, \tag{8}$$

where c does not depend on d, n or ε. Here the dependence of the inverse of the star discrepancy on d is optimal. This was also established in [45] by a lower bound for $n^*_\infty(d,\varepsilon)$, which was later improved by Hinrichs [49] to

$$n^*_\infty(d,\varepsilon) \geq c_0 d\varepsilon^{-1} \quad \text{for } 0 < \varepsilon < \varepsilon_0, \tag{9}$$

where $c_0, \varepsilon_0 > 0$ are suitable constants. The proof of (8) uses a large deviation bound of Talagrand for empirical processes [86] and an upper bound of Haussler for covering numbers of Vapnik-Červonenkis classes [42]. In particular, the proof is not constructive but probabilistic, and the proof approach does not provide an estimate for the value of c. (Hinrichs presented a more direct approach to prove (8) with $c \leq 10$ at the Dagstuhl Seminar 04401 "Algorithms and Complexity for Continuous Problems" in 2004, but this result has not been published. Shortly after the submission of this book chapter Aistleitner gave a proof of (8) with $c \leq 10$ [1]. Since it relies on bracketing entropy and the bracketing covers we present in Sect. 2, we added a discussion of his approach in Sect. 3.2.1.)

In the paper [45] the authors proved also two slightly weaker bounds with explicitly known constants: The first one relies on upper bounds for the *average L_p-star discrepancy* for even p, the fact that the L_p-star discrepancy converges to the star discrepancy as p tends to infinity, and combinatorial arguments. For a detailed description of the approach, improvements, and closely related results we refer to [34, 45, 85].

Here we are more interested in the second bound from [45] with explicitly known small constants, which is of the form

$$d^*_\infty(n,d) \leq kd^{1/2}n^{-1/2}\big(\ln(d) + \ln(n)\big)^{1/2}, \tag{10}$$

and leads to

$$n^*_\infty(d,\varepsilon) \leq O(d\varepsilon^{-2}(\ln(d) + \ln(\varepsilon^{-1}))) \tag{11}$$

where essentially $k \approx 2\sqrt{2}$ and the implicit constant in the big-O-notation is known and independent of d and ε. The proof of (10) is probabilistic and relies on Hoeffding's large deviation bound. (A similar probabilistic approach was already used by Beck in [3] to prove upper bounds for other discrepancies.) From a conceptional point of view it uses *bracketing covers* (although in [45] the authors do not call them that way). As we will see later in Sect. 3.3, the probabilistic proof approach can actually be derandomized to construct point sets deterministically that satisfy the discrepancy bound (10).

1.4 Construction of Small Discrepancy Samples

On the one hand there are several construction methods known that provide point sets satisfying (5), and these constructions can be done quite efficiently. So one can construct, e.g., Hammersley points of size n in dimension d with at most $O(dn\ln(n))$ elementary operations. On the other hand it seems to be hard to construct point sets efficiently that satisfy bounds like (8) or (10), although random sets should do this with high probability. That it is not trivial to find such constructions was underlined by Heinrich, who pointed out in [44] that even the following easier problems are unsolved.

Problems:

(*i*) *For each* $\varepsilon > 0$ *and* $d \in \mathbb{N}$, *give a construction of a point set* $\{t^{(1)},\dots,t^{(n)}\} \subset [0,1]^d$ *with* $n \le c_\varepsilon d^{\kappa_\varepsilon}$ *and* $d_\infty^*(t^{(1)},\dots,t^{(n)}) \le \varepsilon$, *where* c_ε *and* κ_ε *are positive constants which may depend on* ε, *but not on* d.

(*ii*) *For each* $n, d \in \mathbb{N}$, *give a construction of a point set* $\{t^{(1)},\dots,t^{(n)}\} \subset [0,1]^d$ *with* $d_\infty^*(t^{(1)},\dots,t^{(n)}) \le c d^\kappa n^{-\alpha}$, *where* c, κ *and* α *are positive constants not depending on* n *or* d.

Although not stated explicitly in [44], these constructions are required to be efficiently executable, preferably in polynomial time in d, and ε^{-1} or n, respectively, see also [66, Open Problem 6]. If our ultimate goal is numerical integration, we may view the construction of low-discrepancy points as a precomputation. Since we can use the resulting integration points for the (efficient) evaluation of various integrands, we may still accept a little bit higher costs for the construction itself.

As stressed by Heinrich, it remains in particular a challenging question whether any of the various known classical constructions satisfies estimates like in problem (i) and (ii) or even the bound (8) or (10).

There had been attempts from computer scientists to construct small low-discrepancy samples [9, 27], but the size of those samples with guaranteed discrepancy at most ε in dimension d is not polynomial in d and ε^{-1}. The size of the best construction is polynomial in ε^{-1} and $(d/\ln(\varepsilon^{-1}))^{\ln(\varepsilon^{-1})}$ [9]. Formally, those constructions solve problem (i) (but not problem (ii)). Obviously, the size of the samples is a lower bound for the costs of the construction, which are therefore not polynomial in d and ε^{-1}.

We will discuss alternative constructions, based on bracketing covers and derandomization in Sect. 3.3.

1.5 Calculating the Star Discrepancy

In some applications it is of interest to measure the quality of certain point sets by calculating their star discrepancy, e.g., to test whether successive pseudo random numbers are statistically independent [62], or whether sample sets are suitable for multivariate numerical integration of particular classes of integrands, cf. Theorem 1. Apart from that, it is particularly interesting with respect to question (ii) that the fast calculation or approximation of the star discrepancy would allow practicable semi-constructions of low-discrepancy samples of moderate size.

It is known that the L_2-star discrepancy of a given n-point set in dimension d can be calculated via Warnock's formula [91] with $O(dn^2)$ arithmetic operations and similar formulas hold for weighted versions of the L_2-star discrepancy. Heinrich and Frank developed an asymptotically faster algorithm for the L_2-star discrepancy using only $O(n\log(n)^{d-1})$ operations for fixed d [29, 43]. (Due to the exponent of the log-term, the algorithm is only practicable in low dimensions.)

What methods are known to calculate or approximate the star discrepancy of a given set P? At the first glance an exact calculation seems to be difficult since the star discrepancy is defined as the supremum over infinitely many test boxes. But for calculating the discrepancy of P exactly it suffices to consider only finitely many test boxes. So if $P = \{p^{(1)}, \dots, p^{(n)}\} \subset [0,1)^d$, let us define

$$\Gamma_j(P) = \{p_j^{(i)} \mid i \in \{1,...,n\}\} \quad \text{and} \quad \bar{\Gamma}_j(P) = \Gamma_j(P) \cup \{1\},$$

and let us put

$$\Gamma(P) = \Gamma_1(P) \times \cdots \times \Gamma_d(P) \quad \text{and} \quad \bar{\Gamma}(P) = \bar{\Gamma}_1(P) \times \cdots \times \bar{\Gamma}_d(P).$$

Then it is not hard to verify that

$$\begin{aligned} d_\infty^*(P) = \max\Bigg\{ & \max_{y \in \bar{\Gamma}(P)} \left(\lambda_d([0,y)) - \frac{|P \cap [0,y)|}{n} \right), \\ & \max_{y \in \Gamma(P)} \left(\frac{|P \cap [0,y]|}{n} - \lambda_d([0,y)) \right) \Bigg\}, \end{aligned} \tag{12}$$

for a proof see, e.g., [38]. Thus we need to consider at most $O(n^d)$ test boxes to compute $d_\infty^*(P)$. For a random n-point set P we have almost surely $|\Gamma(P)| = n^d$, resulting in $\Omega(n^d)$ test boxes that we have to take into account to calculate (12). This underlines that (12) is in general impractical if n and d are large. There are some more sophisticated methods known to calculate the star discrepancy, which are especially helpful in low dimensions. If we have in the one-dimensional case $p^{(1)} \leq p^{(2)} \leq \cdots \leq p^{(n)}$, then (12) simplifies to

$$d_\infty^*(P) = \frac{1}{2n} + \max_{1 \leq i \leq n} \left| p^{(i)} - \frac{2i-1}{2n} \right|,$$

a result due to Niederreiter, see [60, 61].

In dimension $d = 2$ a reduction of the number of steps to calculate (12) was achieved by de Clerck [10]. In [8] her formula was slightly extended and simplified by Bundschuh and Zhu. If we assume $p_1^{(1)} \leq p_1^{(2)} \leq \cdots \leq p_1^{(n)}$ and rearrange for each $i \in \{1, \dots, n\}$ the numbers $0, p_2^{(1)}, \dots, p_2^{(i)}, 1$ in increasing order and rewrite them as $0 = \xi_{i,0} \leq \xi_{i,1} \leq \cdots \leq \xi_{i,i} \leq \xi_{i,i+1} = 1$, then [8, Theorem 1] states that

$$d_\infty^*(P) = \max_{0\le i\le n} \max_{0\le k\le i} \max \left\{ \left| \frac{k}{n} - p_1^{(i)} \xi_{i,k} \right|, \left| \frac{k}{n} - p_1^{(i+1)} \xi_{i,k+1} \right| \right\}.$$

Bundschuh and Zhu provided also a corresponding formula for the three-dimensional case. The method can be generalized to arbitrary dimension d and requires roughly $O(n^d/d!)$ elementary operations. This method was, e.g., used in [92] to calculate the exact discrepancy of particular point sets, so-called (shifted) rank-1 lattice rules (cf. [81]), up to size $n = 236$ in dimension $d = 5$ and to $n = 92$ in dimension $d = 6$. But as pointed out by Winker and Fang in [92], for this method instances like, e.g., sets of size $n \ge 2,000$ in $d = 6$ are completely infeasible.

Another method to calculate the star discrepancy in time $O(n^{1+d/2})$ was proposed by Dobkin et al. in [17]. It uses sophisticated, but complicated data structures, and the authors implemented only asymptotically slightly slower variants of the algorithm in dimension $d = 2$.

The discussion shows that all known methods that calculate the star discrepancy exactly depend exponentially on the dimension d and are infeasible for large values of n and d.

Indeed, the problem of calculating the star discrepancy is *NP-hard*, as was proved in [38]. We will briefly outline the main proof ideas below in this section. In [32] Giannopoulos et al. proved a result on the *parametrized complexity* of the problem of calculating the star discrepancy, namely they showed that it is *$W[1]$-hard* with respect to the parameter d. It follows from [32] that the general problem cannot be solved in time $O(n^{o(d)})$ unless the *exponential time hypothesis* is false, which is widely regarded as extremely unlikely.

Notice that the complexity results above are about the exact calculation of the discrepancy of arbitrary point sets; they do not directly address the complexity of approximating the discrepancy. So what is known about approximation algorithms?

Since in high dimension no efficient algorithm for the exact calculation of the star discrepancy is known, some authors tried to tackle the large scale enumeration problem (12) by using optimization heuristics. In [92] Winker and Fang used *threshold accepting* [26], a refined randomized local search algorithm based on a similar idea as the well-known simulated annealing algorithm [55], to find lower bounds for the star discrepancy. The algorithm performed well in numerical tests on (shifted) rank-1 lattice rules.

In [89] Thiémard gave an *integer linear programming formulation* for the problem and used techniques as cutting plane generation and branch and bound to tackle it. With the resulting algorithm Thiémard performed non-trivial star discrepancy comparisons between low-discrepancy sequences.

The key observation to approach the non-linear expression (12) via linear programming is that one can reduce it to at most $2n$ sub-problems of the type *"optimal volume subintervals with k points"*. These sub-problems are the problems of finding the largest boxes $[0, y)$, $y \in \bar{\Gamma}(P)$, containing k points, $k \in \{0, 1, \dots, n-1\}$, and the smallest boxes $[0, y]$, $y \in \Gamma(P)$, containing ℓ points for $\ell \in \{1, \dots, n\}$. Thiémard conjectured these sub-problems to be NP-hard.

The conjecture of Thiémard is proved rigorously in [38] by establishing the NP-hardness of the optimal volume subinterval problems. Recall that NP-hardness of an optimization problem U is proved by verifying that deciding the so-called threshold language of U is an NP-hard decision problem (see, e.g., [53, Sect. 2.3.3]). Thus actually the NP-completeness of decision problems corresponding to the optimization problems mentioned above is verified. The verification is done by reduction of the problem DOMINATING SET to the maximal volume subinterval problems and of BALANCED SUBGRAPH to the minimal volume subinterval problems, respectively; the graph theoretical decision problems DOMINATING SET and BALANCED SUBGRAPH are known to be NP-hard, see [31,54]. With the help of these NP-hardness results for the optimal volume subinterval problems it is shown that the problem of calculating the star discrepancy itself is NP-hard. (Furthermore, some minor errors occurring in [89] are listed in [38]. Since those errors may lead to incorrect solutions of Thiémard's algorithm for certain instances, it is explained how to avoid their undesired consequences.)

A *genetic algorithm* to approximate the star discrepancy was recently proposed by Shah [80].

In the recent paper [39] a new randomized algorithm to approximate the star discrepancy based on threshold accepting was presented. Comprehensive numerical tests indicate that it improves on the algorithms from [80,89,92], especially in higher dimension $20 \le d \le 50$.

All the approximation algorithms we have mentioned so far have shown their usefulness in practice, but unfortunately none of them provides an approximation guarantee.

An approach that approximates the star discrepancy of a given set P up to a user-specified error δ was presented by Thiémard [87, 88]. It is in principle based on the generation of small bracketing covers (which were not named this way in [87, 88]).

2 Bracketing Entropy

In this section we want to study the bracketing entropy of axis-parallel boxes anchored at zero. We start by introducing the necessary notion.

2.1 Basic Definitions

Definition 1. Let $x, y \in [0,1]^d$ with $x_i \le y_i$ for $i = 1, \dots, d$. We assign a *weight* $W([x,y])$ to the closed box $[x,y] := [x_1, y_1] \times \cdots \times [x_d, y_d]$ by

$$W([x,y]) = \lambda_d([0,y]) - \lambda_d([0,x]).$$

Let $\delta > 0$. The box $[x, y]$ is a *δ-bracket* if $W([x, y]) \leq \delta$. A set $\mathscr{B}$ of δ-brackets whose union covers $[0, 1]^d$ is a *δ-bracketing cover* of $[0, 1]^d$. The *bracketing number* $N_{[]}(d, \delta)$ denotes the smallest cardinality of any δ-bracketing cover of $[0, 1]^d$. Its logarithm $\ln(N_{[]}(d, \delta))$ is the *bracketing entropy* (or *entropy with bracketing*).

The notion of bracketing entropy is well established in empirical process theory, see, e.g., [86, 90]. In some places it will be more convenient for us to use the related notion of δ-covers from [21] instead of the notion of bracketing covers.

Definition 2. Let $\delta > 0$. A finite set Γ is a *δ-cover* of $[0, 1]^d$ if for all $y \in [0, 1]^d$ there exist $x, z \in \Gamma \cup \{0\}$ such that $[x, z]$ is a δ-bracket and $y \in [x, z]$. Let $N(d, \delta)$ denote the smallest cardinality of any δ-cover of $[0, 1]^d$.

If, on the one hand, we have a δ-bracketing cover $\mathscr{B}$, then it is easy to see that

$$\Gamma_{\mathscr{B}} := \{x \in [0, 1]^d \setminus \{0\} \mid \exists y \in [0, 1]^d : [x, y] \in \mathscr{B} \text{ or } [y, x] \in \mathscr{B}\} \tag{13}$$

is a δ-cover. If, on the other hand, Γ is a δ-cover, then

$$\mathscr{B}_{\Gamma} := \{[x, y] \mid x, y \in \Gamma \cup \{0\}\,,\ [x, y] \text{ is a } \delta\text{-bracket}\,,\ x \neq y\}$$

is a δ-bracketing cover. Therefore we have

$$N(d, \delta) + 1 \leq 2N_{[]}(d, \delta) \leq (N(d, \delta) + 1)N(d, \delta). \tag{14}$$

(The second inequality is obviously a weak one, and it would be nice to have a tighter bound.) The bracketing number and the quantity $N(d, \delta)$ are related to the *covering* and the *L_1-packing number*, see, e.g., [21, Remark 2.10].

2.2 Construction of Bracketing Covers

How large is the bracketing entropy and how does a small δ-bracketing cover look like? To get some idea, we have a look at some examples of δ-bracketing covers.

2.2.1 Cells of an Equidistant Grid

To prove (10), Heinrich et al. used in [45] a δ-cover in form of an equidistant grid $E_\delta = \{0, 1/m, 2/m, \ldots, 1\}^d$ with $m = \lceil d/\delta \rceil$. The grid cells, i.e., all closed boxes of the form $[x, x^+]$, where $x_i \in \{0, 1/m, \ldots, 1 - 1/m\}$ and $x_i^+ = x_i + 1/m$ for $i \in \{1, \ldots, d\}$, form a δ-bracketing cover $\mathscr{E}_\delta$. Indeed, the grid cell with the largest weight is $[(1 - 1/m)\mathbf{1}, \mathbf{1}]$ with

$$W([(1 - 1/m)\mathbf{1}, \mathbf{1}]) = 1 - (1 - 1/m)^d \leq d/m \leq \delta.$$

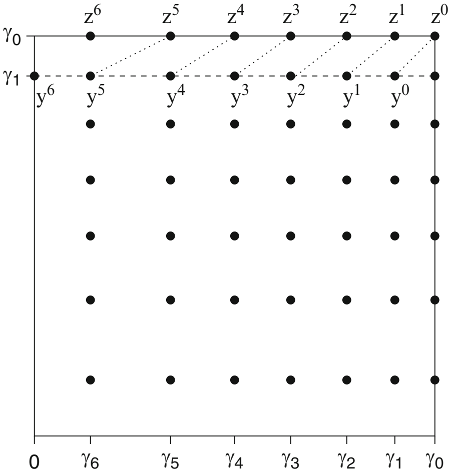


Fig. 1 Construction of the non-equidistant grid Γ_δ for $d = 2$ and $\delta = 0.2$. Here, $\kappa(\delta, d) = 6$.

The cardinality of the δ-bracketing cover $\mathscr{E}_\delta$ is clearly

$$|\mathscr{E}_\delta| = m^d \le (d\delta^{-1} + 1)^d. \tag{15}$$

Although the weight of the grid cell $[(1 - 1/m)\mathbf{1}, \mathbf{1}]$ is nearly δ, the weights of most of the other grid cells are reasonably smaller than δ. For example, the weight of the cell $[0, (1/m)\mathbf{1}]$ is $(1/m)^d \le (\delta/d)^d$, which is for $d \ge 2$ much smaller than δ.

2.2.2 Cells of a Non-equidistant Grid

We generate a smaller δ-bracketing cover by using a *non-equidistant grid* Γ_δ of the form

$$\Gamma_\delta = \{\gamma_0, ..., \gamma_{\kappa(\delta,d)}\}^d , \tag{16}$$

where $\gamma_0, \gamma_1, ..., \gamma_{\kappa(\delta,d)}$ is a decreasing sequence in $(0, 1]$. We calculate this sequence recursively in the following way (cf. Fig. 1):

We set $\gamma_0 := 1$ and choose $\gamma_1 \in (0, 1)$ such that $y^{(0)} := \gamma_1\mathbf{1}$ and $z^{(0)} := \mathbf{1}$ satisfy $W([y^{(0)}, z^{(0)}]) = \delta$. Obviously, $\gamma_1 = (1 - \delta)^{1/d}$. Let γ_i be calculated. If $\gamma_i > \delta$, we compute the real number $\gamma_{i+1} \in (0, \gamma_i)$ that ensures that $y^{(i)} := (\gamma_{i+1}, \gamma_1, ..., \gamma_1)$ and $z^{(i)} := (\gamma_i, 1, ..., 1)$ satisfy $W([y^{(i)}, z^{(i)}]) = \delta$. If $\gamma_i \le \delta$, then we put $\kappa(\delta, d) := i$ and stop. From the geometrical setting it is easy to see that $\gamma_0, \gamma_1, ...$ is a decreasing sequence with $\gamma_i - \gamma_{i+1} \le \gamma_{i+1} - \gamma_{i+2}$. Therefore $\kappa(\delta, d)$ is finite.

The following result was proved in [21, Theorem 2.3].

Theorem 2. *Let* $d \ge 2$*, and let* $0 < \delta < 1$. *Let* $\Gamma_\delta = \{\gamma_0, \gamma_1, \ldots, \gamma_{\kappa(\delta,d)}\}^d$ *be as in (16). Then* Γ_δ *is a* δ*-cover of* $[0, 1]^d$*, and consequently*

$$N(d, \delta) \le |\Gamma_\delta| \le (\kappa(\delta, d) + 1)^d , \tag{17}$$

where

$$\kappa(\delta, d) = \left\lceil \frac{d}{d-1} \frac{\ln(1-(1-\delta)^{1/d}) - \ln(\delta)}{\ln(1-\delta)} \right\rceil. \tag{18}$$

The inequality $\kappa(\delta, d) \le \lceil \frac{d}{d-1} \frac{\ln(d)}{\delta} \rceil$ *holds, and the quotient of the left and the right hand side of this inequality converges to* 1 *as* δ *approaches* 0.

From the δ-cover Γ_δ we obtain a δ-bracketing cover $\mathscr{G}_\delta$ by taking the grid cells of the form $[y, y^+]$, where $y_i = \gamma_j$ for some $j = j(i) \in \{1, \dots, \kappa(\delta, d)\}$ and $y_i^+ = \gamma_{j-1}$ for all $i \in \{1, \dots, d\}$, and the d brackets of the form $[0, z]$ with z having $d-1$ coordinates equal to 1 and one coordinate equal to $\gamma_{\kappa(\delta,d)}$. Thus

$$|\mathscr{G}_\delta| = \kappa(\delta, d)^d + d = \left(\frac{d}{d+1}\right)^d \ln(d)^d \delta^{-d} + O_d\left(\delta^{-d+1}\right); \tag{19}$$

the last identity follows from

$$\kappa(\delta, d) = \frac{d}{d-1} \ln(d)\delta^{-1} + O_d(1) \quad \text{as } \delta \text{ approaches } 0,$$

see [36, Sect. 2]. Note that $(\frac{d}{d-1})^d$ is bounded above by 4 and converges to e as d tends to infinity.

2.2.3 A Layer Construction

By construction the brackets $[y^{(i)}, z^{(i)}]$, $i = 0, 1, \dots, \kappa(\delta, d) - 1$, satisfy $W([y^{(i)}, z^{(i)}]) = \delta$, but it can be shown that the weights of the brackets $[v, w]$ in $\mathscr{G}_\delta$, with $w_i < 1$ for more than one index $i \in \{1, \dots, d\}$, are strictly smaller than δ. It seems obvious that a suitable δ-bracketing cover consisting almost exclusively of brackets with weights exactly δ should exhibit a smaller cardinality than $\mathscr{G}_\delta$. We outline here a construction $\mathscr{Z}_\delta$ which satisfies this specification. To simplify the representation, we confine ourselves to the case $d = 2$ and refer to [35] for a generalization of the construction to arbitrary dimension d. Let δ be given. The essential idea is the following:

We define $a_i = a_i(\delta) := (1 - i\delta)^{1/2}$ for $i = 0, \dots, \zeta = \zeta(\delta) := \lceil \delta^{-1} \rceil - 1$, and $a_{\zeta+1} := 0$. We decompose $[0, 1]^2$ into layers

$$L^{(i)}(\delta) := [0, a_i \mathbf{1}] \setminus [0, a_{i+1} \mathbf{1}), \quad i = 0, \dots, \zeta,$$

and cover each layer separately with δ-brackets. To cover $L^{(0)}(\delta)$, we can simply use the δ-brackets $[y^{(i)}, z^{(i)}]$, $i = 0, 1, \dots, \kappa(\delta, 2) - 1$, from our previous construction and the δ-brackets we obtain after permuting the first and second coordinates of $y^{(i)}$ and $z^{(i)}$, respectively. To cover the remaining layers, we observe that the brackets $[a_{i+1}\mathbf{1}, a_i\mathbf{1}]$, $i = 1, \dots, \zeta - 1$, all have weight δ, and we can cover the layers $L^{(i)}(\delta)$,

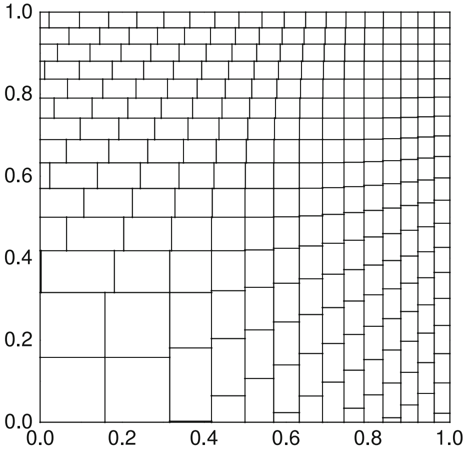


Fig. 2 The layer construction $\mathscr{Z}_\delta$ for $\delta = 0.075$.

$i = 1, \dots, \zeta - 1$, by a straightforward modification of the procedure we used to cover $L^{(0)}(\delta)$.

The final layer $L^{(\zeta)}(\delta) = [0, a_\zeta \mathbf{1}]$ is trivially covered by the δ-bracket $[0, a_\zeta \mathbf{1}]$ itself. Figure 2 shows the resulting bracketing cover $\mathscr{Z}_\delta$ for $\delta = 0.075$.

As shown in [36, Proposition 4.1], the two-dimensional δ-bracketing cover $\mathscr{Z}_\delta$ satisfies

$$|\mathscr{Z}_\delta| = 2\ln(2)\delta^{-2} + O(\delta^{-1}). \tag{20}$$

Notice that the coefficient $2\ln(2) \approx 1.3863$ in front of δ^{-2} is smaller than the corresponding coefficient $(2\ln(2))^2 \approx 1.9218$ in (19).

2.2.4 An Essentially Optimal Construction

The layer construction was generated in a way to guarantee that all δ-brackets have weight exactly δ (except of maybe those which intersect with the coordinate axes). To minimize the number of brackets needed to cover $[0,1]^2$, or, more generally, $[0,1]^d$, it seems to be a good idea to find brackets with weight δ that exhibit maximum volume. The following lemma [35, Lemma 1.1] shows how such δ-brackets look like.

Lemma 1. *Let $d \geq 2$, $\delta \in (0,1]$, and let $z \in [0,1]^d$ with $\lambda_d([0,z]) = z_1 \cdots z_d \geq \delta$. Put*

$$x = x(z,\delta) := \left(1 - \frac{\delta}{z_1 \cdots z_d}\right)^{1/d} z\,. \tag{21}$$

Then $[x,z]$ is the uniquely determined δ-bracket having maximum volume of all δ-brackets containing z. Its volume is

$$\lambda_d([x,z]) = \left(1 - \left(1 - \frac{\delta}{z_1 \cdots z_d}\right)^{1/d}\right)^d \cdot z_1 \cdots z_d .$$

A positive aspect of the previous construction $\mathscr{Z}_\delta$ is that (essentially) all its brackets have largest possible weight δ and overlap only on sets of Lebesgue measure zero. But if we look at the brackets in $\mathscr{Z}_\delta$ which are close to the first or the second coordinate axis and away from the main diagonal, then these boxes do certainly not satisfy the "maximum area criterion" stated in Lemma 1. The idea of the next construction is to generate a bracketing cover $\mathscr{R}_\delta$ similarly as in the previous section, but to "re-orientate" the brackets from time to time in the course of the algorithm to enlarge the area which is covered by a single bracket. Of course this procedure should not lead to too much overlap of the generated brackets. Let us explain the underlying geometrical idea of the construction:

Like all the constructions we have discussed so far, our new bracketing cover should be symmetric with respect to both coordinate axes. Thus we only have to state explicitly how to cover the subset

$$H := \{(x,y) \in [0,1]^2 \,|\, x \le y\}$$

of $[0,1]^2$. For a certain number $p = p(\delta)$ we subdivide H into sectors

$$T^{(h)} := \left\{ (x,y) \in H \setminus \{(0,0)\} \,\middle|\, \frac{h-1}{2^p} \le \frac{x}{y} \le \frac{h}{2^p} \right\} \cup \{(0,0)\}, \quad h = 1, \dots, 2^p.$$

We start with $T^{(2^p)}$ and cover this subset of $[0,1]^2$ in the same way as we covered it in the construction $\mathscr{Z}_\delta$, i.e., we decompose $T^{(2^p)}$ into horizontal stripes $[(0,a_{i+1}),(a_i,a_i)] \cap T^{(2^p)}$, $i = 0,1,\dots,\zeta = \lceil \delta^{-1} \rceil - 1$, and cover each stripe separately with δ-brackets whose weights are (except of maybe one bracket per stripe) exactly δ. Notice that the δ-brackets of $\mathscr{Z}_\delta$ that cover the main diagonal of $[0,1]^2$ are volume optimal due to Lemma 1. Hence, if we choose p sufficiently large, the sector $T^{(2^p)}$ will be thin and all the δ-brackets we use to cover it will have nearly maximum volume.

If $p = 0$, then $H = T^{(2^p)}$ and our new construction $\mathscr{R}_\delta$ will actually be equal to $\mathscr{Z}_\delta$. If $p > 0$, then we have additional sectors $T^{(1)}, \dots, T^{(2^p-1)}$. Again, for a given $i \in \{1, \dots, 2^p - 1\}$ we decompose $T^{(i)}$ into horizontal stripes, but the vertical heights of the stripes increases as i decreases. We essentially choose the heights of each stripe in a way that the bracket on the right hand side of the stripe having this heights and weight exactly δ exhibits maximum volume. Thus, if the sector $T^{(i)}$ is sufficiently thin, again essentially all δ-brackets that cover it will have nearly maximum volume. Therefore we should choose $p = p(\delta)$ large enough. On the other hand, we usually will have overlapping δ-brackets at the common boundary of two sectors. To minimize the number of brackets needed to cover H (and thus $[0,1]^2$), we should try to avoid too much overlap of brackets and consequently not

Fig. 3 The essentially optimal construction $\mathscr{R}_\delta$ for $\delta = 0.075$.

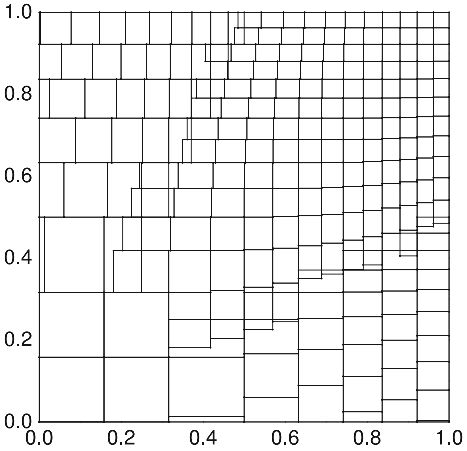
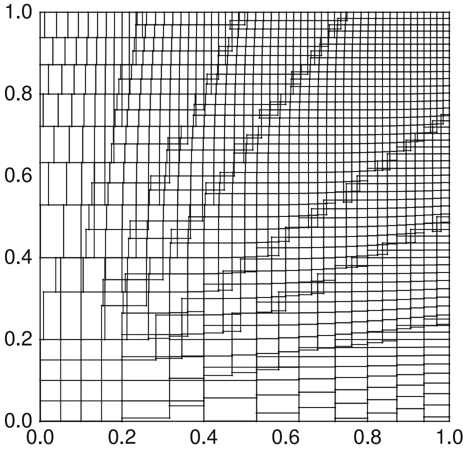


Fig. 4 The essentially optimal construction $\mathscr{R}_\delta$ for $\delta = 0.03$.



choose p too large. Since $T^{(2^p)}$ has the most horizontal stripes of all sectors $T^{(i)}$, namely $\lceil \delta^{-1} \rceil$, a choice satisfying $2^{p(\delta)} = o(\delta^{-1})$ ensures that the overlap has no impact on the coefficient in front the most significant term δ^{-2} in the expansion of $|\mathscr{R}_\delta|$ in terms of δ^{-1}. Figures 3 and 4 show bracketing covers $\mathscr{R}_\delta$ based on this idea constructed in [36] for $\delta = 0.075$ and $\delta = 0.03$. The parameter p was chosen to be

$$p = p(\delta) = \left\lfloor \frac{\ln(\delta^{-1})}{1.7} \right\rfloor .$$

The figures show the overlapping of brackets at the common boundaries of different sectors. Note in particular that the 16 squares near the origin in Fig. 4 are not individual δ-brackets with weight δ—these squares just occur since larger brackets intersect near the origin.

For all technical details of the δ-bracketing cover $\mathscr{R}_\delta$ of $[0,1]^2$ we refer to [36]. As shown there in Proposition 5.1, its size is of order

$$|\mathscr{R}_\delta| = \delta^{-2} + o(\delta^{-2}) \tag{22}$$

as long as $p = p(\delta)$ is a decreasing function on $(0,1)$ with $\lim_{\delta\to 0} p(\delta) = \infty$ and $2^p = o(\delta^{-1})$ as δ tends to zero.

The construction $\mathscr{R}_\delta$ is (essentially) optimal, as will be shown by a lower bound in the next section.

2.3 Bounds for the Bracketing Number

Here we state bounds for the bracketing number for arbitrary dimension d.

Theorem 3. *Let d be a positive integer and $0 < \delta \le 1$. Then we have the following two upper bounds on the bracketing number:*

$$N_{[]}(d,\delta) \le \frac{d^d}{d!}\delta^{-d} + O_d(\delta^{-d+1}) \tag{23}$$

and

$$N_{[]}(d,\delta) \le 2^{d-1}\frac{d^d}{d!}\left(\delta^{-1}+1\right)^d . \tag{24}$$

Both bounds were proved constructively in [35] by a δ-bracketing cover which can be seen as d-dimensional generalization of the two-dimensional construction $\mathscr{Z}_\delta$ from Sect. 2.2.3. In the same paper the following lower bound for the bracketing number was shown, see [35, Theorem 1.5].

Theorem 4. *For $d \ge 2$ and $0 < \delta \le 1$ there exist a constant c_d which may depend on d, but not on δ, with*

$$N_{[]}(d,\delta) \ge \delta^{-d}(1 - c_d\delta). \tag{25}$$

The proof of Theorem 4 is based on the fact that the bracketing number $N_{[]}(d,\delta)$ is bounded from below by the average of $[\lambda_d(B_\delta(x))]^{-1}$ over all $x \in [0,1]^d$, where $B_\delta(x)$ is a δ-bracket containing x with maximum volume.

The lower bound shows that the upper bound $N_{[]}(2,\delta) \le \delta^{-2} + o(\delta^{-2})$, resulting from the bound (22) on the cardinality of $\mathscr{R}_\delta$ from Sect. 2.2.4, is (essentially) optimal.

3 Application of Bracketing to Discrepancy

We want to discuss how the results about bracketing covers and bracketing entropy from the last section can be used to tackle the three questions from Sect. 1.2. We start with question (iii), where our results are most directly applicable.

3.1 Approximation of the Star Discrepancy

Bracketing covers can be used to approximate the star discrepancy by exploiting the following *approximation property*.

Lemma 2. *Let $\mathscr{B}$ be a bracketing cover of* $[0,1]^d$*, and let $\Gamma_{\mathscr{B}}$ as in (13). For finite subsets P of* $[0,1]^d$ *put*

$$d_\Gamma^*(P) := \max_{x\in\Gamma_{\mathscr{B}}} |\Delta(x,P)|. \tag{26}$$

Then we have

$$d_\Gamma^*(P) \le d_\infty^*(P) \le d_\Gamma^*(P) + \delta.$$

The proof is straightforward, but can also be found in, e.g., [21, Lemma 3.1].

The essential idea of Thiémard's algorithm from [87, 88] is to generate for a given point set P and a user-specified error δ a small δ-bracketing cover $\mathscr{B} = \mathscr{B}_\delta$ of $[0,1]^d$ and to approximate $d_\infty^*(P)$ by $\max_{x\in\Gamma_{\mathscr{B}}} |\Delta(x,P)|$.

The costs of generating $\mathscr{B}_\delta$ are of order $\Theta(d|\mathscr{B}_\delta|)$. If we count the number of points in $[0,x)$ for each $x \in \Gamma_{\mathscr{B}}$ in a naive way, this results in an overall running time of $\Theta(dn|\mathscr{B}_\delta|)$ for the whole algorithm. As Thiémard pointed out in [88], this orthogonal range counting can be done in moderate dimension d more effectively by employing data structures based on so-called range trees. This approach reduces the time $O(dn)$ per test box that is needed for the naive counting to $O(\log(n)^d)$. Since a range tree for n points can be generated in $O(C^d n \log(n)^d)$ time, $C > 1$ some constant, this results in an overall running time of

$$O((d + \log(n)^d)|\mathscr{B}_\delta| + C^d n \log(n)^d)\,.$$

For the precise details of the implementation we refer to [88].

The upper bounds on the running time of the algorithm show that smaller δ-bracketing covers $\mathscr{B}_\delta$ will lead to shorter running times. But since the lower bound (25) implies

$$|\mathscr{B}_\delta| \ge \delta^{-d}(1 - c_d\delta),$$

even the time for generating a δ-bracketing cover $\mathscr{B}_\delta$ is bounded from below by $\Omega(d\delta^{-d})$, and this is obviously also a lower bound for the running time of the whole algorithm. This shows that the approach of Thiémard has practical limitations.

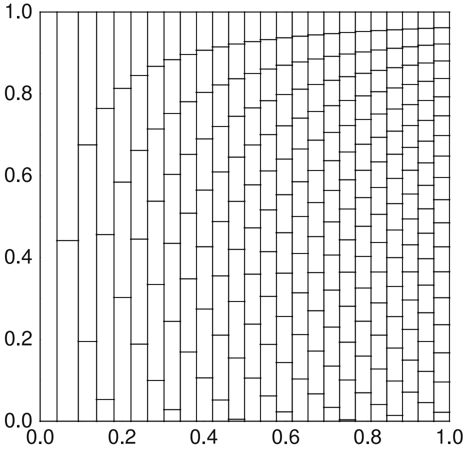


Fig. 5 Thiémard's construction $\mathcal{T}_\delta$ for $\delta = 0.075$.

Nevertheless, it is useful in moderate dimensions as was reported, e.g., in [23] or [70].

The smallest bracketing covers used by Thiémard are different from the constructions we presented in the previous section, see [88]. Figure 5 shows his construction $\mathcal{T}_\delta$ in dimension $d = 2$ for $\delta = 0.075$.

He proved the upper bound

$$|\mathcal{T}_\delta| \leq \binom{d+h}{d}, \quad \text{where } h = \left\lceil \frac{d \ \ln(\delta)}{\ln(1-\delta)} \right\rceil .$$

This leads to

$$|\mathcal{T}_\delta| \leq e^d \left(\frac{\ln \delta^{-1}}{\delta} + 1 \right)^d ,$$

a weaker bound than $|\mathcal{B}_\delta| \leq e^d \delta^{-d} + O_d(\delta^{-d+1})$ and $|\mathcal{B}_\delta| \leq 2^{d-1} e^d (\delta^{-1} + 1)^d$ which hold for the construction $\mathcal{B}_\delta$ that established Theorem 3.

For $d = 2$ the bound $|\mathcal{T}_\delta| = 2\ln(2)\delta^{-2} + O(\delta^{-1})$ was proved in [36], which shows that in two dimensions the quality of $\mathcal{T}_\delta$ is similar to the one of the layer construction $\mathcal{Z}_\delta$ that we presented in the Sect. 2.2.3.

3.2 *Pre-Asymptotic Bounds via Randomization*

Here we want discuss question (i) from Sect. 1.2. We distinguish between deterministic discrepancy bounds for n-point samples in $[0,1]^d$ and for d-dimensional projections of infinite sequences of points with infinitely many coordinates. Furthermore, we mention briefly probabilistic discrepancy bounds for hybrid-Monte Carlo sequences.

3.2.1 Point Sets in the d-Dimensional Unit Cube

Probabilistic pre-asymptotic bounds on the smallest possible star discrepancy of any n-point set in $[0,1]^d$ can be proved in three steps:

Probabilistic Proof Scheme:

1. We discretize the star discrepancy at the cost of an approximation error at most δ. More precisely, we use a δ-bracketing cover $\mathscr{B}$ and consider for a point set P instead of $d_\infty^*(P)$ its approximation $d_\Gamma^*(P)$ defined in (26), where $\Gamma = \Gamma_{\mathscr{B}}$ is as in (13).
2. We perform a random experiment that results in a random n-point set P in $[0,1]^d$ that fails to satisfy the events $\{|\Delta(x,P)| \le \delta\}$, $x \in \Gamma_{\mathscr{B}}$, with small probability. If the random experiment is subject to the concentration of measure phenomenon, then these "failing probabilities" can be controlled with the help of large deviation bounds.
3. Since the event $\{d_\Gamma^*(P) > \delta\}$ is the union of the events $\{|\Delta(x,P)| > \delta\}, x \in \Gamma_{\mathscr{B}}$, a simple union bound shows that P satisfies $d_\Gamma^*(P) \le \delta$ with positive probability if $\mathbb{P}\{|\Delta(x,P)| > \delta\} < |\Gamma_{\mathscr{B}}|^{-1}$ for all $x \in \Gamma_{\mathscr{B}}$.

Then for $\varepsilon = 2\delta$ there exists an n-point set P with $d_\infty^*(P) \le d_\Gamma^*(P) + \delta \le \varepsilon$. The aim is to choose ε as small as possible.

To keep the loss caused by the union bound small, the size of the δ-bracketing cover $\mathscr{B}$ (or the δ-cover $\Gamma_{\mathscr{B}}$, respectively) should be chosen as small as possible. To receive a bound for the star discrepancy with explicit constants, bounds with explicit constants are needed for the size of the δ-bracketing cover used.

The bound (10) from [45] was proved in this way: The δ-cover Γ was chosen to be the equidistant grid from Sect. 2.2.1 and the random experiment was to distribute n points uniformly and independently in $[0,1]^d$. The "failing probability" in each single test box was bounded above by *Hoeffding's large deviation bound* [52], which reads as follows:

Let $X_1, \ldots, X_n$ be independent random variables with $a_i \le X_i \le b_i$ for all i. Then for all $\delta > 0$

$$\mathbb{P}\left\{ \frac{1}{n} \left| \sum_{k=1}^{n} (X_i - \mathbb{E}(X_i)) \right| \ge \delta \right\} \le 2 \exp\left(- \frac{2\delta^2 n^2}{\sum_{k=1}^{n} (b_i - a_i)^2} \right).$$

Using the same probabilistic experiment and again Hoeffding's large deviation bound, but instead of the bracketing cover from Sect. 2.2.1 the one that implied the estimate (24), one obtains the improved discrepancy bound

$$d_\infty^*(n,d) \le k' d^{1/2} n^{-1/2} \ln\left(1 + \frac{n}{d}\right)^{1/2} \tag{27}$$

(here we have essentially $k' \approx \sqrt{2}$, see [35, Theorem 2.1]). Since the inverse of the star discrepancy depends linearly on the dimension d, the practically most relevant choice of n seems to be n proportional to d. Note that in this case (27) behaves asymptotically as the bound (8). In fact, if (8) holds with $c = 10$ (as claimed by Hinrichs and recently published by Aistleitner), then the bound [35, (22)], a version of (27), is still better than (8) for all $n \leq 1.5 \cdot \mathrm{e}^{95} d$. Actually, we may use the upper bound in (24) to reprove (8) without using Haussler's result on covering numbers of Vapnik-Červonenkis classes—a version of Talagrand's large deviation bound for empirical processes holds under the condition that the δ-bracketing number of the set system under consideration is bounded from above by $(C\delta^{-1})^d$ for some constant C not depending on δ or d, see [86, Theorem 1.1]. (As we discuss at the end of this subsection, Aistleitner's approach to prove (8) with a constant $c \leq 10$ indeed uses the upper bound (24).)

For other discrepancy notions similar approaches, relying on uniformly and independently distributed random points, were used to prove pre-asymptotic bounds with explicitly given constants. This was done, e.g., for the *same-quadrant discrepancy* [47], discrepancies with respect to *ellipsoids*, *stripes*, and *spherical caps* in $\mathbb{R}^d$ [59], the *extreme discrepancy* [35], and the *weighted star discrepancy* [50].

One can modify the probabilistic experiment by using, e.g., the variance reduction technique *stratified sampling*. If, e.g., $n = \nu^d$, then one can subdivide $[0,1]^d$ into n subcubes of the same size and distribute in each subcube one point uniformly at random (and independently from the other points). This experiment was used in [20, Theorem 4.3] (a preprint version of [21]) to derive

$$d_\infty^*(n,d) \leq k'' d n^{-\frac{1}{2}-\frac{1}{2d}} \ln(n)^{1/2}. \tag{28}$$

(Again, we have essentially $k'' \approx \sqrt{2}$. The proof used the δ-cover Γ_δ from (16).)

For the discrepancy of *tilted boxes* and of *balls* with respect to probability measures on $[0,1]^d$ which are absolutely continuous with respect to λ_d, a similar approach relying on a stratified sampling technique was used by Beck in [3] to prove asymptotic probabilistic upper bounds. But these bounds do not exhibit the dependence on the dimension; in particular, the involved constants are not explicitly known.

We will discuss a further random experiment in more detail in Sect. 3.3.

Let us finish this subsection with the discussion of the recent result of Aistleitner, who proved in [1] that the constant c in (8) is smaller than 10. As in the probabilistic proof scheme stated above, his approach starts by discretizing the star discrepancy at the cost of an approximation error $\delta = 2^{-K}$, where $K \approx -\log_2(d/n)/2$. The underlying probabilistic experiment is to distribute n points $p^{(1)}, \ldots, p^{(n)}$ uniformly and independently in $[0,1]^d$. An important observation is now that for measurable subsets A of $[0,1]^d$ the variance of the random variables $\xi_A^{(i)} := \lambda_d(A) - |\{p^{(i)}\} \cap A|$, $i = 1, \ldots, n$, depends strongly on the volume $\lambda_d(A)$ of A:

$$\mathrm{Var}(\xi_A^{(i)}) = \lambda_d(A)(1-\lambda_d(A)).$$

Now Hoeffding's large deviation bound gives good bounds for the failing probabilities $\mathbb{P}\{|\frac{1}{n}\sum_{i=1}^{n}\xi_A^{(i)}| > \delta_A\}$ for $\delta_A > 0$ if $\lambda_d(A) \approx 1/2$. But if $\lambda_d(A)$ is much smaller or larger than $1/2$, then Hoeffding's bound cannot exploit the fact that the variance of the random variable $\xi_A^{(i)}$ is small. A large deviation bound which can exploit this fact is *Bernstein's inequality* which reads as follows (see, e.g., [90]):

Let $X_1, \dots, X_n$ be independent random variables with zero means and bounded ranges $|X_i| \leq M$ for all i. Then for all $t > 0$

$$\mathbb{P}\left\{\left|\sum_{k=1}^{n} X_i\right| \geq t\right\} \leq 2\exp\left(-\frac{t^2/2}{\sum_{k=1}^{n}\operatorname{Var}(X_i) + Mt/3}\right).$$

Aistleitner uses Bernstein's inequality and the *dyadic chaining* technique, which can be seen as a "multi-cover" approach:

For all $k = 1, 2, \dots, K$ consider a 2^{-k}-cover $\Gamma_{2^{-k}}$, and put $x^{(0)} := 0$. From the definition of a δ-cover it follows that for any $x^{(K)} \in \Gamma_{2^{-K}}$ one recursively finds points $x^{(k)} \in \Gamma_{2^{-k}}$, $k = K-1, \dots, 1$, such that $x_j^{(K)} \geq x_j^{(K-1)} \geq \dots \geq x_j^{(1)}$ for $j = 1, \dots, d$, and

$$A_k = A_k\left(x^{(K)}\right) := \left[0, x^{(k)}\right) \setminus \left[0, x^{(k-1)}\right)$$

has volume at most $2^{-(k-1)}$. We have $\left[0, x^{(K)}\right) = \cup_{k=1}^{K} A_k$ and, if P denotes the set $\{p^{(1)}, \dots, p^{(n)}\}$,

$$\left|\Delta\left(x^{(K)}, P\right)\right| \leq \sum_{k=1}^{K}\left|\lambda_d\left(A_k\right) - \frac{1}{n}\left|P \cap A_k\right|\right| = \sum_{k=1}^{K}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_{A_k}^{(i)}\right|.$$

If for $k = 1, \dots, K$ we define $\mathscr{A}_k := \left\{A_k\left(x^{(K)}\right) | x^{(K)} \in \Gamma_{2^{-K}}\right\}$, then $|\mathscr{A}_k| \leq |\Gamma_{2^{-k}}|$. Using a 2^{-k}-bracketing cover as constructed in [35], we obtain via (13) a 2^{-k}-cover $\Gamma_{2^{-k}}$ satisfying $|\Gamma_{2^{-k}}| \leq (2e)^d(2^k+1)^d$, see (24) and (14). Choosing a suitable sequence c_k, $k = 1, \dots, K$, one essentially obtains with the help of a union bound, Bernstein's inequality, and the estimate (24)

$$\mathbb{P}\left(\bigcup_{A_k \in \mathscr{A}_k}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}\xi_{A_k}^{(i)}\right| > c_k 2^{-K}\right\}\right) \leq \sum_{A_k \in \mathscr{A}_k}\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}\xi_{A_k}^{(i)}\right| > c_k 2^{-K}\right\} \leq 2^{-k}.$$

Recall that $|\mathscr{A}_k| \leq |\Gamma_{2^{-k}}| \leq O_d(2^{kd})$ and $\operatorname{Var}(\xi_{A_k}^{(i)}) \leq 2^{-(k-1)}$. In particular, $|\mathscr{A}_K|$ is of the size of the finest δ-cover $\Gamma_{2^{-K}}$, but, since the variance of all $\xi_{A_K}^{(i)}$ is small (namely at most $2^{-(K-1)}$), Bernstein's inequality ensures that we can choose a small c_K. If, on the other hand, $k = 1$, then it may happen that $\lambda_d(A_1) \approx 1/2$, so Bernstein's inequality gives us no advantage over Hoeffding's bound. But the size of $\mathscr{A}_1$ is relatively small, namely at most $O_d(2^d)$. In general, the larger k is, the more we can exploit the small variance of all $\xi^{(i)}_{A_k}$, but the larger is the size of $\mathscr{A}_k$. Aistleitner proved that this “trade off” ensures that one can choose $(c_k)_{k=1}^K$ such that $\sum_{k=1}^K c_k \leq 8.65$ holds. Thus the approximation property (see Lemma 2) leads to the estimate

$$\begin{aligned}
\mathbb{P}\left\{ d^*_\infty(P) > \left(1+\sum_{k=1}^K c_k\right)2^{-K}\right\} &\leq \mathbb{P}\left\{ d^*_{\Gamma_{2^{-K}}}(P) > \sum_{k=1}^K c_k 2^{-K}\right\}\\
&= \mathbb{P}\left(\bigcup_{x^{(K)}\in\Gamma_{2^{-K}}} \left\{ \left|\Delta\left(x^{(K)},P\right)\right| > \sum_{k=1}^K c_k 2^{-K}\right\}\right)\\
&\leq \mathbb{P}\left(\bigcup_{x^{(K)}\in\Gamma_{2^{-K}}} \bigcup_{k=1}^K \left\{ \left|\frac{1}{n}\sum_{i=1}^n \xi^{(i)}_{A_k}\right| > c_k 2^{-K}\right\}\right)\\
&\leq \sum_{k=1}^K \mathbb{P}\left(\bigcup_{A_k\in\mathscr{A}_k} \left\{ \left|\frac{1}{n}\sum_{i=1}^n \xi^{(i)}_{A_k}\right| > c_k 2^{-K}\right\}\right) < 1,
\end{aligned}$$

showing that there exists an n-point set P in $[0,1]^d$ that satisfies the estimate (8) with $c = 9.65$. (For the technical details we refer, of course, to [1].)

3.2.2 Infinite Dimensional Infinite Sequences

So far we have discussed the existence of point sets that satisfy reasonably good discrepancy bounds. In practice it is desirable to have integration points that can be extended in the number of points, and preferably also in the dimension d. This allows to achieve higher approximation accuracy while still being able to reuse earlier calculations.

In [14] the probabilistic bounds stated in the previous subsection were extended by Dick to infinite sequences of infinite dimensional points. For an infinite sequence P of points in $[0,1)^{\mathbb{N}}$, let us denote by P_d the sequence of the projections of the points of P onto their first d components, and by $P_{n,d}$ the first n points of P_d. Then in [14] the following results were shown:

There exists an unknown constant C such that for every strictly increasing sequence $(n_m)_{m\in\mathbb{N}}$ in $\mathbb{N}$ there is an infinite sequence P in $[0,1)^{\mathbb{N}}$ satisfying

$$d^*_\infty(P_{n_m,d}) \leq C\sqrt{d/n_m}\sqrt{\ln(m+1)} \quad \text{for all } m,d\in\mathbb{N}.$$

(We add here that with the help of Aistleitner’s approach in [1] one can derive an upper bound for C.)

Furthermore, there exists an explicitly given constant C' such that for every strictly increasing sequence $(n_m)_{m\in\mathbb{N}}$ in $\mathbb{N}$ there is an infinite sequence P satisfying

$$d_\infty^*(P_{n_m,d}) \le C' \sqrt{\left(m + d + d \ln\left(1 + \frac{d\sqrt{n_m}}{m+d}\right)\right)/n_m} \quad \text{for all } m, d \in \mathbb{N}. \tag{29}$$

The results from [14] show that there exist point sets that can be extended in the dimension and in the number of points while bounds similar to (10) or (27) remain valid.

A disadvantage of (29) is nevertheless that in the case where, e.g., $n_m = m$ for all m it is not better than the trivial bound $d_\infty^*(P_{m,d}) \le 1$.

By using the bound (24), another result for infinite sequences P in $[0,1)^{\mathbb{N}}$ was presented in [19]: There exists an explicitly given constant C'' such that for every strictly increasing sequence $(n_m)_{m\in\mathbb{N}}$ in $\mathbb{N}$ there is an infinite sequence P satisfying

$$d_\infty^*(P_{n_m,d}) \le C'' \sqrt{d \ln\left(1 + \frac{n_m}{d}\right)/n_m} \quad \text{for all } m, d \in \mathbb{N}. \tag{30}$$

This bound is an improvement of (29), which in particular is still useful in the case $n_m = m$ for all m. Moreover, it establishes the existence of infinite sequences P in $[0,1)^{\mathbb{N}}$ having the following property: To guarantee $d_\infty^*(P_{n,d}) \le \varepsilon$ for a given ε, we only have to take $n \ge c_\varepsilon d$, where c_ε is a constant depending only on ε, see [19, Corollory 2.4]. Note that this result cannot be deduced directly from the results in [14]. As mentioned above, it is known from [45, 49] that we have to take at least $n \ge c'_\varepsilon d$ if ε is sufficiently small. (Here c'_ε depends again only on ε.) In this sense [19, Corollory 2.4] shows that the statement "the inverse of the star discrepancy depends linearly on the dimension" (which is the title of the paper [45]) extends to the projections of infinite sequences in $[0,1)^{\mathbb{N}}$. To make this more precise, the notion of the *inverse of the star discrepancy of an infinite sequence* P is introduced in [19], given by

$$N_P^*(\varepsilon, d) := \min\{n : \forall m \ge n : d_\infty^*(P_{m,d}) \le \varepsilon\}.$$

Then Corollary 2.4 of [19] states that there exist sequences P such that

$$N_P^*(\varepsilon, d) \le O(d\varepsilon^{-2} \ln(1 + \varepsilon^{-1})) \quad \text{for all } d \in \mathbb{N}, \varepsilon \in (0,1]. \tag{31}$$

In fact even more holds: If we endow the set $[0,1)^{\mathbb{N}}$ with the canonical probability measure $\lambda_{\mathbb{N}} = \otimes_{i=1}^\infty \lambda_1$ and allow the implicit constant in the big-O-notation to depend on the particular sequence P, then inequality (31) holds almost surely for a random sequence P, see again [19, Corollory 2.4]. In [19, Theorem 2.3] bounds of the form (30) and (31) with explicitly given constants and estimates for the measure of the sets of sequences satisfying such bounds are provided.

3.2.3 Hybrid-Monte Carlo Sequences

A *hybrid-Monte Carlo sequence*, which is sometimes also called a *mixed sequence*, results from extending a low-discrepancy sequence in the dimension by choosing the additional coordinates randomly. In several applications it has been observed that hybrid-Monte Carlo sequences perform better than pure Monte Carlo and pure quasi-Monte Carlo sequences, especially in difficult problems, see, e.g., [69,71,83].

For a mixed d-dimensional sequences m, whose elements are, technically speaking, vectors obtained by concatenating the d'-dimensional vectors from a low-discrepancy sequence q with $(d - d')$-dimensional random vectors, probabilistic upper bounds for its star discrepancy have been provided. If m_n and q_n denote the sets of the first n points of the sequences m and q respectively, then Ökten et al. showed in [71] that

$$\mathbb{P}(d_\infty^*(m_n) - d_\infty^*(q_n) < \varepsilon) \geq 1 - 2\exp\left(-\frac{\varepsilon^2 n}{2}\right) \quad \text{for } n \text{ sufficiently large.} \tag{32}$$

The authors did not study how large n actually has to be and if and how this actually depends on the parameters d and ε. In the note [37] a lower bound for n is derived, which significantly depends on d and ε. Furthermore, with the help of the probabilistic proof scheme the probabilistic bound

$$\mathbb{P}(d_\infty^*(m_n) - d_\infty^*(q_n) < \varepsilon) > 1 - 2N(d, \varepsilon/2)\exp\left(-\frac{\varepsilon^2 n}{2}\right) \tag{33}$$

was established, which holds without any restriction on n. In this sense it improves the bound (32) and is more helpful in practice, especially for small samples sizes n. As we know from (25) and (14), for small ε the quantity $N(d, \varepsilon/2)$ grows exponentially in d. As pointed out in [37, Remark 3.4] a factor depending exponentially on d has to appear in front of $\exp(-\varepsilon^2 n/2)$ in the bound (33) if we want it to hold for all $n \in \mathbb{N}$. Recall that we can use the bound (24) on the bracketing number to obtain an upper bound for $N(d, \varepsilon/2)$ with explicit constants.

Recently, there has been increasing interest in (deterministic) discrepancy bounds for (deterministic) mixed sequences, see, e.g., [63, 64].

3.3 Small Discrepancy Samples via Derandomization

Here we want to consider question (ii) from Sect. 1.2: How can we construct point sets that satisfy the probabilistic bounds stated in Sect. 3.2? How can we derandomize the probabilistic experiments to get deterministic point sets with low discrepancy? The probabilistic experiment of distributing n points uniformly at random in $[0, 1]^d$ was derandomized in [21]. We illustrate the derandomization idea for a different probabilistic experiment used in [23], which leads to a simpler and faster algorithm.

3.3.1 Random Experiment

Let $k \in \mathbb{N}$ be given and let δ be the largest value that satisfies $k = \kappa(\delta, d)$, where $\kappa(\delta, d)$ is as in (18). Let $\Gamma = \Gamma_\delta$ be the non-equidistant grid from (16). Put $\gamma_{k+1} := 0$ and let $\mathscr{B} = \mathscr{B}_\delta$ the set of all (half-open) grid cells, i.e., all boxes $[y, y^+)$ with $y_i = \gamma_j$ for some $j = j(i) \in \{1, \dots, k+1\}$ and $y_i^+ = \gamma_{j-1}$ for all $i \in d$. Then obviously $|\Gamma| = |\mathscr{B}|$.

Let $n \in \mathbb{N}$ be given. For $B \in \mathscr{B}$ let $x_B := n \cdot \lambda_d(B)$, i.e., x_B is the expected number of points inside B if we distribute n points independently at random in $[0,1]^d$.

Our aim is now to *round randomly* for each $B \in \mathscr{B}$ the real number x_B to an integer y_B such that the following two constraints are satisfied:

- *Weak constraint*: Each set Y with y_B points in B for all $B \in \mathscr{B}$ should have small discrepancy with high probability.
- *Hard constraint*: The equation $|Y| = \sum_{B \in \mathscr{B}} y_B = \sum_{B \in \mathscr{B}} x_B = n$ should hold.

We saw in Sect. 3.2 that in the previous random experiments the weak constraint can be satisfied for independent random points with the help of large deviation inequalities. But if our rounding procedure has to satisfy the hard constraint our random variables y_B, $B \in \mathscr{B}$, are clearly not independent any more.

Nevertheless, such a randomized rounding that satisfies the weak constraint with high probability and respects the hard constraint can be done. There are two approaches known, due to Srinivasan [84] and to Doerr [18]. We present here the randomized rounding procedure of Srinivasan:

Randomized Rounding Procedure:

- Initialize $y_B = x_B$ for all $B \in \mathscr{B}$.
- Repeat the following step until all y_B are integral:

  Pair Rounding Step: Choose y_B, $y_{B'}$ not integral.
  Choose $\sigma \in [0,1]$ minimal such that $y_B + \sigma$ or $y_{B'} - \sigma$ is integral.
  Choose $\tau \in [0,1]$ minimal such that $y_B - \tau$ or $y_{B'} + \tau$ is integral.
  Set

$$(y_B, y_{B'}) := \begin{cases} (y_B + \sigma, y_{B'} - \sigma) & \text{with probability } \frac{\tau}{\sigma+\tau}, \\ (y_B - \tau, y_{B'} + \tau) & \text{with probability } \frac{\sigma}{\sigma+\tau}. \end{cases}$$

- Output: $(y_B)_{B \in \mathscr{B}}$.

The pair rounding step leaves $\sum_{B \in \mathscr{B}} y_B$ invariant. Hence we have always

$$\sum_{B \in \mathscr{B}} y_B = \sum_{B \in \mathscr{B}} x_B = n.$$

This shows particularly that if there is a variable y_B left which is not integral, there has to be another one $y_{B'}$, $B \neq B'$, which is not integral. Thus the algorithm terminates and the output set y_B, $B \in \mathscr{B}$, satisfies the hard constraint. Furthermore, the pair rounding step leaves $\mathbb{E}(y_B)$ invariant, hence $\mathbb{E}(y_B) = x_B$. Now let Y be a set with y_B points in B for all $B \in \mathscr{B}$. Then

$$\mathbb{E}(n\Delta(g,Y)) = \mathbb{E}\Bigg(\sum_{B\in\mathscr{B}\,;\,B\subseteq[0,g)} (x_B - y_B)\Bigg) = 0 \qquad \text{for all } g \in \Gamma.$$

Furthermore, a concentration of measure result holds. The y_B, $B \in \mathscr{B}$, are not independent, but it can be shown that they satisfy certain negative correlation properties, cf. [84]. As shown by Panconesi and Srinivasan, Chernoff-Hoeffding-type bounds hold also in this situation [74]. This result and the earlier observations yield the following theorem, see [23].

Theorem 5. *The randomized rounding procedure generates in time $O(|\mathscr{B}|)$ randomized roundings y_B of x_B for all $B \in \mathscr{B}$ such that $\sum_{B\in\mathscr{B}} y_B = \sum_{B\in\mathscr{B}} x_B = n$ and*

$$\mathbb{P}\{|\Delta(g,Y)| > \lambda\} < 2\exp\left(-\frac{\lambda^2 n}{3}\right) \qquad \textit{for all } g \in \Gamma.$$

If we now choose $\lambda = \sqrt{3n^{-1}\ln(2|\Gamma|)}$ and $\delta \approx \sqrt{d/n}\sqrt{\ln\ln(d)}$, then the next theorem can be proved by following the three steps of the proof sheme in Sect. 3.2.1, see [23].

Theorem 6. *There exists a constant $C > 0$ such that*

$$\mathbb{P}\left\{d^*_\infty(Y) \le C\sqrt{d/n}\sqrt{\ln(\sigma n)}\right\} > 0, \tag{34}$$

where $\sigma = \sigma(d) < 1.03$ tends to zero if $d \to \infty$.

(Essentially we have $C \approx \sqrt{6}$.)

3.3.2 Derandomized Construction

Now we want to derandomize the random experiment, i.e., we want to construct an n-point set Y deterministically that satisfies the bound (34) in Theorem 6. More precisely, we want to compute a rounding $(y_B)_{B\in\mathscr{B}}$ of $(x_B)_{B\in\mathscr{B}}$ that satisfies

$$\sum_{B\in\mathscr{B}} y_B = \sum_{B\in\mathscr{B}} x_B = n \tag{35}$$

and

$$\left|\sum_{B\subseteq[0,g)} (x_B - y_B)\right| \le \delta_g \cdot n \cdot \lambda_d([0,g)) \qquad \text{for all } g \in \Gamma, \tag{36}$$

where the δ_gs are error tolerances fixed in the algorithm. If then Y is a set with y_B points in B for all $B \in \mathscr{B}$, we obtain $|Y| = n$ and

$$\left|\lambda_d([0,g)) - \frac{1}{n}|Y \cap [0,g)|\right| \le \delta_g \cdot \lambda_d([0,g)) \quad \text{for all } g \in \Gamma.$$

To compute such a rounding we follow Raghavan [75] and define *pessimistic estimators* P_g^+, P_g^-, $g \in \Gamma$. For $B \in \mathscr{B}$ let $p_B = \{x_B\}$, where $\{x_B\}$ denotes the fractional part of x_B, and for $g \in \Gamma$ let $\mu_g := \sum_{B \subseteq [0,g)} \{x_B\}$. The pessimistic estimators are defined as

$$P_g^+ = (1+\delta_g)^{-(1+\delta_g)\mu_g} \prod_{B \subseteq [0,g)} \left(1 + \delta_g p_B\right)$$

$$P_g^- = (1+\delta_g)^{(1-\delta_g)\mu_g} \prod_{B \subseteq [0,g)} \left(1 + \left(\frac{1}{1+\delta_g} - 1\right) p_B\right).$$

With the help of the pessimistic estimators we can see whether (36) is satisfied or not. This is easily seen by making the following observation: For $B \in \mathscr{B}$ let $q_B \in \{0,1\}$, and for $q \in \Gamma$ let Q_g^+, Q_g^- be the values of P_g^+ and P_g^-, respectively, calculated on values q_B instead of p_B (with μ_g unchanged). Then it is a simple observation that $Q_g^+ \ge 1$ if and only if $\sum_{B \subseteq [0,g)} q_B \ge (1+\delta_g)\mu_g$, and $Q_g^- \ge 1$ if and only if $\sum_{B \subseteq [0,g)} q_B \le (1-\delta_g)\mu_g$.

By *updating* the pessimistic estimators for some adjustment $p_B \leftarrow x$, we shall mean the operation of replacing the factor $(1+\delta_g p_B)$ in P_g^+ by $(1+\delta_g x)$, and analogously for P_g^-, for each $g \in \Gamma$ such that $B \subseteq [0,g)$. (Again, μ_g stays unchanged.)

The derandomized rounding algorithm proceeds as follows.

Derandomized Rounding Procedure:

1. Initialize $p_B := \{x_B\}$ for all $B \in \mathscr{B}$.
2. Set the error tolerances δ_g such that for each $g \in \Gamma$ we have $P_g^+, P_g^- < 1/(2|\Gamma|)$. Let $U := \sum_{g \in \Gamma}(P_g^+ + P_g^-)$.
3. Let $\mathscr{J} = \{B \in \mathscr{B} \mid p_B \notin \{0,1\}\}$. While $|\mathscr{J}| \ge 2$:
   (a) Pick $B, B' \in \mathscr{J}$.
   (b) Let $(p_B^{(i)}, p_{B'}^{(i)})$, $i = 1,2$, be the two possible outcomes of the pair-rounding step of the randomized rounding procedure with respect to the pair of variables $(p_B, p_{B'})$. Let U_i, $i = 1,2$, be the sum of the pessimistic estimators U updated according to the corresponding outcome.
   (c) Pick $i \in \{1,2\}$ to minimize U_i. Let $p_B \leftarrow p_B^{(i)}$, $p_{B'} \leftarrow p_{B'}^{(i)}$ and update $\mathscr{J}$, the pessimistic estimators, and U.
4. Output: $y_B = \lfloor x_B \rfloor + p_B$, $B \in \mathscr{B}$.

Note that in step 2 we have $U < 1$. Furthermore, it was shown in [24, Sect. 3.1] that the minimum U_i of $\{U_1, U_2\}$ appearing in step 3.c satisfies $U_i \le U$. After step 3 we have $\mathscr{J} = \emptyset$ and $p_B \in \{0,1\}$ for every $B \in \mathscr{B}$. By our previous observation, $\sum_{B \subseteq [0,g)} p_B \ge (1+\delta_g)\mu_g$ if and only if $P_g^+ \ge 1$, and analogously for the lower bound. Since $U < 1$ is maintained throughout the algorithm and since the pessimistic estimators are non-negative, this cannot occur. The process thus produces a rounding satisfying equation (36). Note that as in the randomized rounding, the value of $\sum_{B \in \mathscr{B}} p_B$ is kept constant throughout the process, thus (35) is satisfied.

Although the order in which variables are picked in step 3.a is not important for the theoretical bound, numerical tests indicate that it is preferable to use an order in which the tree formed by the pairings is a balanced binary tree (so that each value p_B is adjusted only $O(\log |\Gamma|)$ times), see [24] for details.

Using the bounds on the δ_gs derived by Raghavan [75] and choosing δ of order $\delta \approx \sqrt{d/n}\sqrt{\ln \ln(d)}$, the derandomized rounding algorithm leads to the following theorem, see [23].

Theorem 7. *There exists a deterministic algorithm which, on input n and d, computes in time $O(d \ln(dn)(\sigma n)^d)$ an n-point set $Y \subset [0,1]^d$ with discrepancy*

$$d_\infty^*(Y) \leq C \sqrt{d/n} \sqrt{\ln(\sigma n)};$$

here $C < 2.44$, and $\sigma = \sigma(d) < 1.03$ tends to zero if $d \to \infty$.

The output set Y has y_B points in each grid cell $B \in \mathscr{B}$. Although the exact placement of these points inside the boxes B does not affect the theoretical bound on $d_\infty^*(Y)$ from Theorem 7, numerical experiments indicate that it is a good idea to place these points independently, uniformly at random in B.

3.3.3 A Component-by-Component Derandomization

Another approach is presented in [19]. There a component-by-component (CBC) construction of n-point sets via derandomization is proposed. In particular, via this approach given point sets can be extended in the dimension. Here the underlying random experiment is as follows: Given an n-point set $P_{d'} = \{p^{(1)}, \ldots, p^{(n)}\}$ in dimension d', we choose new components $x^{(1)}, \ldots, x^{(n)}$ randomly from some one-dimensional grid and receive the n-point set $P_{d'+1} = \{(p^{(1)}, x^{(1)}), \ldots, (p^{(n)}, x^{(n)})\}$. We may repeat this procedure until we obtain an n-point set in the desired dimension d. This probabilistic experiment can be derandomized with the classical method of Raghavan [75]. If we start the CBC-construction in dimension one, the deterministic output set P_d of size n in dimension d satisfies the bound

$$d_\infty^*(P_d) \leq O(d^{3/2} n^{-1/2} \ln(1 + n/d)^{1/2}). \tag{37}$$

and the running time of the algorithm is bounded by

$$O(c^d n^{(d+3)/2} (d \ln(1 + n/d))^{-(d+1)/2}),$$

c a suitable constant independent of n and d. Certainly the bound (37) is weaker than the bound in Theorem 7, but the bound on the running time of the CBC algorithm is a reasonable improvement upon the running time guarantee of the derandomized algorithm discussed before. The CBC-algorithm has the additional nice feature that it can calculate the exact discrepancy of the output set without essentially more effort.

In [22] some more implementation details of the CBC-algorithm are provided and several numerical tests are performed. In particular, the experiments indicate that the discrepancies of the output sets of the CBC-algorithm behave in practice much better than predicted by the theoretical bound (37). They depend rather linear on the dimension d than proportional to $d^{3/2}$. The numerical experiments reveal that the discrepancies of the output sets, which are subsets of certain full d-dimensional grids, are almost exactly equal to the discrepancies of the full grids (for reasons explained in [22] we want to call the latter discrepancies "grid gaps"). For output sets of size n the corresponding full grid has size larger than $n^{d/2}/d!$. We may interpret this result in a positive way: The CBC-algorithm provides a sparse sample from a complete d-dimensional grid, which exhibits essentially the same discrepancy as the full grid.

To overcome the lower bound on the discrepancy given by the "grid gap", we also consider a randomized CBC-variant: After receiving an output set P_d, we randomize its points locally to receive a new output set P_d^*. For the randomized set P_d^* the theoretical discrepancy bound (37) still holds, and in all the numerical tests in dimension $d = 10$ its discrepancy was always much smaller than the corresponding grid gap (which, as already said, is a lower bound for $d_\infty^*(P_d)$). (To be more precise, an estimator for $d_\infty^*(P_d^*)$, which majorizes $d_\infty^*(P_d^*)$ with certainty at least 95%, is always much smaller than the corresponding grid gap. We use this estimator, since calculating the actual discrepancy of P_d^* is a much harder problem than calculating the discrepancy of P_d.)

The star discrepancy of the output sets of both derandomized algorithms we presented here was compared in [23] to the star discrepancy of other low discrepancy point sets. These experiments took place in dimensions from 5 to 21 and indicate that the first derandomized algorithm leads to superior results if the dimension is relatively high and the number of points is rather small. (We use the phrase "indicate", since for dimension 10 or more, we are not able to calculate the exact discrepancy, but can only use upper and lower bounds on it.) For details see [23].

4 Conclusion and Open Problems

In the previous sections we discussed questions (i), (ii), and (iii) and described in particular how approaches based on bracketing entropy, randomization, and derandomization lead to improvements on previously achieved results.

The discussion shows that good bounds for the star discrepancy with explicitly known constants are available. Similar bounds hold also for the star discrepancy of point sets that are extensible in the number of points and in the dimension, and the statement that the inverse of the star discrepancy depends linearly on the dimension d [45] can be extended to this situation: The inverse of the star discrepancy of infinite sequences in $[0,1)^{\mathbb{N}}$ depends almost surely linearly on the dimension d.

Can we find even better bounds than (27) or (8)? A lower bound for the star discrepancy that follows directly from (9) is of the form $d_\infty^*(n,d) \geq \min\{\varepsilon_0, c_0 d n^{-1}\}$, c_0 and ε_0 suitable constants [49, Theorem 1], and leaves some room for improvements of (27) or (8). Also the bound (28) shows that some trade-off between the dependence on the number of points and on the dimension is possible. But instead of agonizing over this intriguing question, let us state the *conjecture of Woźniakowski* (see [44], or [66, Open Problem 7]): *If there exist constants* $C, \alpha > 0$ *and a polynomial* p *such that*

$$d_\infty^*(n,d) \leq C\, p(d) n^{-\alpha} \quad \textit{for all } d, n \in \mathbb{N}, \tag{38}$$

then necessarily $\alpha \leq 1/2$.

The construction of point sets satisfying bounds like (8) or (27) can be done with the help of derandomized algorithms [19, 21–23]. Unfortunately, these algorithms exhibit running times that are exponential with respect to the dimension d, a fact prohibiting their use in really high dimensions.

This is maybe not too surprising, since even the seemingly easier problem of calculating the star discrepancy of an arbitrary point set (or approximating it up to a user-specified error) can only be solved in exponential time in d so far. And indeed the problem of calculating the star discrepancy is known to be *NP*-hard.

Nevertheless, the discussed derandomized algorithms can be used in low and modestly high dimension d.

In light of the discussion above, it would be of interest to make further progress in designing algorithms that construct low-discrepancy point sets of small size and algorithms that approximate the star discrepancy of arbitrary n-point sets (which would allow "semi-constructions" as described above). Furthermore, it would be interesting to learn more about the dependence of the star discrepancy of classical constructions on the dimension d and the complexity of approximating the star discrepancy of given point sets.

Acknowledgements The author would like to thank two anonymous referees for their helpful suggestions.

Part of the work on this book chapter was done while the author was research fellow at the Department of Computer Science of Columbia University in the City of New York.

He gratefully acknowledges support from the German Science Foundation DFG under grant GN91/3-1 and GN91/4-1.

References

1. Aistleitner, C.: Covering numbers, dyadic chaining and discrepancy. J. Complexity **27**, 531–540 (2011)
2. Atanassov, E.: On the discrepancy of the Halton sequences. Math. Balkanica (N.S.) **18**, 15–32 (2004)
3. Beck, J.: Some upper bounds in the theory of irregularities of distribution. Acta Arith. **43**, 115–130 (1984)
4. Beck, J.: A two-dimensional van Aardenne-Ehrenfest theorem in irregularities of distribution. Composito Math. **72**, 269–339 (1989)
5. Beck, J., Chen, W.W.L.: Irregularities of Distribution. Cambridge University Press, Cambridge (1987)
6. Bilyk, D., Lacey, M.T.: On the small ball inequality in three dimensions. Duke Math. J. **143**, 81–115 (2008)
7. Bilyk, D., Lacey, M.T., Vagharshakyan, A.: On the small ball inequality in all dimensions. J. Funct. Anal. **254**, 2470–2502 (2008)
8. Bundschuh, P., Zhu, Y.C.: A method for exact calculation of the discrepancy of low-dimensional point sets I. Abh. Math. Sem. Univ. Hamburg **63**, 115–133 (1993)
9. Chari, S., Rohatgi, P., Srinivasan, A.: Improved algorithms via approximation of probability distributions. J. Comput. System Sci. **61**, 81–107 (2000)
10. de Clerck, L.: A method for exact calculation of the star-discrepancy of plane sets applied to the sequence of Hammersley. Monatsh. Math. **101**, 261–278 (1986)
11. van der Corput, J.G.: Verteilungsfunktionen I. Nederl. Akad. Wetensch. Proc. Ser. B **38**, 813–821 (1935)
12. van der Corput, J.G.: Verteilungsfunktionen II. Nederl. Akad. Wetensch. Proc. Ser. B **38**, 1058–1066 (1935)
13. Davenport, H.: Note on irregularities of distribution. Matematika **3**, 131–135 (1956)
14. Dick, J.: A note on the existence of sequences with small star discrepancy. J. Complexity **23**, 649–652 (2007)
15. Dick, J.: Koksma-Hlawka type inequalities of fractional order. Ann. Mat. Pura Appl. **187**, 385–403 (2008)
16. Dick, J., Pillichshammer, F.: Digital Nets and Sequences. Cambridge University Press, Cambridge (2010)
17. Dobkin, D.P., Eppstein, D., Mitchell, D.P.: Computing the discrepancy with applications to supersampling patterns. ACM Trans. Graph. **15**, 354–376 (1996)
18. Doerr, B.: Generating randomized roundings with cardinality constraints and derandomizations. In: B. Durand, W. Thomas (eds.) STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, *Lecture Notes in Computer Science*, vol. 3884, pp. 571–583. Springer (2006)
19. 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)
20. Doerr, B., Gnewuch, M., Srivastav, A.: Bounds and construction for the star discrepancy via δ-covers (2004). Berichtsreihe des Mathematischen Seminars der Christian-Albrechts-Universität zu Kiel, Report 04–13 (Preliminary preprint version of [21])
21. Doerr, B., Gnewuch, M., Srivastav, A.: Bounds and constructions for the star discrepancy via δ-covers. J. Complexity **21**, 691–709 (2005)
22. Doerr, B., Gnewuch, M., Wahlström, M.: Implementation of a component-by-component algorithm to generate low-discrepancy samples. In: P. L'Ecuyer, A.B. Owen (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2008, pp. 323–338. Springer, Berlin Heidelberg (2009)
23. Doerr, B., Gnewuch, M., Wahlström, M.: Algorithmic construction of low-discrepancy point sets via dependent randomized rounding. J. Complexity **26**, 490–507 (2010)

24. Doerr, B., Wahlström, M.: Randomized rounding in the presence of a cardinality constraint. In: I. Finocchi, J. Hershberger (eds.) Proceedings of the 10th Workshop on Algorithm Engineering and Experiments (ALENEX 2009), pp. 162–174. SIAM, New York, USA (2009)
25. Drmota, M., Tichy, R.F.: Sequences, Discrepancies, and Applications, *Lecture Notes in Math.*, vol. 1651. Springer, Berlin Heidelberg New York (1997)
26. Dueck, G., Scheuer, T.: Threshold accepting: A general purpose algorithm appearing superior to simulated annealing. J. Comp. Phys. **90** (1990)
27. Even, G., Goldreich, O., Luby, M., Nisan, N., Veličković, B.: Approximations of general independent distributions. In: Proceedings of the 24th ACM Symposium on Theory of Computing (STOC), pp. 10–16 (1992)
28. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. **41**, 337–351 (1982)
29. Frank, K., Heinrich, S.: Computing discrepancies of Smolyak quadrature rules. J. Complexity **12**, 287–314 (1996)
30. Frolov, K.K.: An upper estimate for the discrepancy in the L_p-metric, $2 \leq p \leq \infty$. Soviet. Math. Dokl. **21**, 840–842 (1980)
31. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., San Francisco (1979)
32. Giannopoulos, P., Knauer, C., Wahlström, M., Werner, D.: Hardness of discrepancy computation and ε-net verification in high dimension. J. Complexity **28**, 162–176 (2012)
33. Gnewuch, M.: Weighted geometric discrepancies and numerical integration on reproducing kernel Hilbert spaces. J. Complexity **28**, 2–17 (2012)
34. Gnewuch, M.: Bounds for the average L^p-extreme and the L^∞-extreme discrepancy. Electron. J. Combin. **12**, Research Paper 54 (2005)
35. Gnewuch, M.: Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy. J. Complexity **24**, 154–172 (2008)
36. Gnewuch, M.: Construction of minimal bracketing covers for rectangles. Electron. J. Combin. **15**, Research Paper 95 (2008)
37. Gnewuch, M.: On probabilistic results for the discrepancy of a hybrid-Monte Carlo sequence. J. Complexity **23**, 312–317 (2009)
38. 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)
39. Gnewuch, M., Wahlström, M., Winzen, C.: A new randomized algorithm to approximate the star discrepancy based on threshold-accepting. SIAM J. Numer. Anal. **50**, 781–807 (2012)
40. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multidimensional integrals. Numer. Math. **2**, 84–90 (1960)
41. Hardy, G.H., Littlewood, J.E.: Some problems of diophantine approximation: the lattice points of a right-angled triangle. Part I. Proc. London Math. Soc. **20**, 212–249 (1922)
42. Haussler, D.: Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. J. Combin. Theory A **69**, 279–302 (1995)
43. Heinrich, S.: Efficient algorithms for computing the L_2 discrepancy. Math. Comp. **65**, 1621–1633 (1996)
44. Heinrich, S.: Some open problems concerning the star-discrepancy. J. Complexity **19**, 416–419 (2003)
45. 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)
46. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Math. Comp. **67**, 299–322 (1998)
47. 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)
48. Hickernell, F.J., Wang, X.: The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimensions. Math. Comp. **71**, 1641–1661 (2001)
49. Hinrichs, A.: Covering numbers, Vapnik-Červonenkis classes and bounds for the star-discrepancy. J. Complexity **20**, 477–483 (2004)

50. Hinrichs, A., Pillichshammer, F., Schmid, W.C.: Tractability properties of the weighted star discrepancy. J. Complexity **23**, 134–143 (2008)
51. Hlawka, E.: Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. Ann. Mat. Pura Appl. **54**, 325–333 (1961)
52. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Amer. Statist. Assoc. **58**, 13–30 (1963)
53. Hromković, J.: Algorithms for Hard Problems. Springer, Berlin Heidelberg (2003). 2nd edition
54. Johnson, D.S.: The NP-completeness column: an ongoing guide. J. Algorithms **8**, 438–448 (1987)
55. Kirckpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **20**, 671–680 (1983)
56. Koksma, J.F.: Een algemeene stelling uit de theorie der gelijkmatige verdeeling modulo 1. Mathematica B (Zutphen) **11**, 7–11 (1942/3)
57. Lerch, M.: Question 1547. L'Intermédiaire des Mathématiciens **11**, 145–146 (1904)
58. Matoušek, J.: Geometric Discrepancy. Springer-Verlag, Berlin Heidelberg (1999). Revised second printing 2010.
59. Mhaskar, H.N.: On the tractability of multivariate integration and approximation by neural networks. J. Complexity **20**, 561–590 (2004)
60. Niederreiter, H.: Discrepancy and convex programming. Ann. Mat. Pura Appl. **93**, 89–97 (1972)
61. Niederreiter, H.: Methods for estimating discrepancy. In: S.K. Zaremba (ed.) Applications of number theory to numerical analysis, pp. 203–236. Academic Press, New York (1972)
62. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
63. Niederreiter, H.: On the discrepancy of some hybrid sequences. Acta Arith. **138**, 373–398 (2009)
64. Niederreiter, H.: Further discrepancy bounds and an Erdös-Turán-Koksma inequality for hybrid sequences. Monatsh. Math. **161**, 193–222 (2010)
65. Niederreiter, H., Xing, C.: Low-discrepancy sequences and global function fields with many rational places. Finite Fields and their Applications **2**, 241–273 (1996)
66. Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems, Volume I: Linear Information. European Mathematical Society, Freiburg (2008)
67. Novak, E., Woźniakowski, H.: L_2 discrepancy and multivariate integration. In: W.W.L. Chen, W.T. Gowers, H. Halberstam, W.M. Schmidt, R.C. Vaughan (eds.) Essays in honour of Klaus Roth, pp. 359–388. Cambridge Univ. Press, Cambridge (2009)
68. Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems, Volume II: Standard Information for Functionals. European Mathematical Society, Freiburg (2010)
69. Ökten, G.: A probabilistic result on the discrepancy of a hybrid Monte Carlo sequence and applications. Monte Carlo Methods Appl. **2**, 255–270 (1996)
70. Ökten, G., Shah, M., Goncharov, Y.: Random and deterministic digit permutations of the Halton sequence. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 609–622. Springer-Verlag, Berlin (2012)
71. 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)
72. Ostrowski, A.: Bemerkungen zur Theorie der Diophantischen Approximationen I. Abh. Math. Semin. Univ. Hamburg **1**, 77–98 (1922)
73. Ostrowski, A.: Bemerkungen zur Theorie der Diophantischen Approximationen II. Abh. Math. Semin. Univ. Hamburg **1**, 250–251 (1922)
74. Panconesi, A., Srinivasan, A.: Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding bounds. SIAM J. Comput. **26**, 350–368 (1997)
75. Raghavan, P.: Probabilistic construction of deterministic algorithms: approximating packing integer programs. J. Comput. Syst. Sci. **37**, 130–143 (1988)
76. Roth, K.F.: On irregularities of distribution. Mathematika **1**, 73–79 (1954)

77. Roth, K.F.: On irregularities of distribution III. Acta Arith. **35**, 373–384 (1979)
78. Roth, K.F.: On irregularities of distribution IV. Acta Arith. **37**, 67–75 (1980)
79. Schmidt, W.M.: On irregularities of distribution VII. Acta Arith. **21**, 45–50 (1972)
80. Shah, M.: A genetic algorithm approach to estimate lower bounds of the star discrepancy. Monte Carlo Methods Appl. **16**, 379–398 (2010)
81. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Oxford University Press, New York (1994)
82. Sloan, I.H., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? J. Complexity **14**, 1–33 (1998)
83. 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, *Lecture Notes in Statistic*, vol. 106, pp. 121–148. Springer, Berlin (1995)
84. Srinivasan, A.: Distributions on level-sets with applications to approximation algorithms. In: Proceedings of FOCS'01, pp. 588–597 (2001)
85. Steinerberger, S.: The asymptotic behavior of the average L^p-discrepancies and a randomized discrepancy. Electron. J. Combin. **17(1)**, Research paper 106 (2010)
86. Talagrand, M.: Sharper bounds for Gaussian and empirical processes. Ann. Probab. **22**, 28–76 (1994)
87. Thiémard, E.: Computing bounds for the star discrepancy. Computing **65**, 169–186 (2000)
88. Thiémard, E.: An algorithm to compute bounds for the star discrepancy. J. Complexity **17**, 850–880 (2001)
89. Thiémard, E.: Optimal volume subintervals with k points and star discrepancy via integer programming. Math. Meth. Oper. Res. **54**, 21–45 (2001)
90. van der Vaart, A.W., Wellner, J.A.: Weak Convergence and Empirical Processes. Springer Series in Statistics. Springer, New York (1996)
91. Warnock, T.T.: Computational investigations of low-discrepancy point sets. In: S.K. Zaremba (ed.) Applications of number theory to numerical analysis, pp. 319–343. Academic Press, New York (1972)
92. Winker, P., Fang, K.T.: Application of threshold-accepting to the evaluation of the discrepancy of a set of points. SIAM J. Numer. Anal. **34**, 2028–2042 (1997)
93. Woźniakowski, H.: Average case complexity of multivariate integration. Bull. Amer. Math. Soc. (N. S.) **24**, 185–191 (1991)
94. Zaremba, S.K.: Some applications of multidimensional integration by parts. Ann. Polon. Math. **21**, 85–96 (1968)

Asymptotic Equivalence Between Boundary Perturbations and Discrete Exit Times: Application to Simulation Schemes

E. Gobet

Abstract We present two problems that are apparently disconnected, and we show how they are actually related to each other. First, we investigate the sensitivity of the expectation of functionals of diffusion process stopped at the exit from a domain, as the boundary is perturbed. Second, we analyze the discrete monitoring bias when simulating stopped diffusions, emphasizing the role of overshoot asymptotics. Then, we derive a simple and accurate scheme for simulating stopped diffusions.

1 Introduction

The problem. This work is motivated by a nice boundary shifting result, proved by Broadie et al. [8], in the context of killed scalar Brownian motion. To introduce the topic, let us fix few notations. We consider a standard one-dimensional Brownian motion $(W_t)_{t\geq 0}$ defined in a standard manner on a filtered probability space $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, \mathbb{P})$, and we set $X_t = x + \mu t + \sigma W_t$, for parameters $\mu \in \mathbb{R}$ and $\sigma > 0$. Let us fix a terminal time $T > 0$ and an upper barrier $U > x$: for a time-step $\Delta = T/m$ ($m \in \mathbb{N}$), define the discretization times $(t_i := i\Delta)_{i\geq 0}$. Consider the function $\Phi(y) = (K - \exp(y))_+$ (assuming $K < \exp(U)$), which choice is related to the pricing of barrier options in financial engineering: then we have [8]

$$\mathbb{E}\big(\mathbf{1}_{\forall t_i \leq T: X_{t_i} < U}\ \Phi(X_T)\big) = \mathbb{E}\big(\mathbf{1}_{\forall t \leq T: X_t < U + c_0\sigma\sqrt{\Delta}}\ \Phi(X_T)\big) + o(\Delta^{1/2}) \tag{1}$$

where

$$c_0 = \frac{-\zeta(1/2)}{\sqrt{2\pi}} = 0.5826\cdots, \tag{2}$$

E. Gobet
Ecole Polytechnique, CMAP, Route de Saclay, 91128, Palaiseau Cedex, France
e-mail: emmanuel.gobet@polytechnique.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_4, © Springer-Verlag Berlin Heidelberg 2012

$\zeta(.)$ being the Riemann Zeta function. In other words, killing the arithmetic Brownian motion at discrete times is asymptotically equivalent (as the monitoring frequency $1/\Delta$ goes to infinity) to the continuous-time killing, provided that one appropriately shifts the boundary from a quantity proportional to $\sqrt{\Delta}$.

Actually, even for more general Φ, it is easy to show that both expectations in (1) converge to the same limit $\mathbb{E}\big(\mathbf{1}_{\forall t \le T: X_t < U}\ \Phi(X_T)\big)$, as $\Delta \to 0$. The striking feature in the approximation (1) is the magnitude of the remainder term: it is smaller that $\Delta^{1/2}$. Indeed, for years it has been numerically observed (see [2,7] among others) that the bias due to discrete exit time presumably yields an error of order $\frac{1}{2}$ w.r.t. Δ:

$$\begin{aligned}\mathbb{E}\big(\mathbf{1}_{\forall t_i \le T: X_{t_i} < U}\ \Phi(X_T)\big) =& \mathbb{E}\big(\mathbf{1}_{\forall t \le T: X_t < U}\ \Phi(X_T)\big)\\ &+ c_1(T, X, U, \Phi)\Delta^{1/2} + o(\Delta^{1/2}).\end{aligned} \tag{3}$$

Since $\inf\{t_i \ge 0 : X_{t_i} \ge U\} \ge \inf\{t \ge 0 : X_t \ge U\}$ a.s., the (numerical) constant $c_1(T, X, U, \Phi)$ is positive for non-negative and non-zero function Φ, which corresponds to a systematic overestimation of the expectation related to continuous killing using a discrete killing. On the other hand, it is reasonable (see Sect. 2) to guess that $U \mapsto \mathbb{E}\big(\mathbf{1}_{\forall t \le T: X_t < U}\ \Phi(X_T)\big)$ is smooth, so that

$$\begin{aligned}\mathbb{E}\big(\mathbf{1}_{\forall t \le T: X_t < U+\varepsilon}\ \Phi(X_T)\big) =& \mathbb{E}\big(\mathbf{1}_{\forall t \le T: X_t < U}\ \Phi(X_T)\big)\\ &+ c_2(T, X, U, \Phi)\varepsilon + o(\varepsilon).\end{aligned} \tag{4}$$

Then, by equating both expansions (3) and (4) when $\varepsilon = c_0\sigma\sqrt{\Delta}$, the result (1) can be simply reformulated as the equality

$$c_1(T, X, U, \Phi) = c_2(T, X, U, \Phi)c_0\sigma. \tag{5}$$

In other words, it suggests that discrete killing and boundary perturbation are asymptotically equivalent. Our aim is to show that such an equivalence can be generalized far beyond the framework of the scalar arithmetic Brownian motion, i.e. it holds true for multi-dimensional time-inhomogeneous diffusion processes in time-dependent domains. Of course, for such a generalization with time and space dependent σ, the previous equality (5) should be modified. However, we will show (see (13) and Theorem 4) that an adaptation of the identity (1) holds true, still involving the *same universal constant* $c_0 \approx 0.5826$, *whatever the model, the domain or the dimension are.* In this sense, from this intriguing result, we claim that there is a general asymptotic equivalence between discrete killing and boundary perturbation.

Mathematical tools. To achieve such a level of generality, new results have to be developed. Indeed, in the original proof of (1), Broadie et al. leverage two fine properties, mostly known in the Brownian case.

1. On the one hand, they use previous asymptotic results about Brownian overshoots by Siegmund [29, 30]. Siegmund shows that the exit time probability of random walks can be expanded at first order in the diffusion asymptotics. In the limit, the aforementioned constant c_0 comes into play: it is equal to the expectation of the asymptotic overshoot of the centered Gaussian random walk (it is also connected to moments of the first ladder height of a Gaussian random walk). We detail the related statement in Sect. 3. There, we also expose the generalization to multi-dimensional diffusion processes, recently proved by Gobet et al. in [21].
2. On the other hand, using the explicit joint law of the supremum of drifted Brownian motion and of its terminal value, it is possible to compute the quantity $\mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t<U}\ \Phi(X_T)\big)$ for $\Phi(y) = (K - \exp(y))_+$, and thus to differentiate the quantity w.r.t. U (in order to obtain the expansion (4)). But it is hopeless to make the probability laws explicit in the case of general diffusion processes. However, quite surprisingly, without knowing explicitly the underlying law, it is possible to differentiate $\mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t<U}\ \Phi(X_T)\big)$ w.r.t. U for scalar diffusion and to derive a probabilistic representation of the derivative for quite arbitrary function Φ (see [13]); these computations can be generalized to several dimensions as well. We will present these results in Sect. 2.

Applications. In the original paper [8], the authors take advantage of the equality (1) to propose a simple and efficient way to compute the price of discrete barrier options, which is up to a discounted factor equal to $P^{\Delta} = \mathbb{E}(\mathbf{1}_{\forall t_i\leq T: X_{t_i}<U}\ \Phi(X_T))$. Indeed, for some choices of *payoff* functions Φ, owing to the explicit law of the killed drifted Brownian motion, the expectation $\mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t<U}\ \Phi(X_T)\big)$ has closed forms, from which (using (1)) we may deduce an approximation of P^{Δ} for small Δ. In [9], these ideas have been extended to smooth functionals of the maximum of X.

Another application, which becomes meaningful for general diffusion processes, is the computation of $\mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t<U}\ \Phi(X_T)\big)$ by Monte Carlo simulations. In that case, the equality (1) should be transformed into

$$\mathbb{E}\big(\mathbf{1}_{\forall t_i\leq T: X_{t_i}<U-c_0\sigma\sqrt{\Delta}}\ \Phi(X_T)\big) = \mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t<U}\ \Phi(X_T)\big) + o(\Delta^{1/2}). \qquad (6)$$

Since discrete killing yields an overestimation regarding the continuous-time case, we compensate this bias by making the barrier closer, so that the main error term is removed. In the same way that we extend (1) to general diffusions, we do so for the approximation (6).

The interest in this generalization is the simplicity of the resulting numerical scheme: we only need to monitor the process X at discrete times, in a domain which boundary has been shifted inwards. We actually prove that the diffusion process can be approximated using an Euler scheme with time step Δ, maintaining the same global accuracy $o(\Delta^{\frac{1}{2}})$, see Theorem 5.

Regarding the applications in financial engineering, the approximations potentially apply to general models, with stochastic volatility and stochastic short rate, or even including Poisson jumps; see numerical evidences from the tests in [18]. Some jump diffusion models are analyzed in [14] and [15]. There are also potential applications in credit risk, regarding the structural credit models where the default is associated to the exit time of domains by some processes (see [6, Chap. 3] for instance). Here, we are not discussing applications to American options and optimal stopping problems: see the discussion in [13], or in Labart's PhD thesis [23] with partial results. We also refer the reader to the article [24] by Lai et al. about corrected random walk approximations in free boundary problems.

Few other related references. During the last 15 years, a lot of attention has been paid to the evaluation of $\mathbb{E}\big(\mathbf{1}_{\forall t\leq T: X_t\in D}\ \Phi(X_T)\big)$ by means of Monte Carlo simulations, for general diffusion process X and for general domain $D \subset \mathbb{R}^d$. Indeed, the functional of interest is very irregular w.r.t. the process path, which induces large errors when the process is sampled[1] at discrete times. By monitoring the process X at discrete times, one may not detect an exit, although the *continuous* path has reached the boundary. Thus, the corner stone is the quantitative study of the conditional exit probability given two simulated values of the process:

$$\mathbb{P}(\exists t \in [t_i, t_{i+1}] : X_t \notin D | X_{t_i}, X_{t_{i+1}}). \tag{7}$$

In [3], Baldi derives an expansion of the exit probability of a pinned Brownian motion, as time is small. He applies this expansion to the efficient simulation of a multidimensional Brownian motion killed at its exit from a domain. In [4], Baldi and Caramellino extend the expansion result to scalar diffusions. In [16], for elliptic diffusions in $\mathbb{R}^d$, Gobet proves that when X is replaced by its continuous Euler scheme (meaning that, at each time step, we take into account the exit probability of a pinned scaled Brownian motion), the resulting numerical scheme achieves the optimal convergence rate Δ (optimal in the sense that it coincides with the convergence rate without domain [5]). To get an easily implementable scheme for arbitrary domains, it is enough to replace locally the domain by half-space approximations in the computations of (7), maintaining the same convergence rate: it is proved in [17]. A different approach is followed by Jansons and Lythe [22]: the discretization times are given by the random jump times of a Poisson process with large intensity. The exponential distribution of interarrival time allows for quite explicit approximations of the exit probability. Another approach is developed by Shevchenko in [28] when the domain is the intersection of half-spaces and the process is an arithmetic Brownian motion: relying on the Fréchet copula inequalities, the author bounds from above and from below the related exit probability by two explicit quantities. The procedure gives accurate numerical results. Yet another approach is developed by Roberts and co-authors (see [10] and references therein), using a smart rejection sampling for exact simulation: it applies to one-dimensional killed/stopped diffusions but it is hard to generalize to general multi-dimensional models.

[1] Exactly or approximatively using an Euler scheme for instance.

The analysis of the discrete time error (neglecting the possible exit between two consecutive discretization times) is much more delicate. Upper bounds of magnitude $\Delta^{1/2}$ are established in [16], and similar lower bounds for hypoelliptic diffusions in [19]. Same upper bounds in the case of Itô processes are provided in [20], assessing that the discrete time error is not related to the underlying Markov structure. The expansion at order $\frac{1}{2}$ w.r.t. Δ has been established recently by Gobet and Menozzi [21], this is presented in Sect. 3.

Organization of the paper. In Sect. 2, we analyze the boundary sensitivity of expectations of killed/stopped diffusion processes: our purpose is to derive the expansion (4), in the more general framework of diffusion process in time-dependent domain. In Sect. 3, we derive expansion results related to discretization error: these results are based on recent asymptotic results of diffusion overshoots. In Sect. 4, we combine the two previous asymptotic results to design a simple and efficient numerical scheme for the weak approximation of stopped diffusion processes. Then, we illustrate its numerical performance.

Notation and assumptions used throughout the paper. Regarding the assumptions, we state sufficient conditions that make valid all the following results. However for some of them, these assumptions are too strong and we refer the reader to [21] for detailed statements.

Stochastic differential equation. Let us consider a d-dimensional diffusion process whose dynamics is given by

$$X_t = x + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)dW_s \tag{8}$$

where W is a standard d'-dimensional Brownian motion defined on a filtered probability space $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, \mathbb{P})$ satisfying the usual conditions. In the following, we assume that

- *(Smoothness)* the functions b and σ belong to the $C_b^{1,2}$-space[2];
- *(Uniform ellipticity)* for some $a_0 > 0$, it holds

$$\xi^*[\sigma\sigma^*](t,x)\xi \geq a_0|\xi|^2$$

for any $(t, x, \xi) \in [0, T] \times \mathbb{R}^d \times \mathbb{R}^d$.

The regularity assumption ensures the existence and uniqueness of a strong solution to (8). As usual, we freely write $\mathbb{E}_x[.] := \mathbb{E}[.|X_0 = x]$.

[2] The functions are bounded, with bounded partial derivatives $\partial_t, \partial_x, \partial_x^2$.

Time-dependent domain and exit time. We consider $(D_t)_{t\geq 0}$, a time-dependent family of smooth bounded domains of $\mathbb{R}^d$, that is also smooth with respect to t. For a fixed deterministic time $T > 0$, this defines a time-space domain

$$D := \{(t,x) : 0 < t < T, x \in D_t\} \subset]0,T[\times\mathbb{R}^d.$$

We require this domain D to be of class C^3 (see [21] for a precise definition). We write ∂D_t for the boundary of D_t. Cylindrical domains are specific cases of time-dependent domains of the form $D =]0,T[\times D_0$, where D_0 is a usual domain of $\mathbb{R}^d$ ($D_t = D_0$ for any t). Time-dependent domains in dimension $d = 1$ are typically of the form $D = \{(t,x) : 0 < t < T, \varphi_1(t) < x < \varphi_2(t)\}$ for two functions φ_1 and φ_2 (the time-varying boundaries).
Now, set

$$\tau := \inf\{t > 0 : X_t \notin D_t\},$$

then $\tau \wedge T$ is the first exit time of $(s, X_s)_s$ from the time-space domain D.

Distance function. To appropriately describe the overshoot beyond the boundary, we will use the signed distance function F defined as follows. Under the assumption on D, there is a constant $r_0 > 0$ such that[3]

$$F(t,x) := \begin{cases} -d(x,\partial D_t), & \text{for } x \in D_t^c,\ d(x,\partial D_t) \leq r_0,\ 0 \leq t \leq T, \\ d(x,\partial D_t), & \text{for } x \in D_t,\ d(x,\partial D_t) \leq r_0,\ 0 \leq t \leq T, \end{cases} \tag{9}$$

and F can be extended to the whole space while preserving the sign, i.e. $F(t,x) < 0$ (> 0) iif $x \notin \bar{D}_t$ ($x \in D_t$); see [25], Sect. X.3. In our case, the extension can be achieved as a H_3-function; moreover, in the r_0-neighborhood of the spatial boundary ∂D_t, the projection $\pi_{\partial D_t}(x) = \operatorname{argmin}_{y\in\partial D_t}|y-x|$ is uniquely defined, and $n(t,x) = [\nabla F]^*(t,x)$ is the unit inward normal vector to ∂D_t at $\pi_{\partial D_t}(x)$.

Data. To define the path functional of interest, we are given continuous functions $f, g, k : \bar{D} \to \mathbb{R}$: they are assumed to be in the $H_{1+\theta}$-space[4] for some $\theta \in]0,1]$. In the following, we study and approximate the quantity

$$\mathbb{E}_x[g(\tau\wedge T, X_{\tau\wedge T})Z_{\tau\wedge T} + \int_0^{\tau\wedge T} Z_s f(s,X_s)ds], \tag{10}$$

$$Z_s = \exp(-\int_0^s k(r,X_r)dr).$$

[3] As usual, $d(x,C) = \inf_{y\in C}|x-y|$.

[4] Meaning, as usual, that the functions are $(1+\theta)/2$-Hölder continuous in time, they are continuously differentiable in space, the derivatives are θ-Hölder continuous in space and $\theta/2$-Hölder continuous in time; for a precise statement, see [21].

This includes the example presented at the beginning by taking $f, k \equiv 0$ and $g(t,x) = \mathbf{1}_{t=T}\Phi(x)$. Observe that the smoothness requirement on g implies in particular that Φ vanishes at $x \in \partial D_T$: this condition is connected to the condition $K < \exp(U)$ imposed in the first example of the introduction. Roughly speaking, it ensures that the related PDE (defined below) is smooth up the corner $\{t = T\} \times \partial D_T$. It is still an open issue to know whether the current analysis on asymptotic equivalence holds true without this requirement.

Parabolic Partial Differential Equations. The latter expectation in (10) gives a probabilistic representation of solution to second-order parabolic linear PDE with Cauchy-Dirichlet boundary conditions

$$\begin{cases} \partial_t u + Lu - ku + f = 0 & \text{in } D, \\ u = g & \text{on } \{t = T, x \in \bar{D}_T\} \bigcup \{0 < t < T, x \in \partial D_t\}, \end{cases} \tag{11}$$

where L is the infinitesimal generator of X. Under our assumptions, there is a unique classical $C^{1,2}(D)$ solution to (11), which is in the class $H_{1+\theta}(\bar{D})$: in particular, the gradient $\nabla_x u$ exists and is Hölder-continuous up to the boundary. We have

$$u(0,x) = \mathbb{E}_x\big[g(\tau \wedge T, X_{\tau\wedge T}) Z_{\tau\wedge T} + \int_0^{\tau\wedge T} Z_s f(s, X_s) ds\big].$$

Other values $u(t,x)$ $(0 < t < T)$ are obtained by starting the diffusion at time t.

Time discretization and Euler scheme. The time step is denoted by $\Delta = T/m > 0$ $(m \in \mathbb{N}^*)$ and the discretization times are $(t_i = i\Delta)_{i\geq 0}$. For $t \geq 0$, define $\varphi(t) = t_i$ for $t_i \leq t < t_{i+1}$ and introduce

$$X_0^\Delta = x, \quad X_{t_{i+1}}^\Delta = X_{t_i}^\Delta + b(t_i, X_{t_i}^\Delta)\Delta + \sigma(t_i, X_{t_i}^\Delta)(W_{t_{i+1}} - W_{t_i}). \tag{12}$$

2 Boundary Sensitivity

In this section, we analyse the sensitivity of

$$\mathbb{E}_x[g(\tau \wedge T, X_{\tau\wedge T}) Z_{\tau\wedge T} + \int_0^{\tau\wedge T} Z_s f(s, X_s) ds]$$

w.r.t. boundary perturbations.

Background results. In the one-dimensional time-homogenous case with $f, k \equiv 0$ and $g(t,x) = \mathbf{1}_{t=T}\Phi(x)$, it corresponds to the investigation of regularity of the expectation

$$\mathbb{E}_x(\mathbf{1}_{\forall t\leq T:X_t<U}\ \Phi(X_T)) = \int_{-\infty}^{U} q(0,x;T,y;U)\Phi(y)dy$$

w.r.t. U, assuming for the moment that the density q exists. Let us discuss the different possible strategies to tackle this regularity issue.

- In the Brownian case, the reflection principle gives access to the explicit form of the density q of the killed process.
- To allow non-constant coefficients, one could go back to the Brownian motion using a Lamperti and a Girsanov transform: this approach has been developed by Pauwels [27]. He obtains a representation of the density q using expectations of Bessel bridges. However, the differentiation of this representation w.r.t. U seems to be delicate.
- A direct application of Malliavin calculus [26] is not possible, since the maximum of a random process is usually not very smooth in Malliavin sense. In [11], Cattiaux applies specific stochastic perturbations to prove the regularity of $q(t,x;T,y;U)$ w.r.t. all variables except U.

On the other hand, from the PDE point of view, computing the boundary sensitivity is a very standard issue which dates back to Hadamard, at the beginning of the twentieth century. Nowadays, it is motivated by issues in shape optimization of elastic structures, see [1] for instance. However, in these applications, we mainly consider stationary problems (elliptic PDEs) or heat equations with constant coefficients. It has justified in [13] the development of results in a wider framework.

General sensitivity results. We apply spatial perturbations to the domain, as follows. For $\varepsilon \in \mathbb{R}$ and $\Theta \in C^{1,2}([0,T]\times\mathbb{R}^d,\mathbb{R}^d)$, we define the perturbed domain in the direction Θ by

$$D_t^{\epsilon} = \{x : x + \varepsilon\Theta(t,x) \in D_t\}.$$

We denote the new exit time by

$$\tau_\varepsilon := \inf\{s > 0 : X_s \notin D_s^{\epsilon}\}.$$

The main result is

Theorem 1. *[13, Theorem 2.2] For $x \in D_0$, the mapping*

$$J : \varepsilon \mapsto \mathbb{E}_x\big[g(\tau_\varepsilon\wedge T, X_{\tau_\varepsilon\wedge T})Z_{\tau_\varepsilon\wedge T} + \int_0^{\tau_\varepsilon\wedge T} Z_s f(s,X_s)ds\big]$$

is differentiable at $\varepsilon = 0$ and

$$\partial_\varepsilon J(\varepsilon)\big|_{\varepsilon=0} = \mathbb{E}_x\left[\mathbf{1}_{\tau\leq T} Z_\tau(\partial_n u - \partial_n g)[n\cdot\Theta](\tau, X_\tau)\right],$$

where we write ∂_n for the normal derivative.

Note that the derivative above depends only on the normal component of Θ. In terms of approximation, the above differentiation result writes

$$\begin{aligned}\mathbb{E}_x\big[g(\tau_\varepsilon \wedge T, X_{\tau_\varepsilon\wedge T})Z_{\tau_\varepsilon\wedge T} + \int_0^{\tau_\varepsilon\wedge T} Z_s f(s, X_s)ds\big] \\ = \mathbb{E}_x\big[g(\tau \wedge T, X_{\tau\wedge T})Z_{\tau\wedge T} + \int_0^{\tau\wedge T} Z_s f(s, X_s)ds\big] \\ + \varepsilon\mathbb{E}_x\left[\mathbf{1}_{\tau\leq T} Z_\tau(\partial_n u - \partial_n g)[n\cdot\Theta](\tau, X_\tau)\right] + o(\varepsilon).\end{aligned} \tag{13}$$

Sketch of proof. Of course, a pathwise differentiation of τ_ε w.r.t. ε is impossible. The trick is to transfer the domain perturbation to a process perturbation, which is more convenient for using stochastic tools. Namely, if we define

$$X_s^\epsilon := X_s + \epsilon\Theta(s, X_s), \quad \tau^\epsilon := \inf\{s > 0 : X_s^\epsilon \notin D_s\},$$

we easily justify that X^ϵ converges uniformly to X as $\epsilon \to 0$ and that $\tau_\epsilon = \tau^\epsilon$ converges a.s. towards τ (under uniform ellipticity condition).

To complete the proof, assume for simplicity $f, k \equiv 0$. Then, decompose the difference

$$J(\varepsilon) - J(0) = \Delta_{1,\epsilon} + \Delta_{2,\epsilon} + \Delta_{3,\epsilon} + \Delta_{4,\epsilon},$$

where

$$\begin{aligned}\Delta_{1,\varepsilon} &= \mathbb{E}_x\big[g(\tau^\varepsilon \wedge T, [\mathrm{Id} + \varepsilon\Theta(\tau^\varepsilon \wedge T, .)]^{-1}X^\varepsilon_{\tau^\varepsilon\wedge T})\big] - \mathbb{E}_x\big[g(\tau^\varepsilon \wedge T, X^\varepsilon_{\tau^\varepsilon\wedge T})\big],\\ \Delta_{2,\varepsilon} &= \mathbb{E}_x\big[g(\tau^\varepsilon \wedge T, X^\varepsilon_{\tau^\varepsilon\wedge T}) - u(\tau^\varepsilon \wedge \tau \wedge T, X^\varepsilon_{\tau^\varepsilon\wedge\tau\wedge T})\big],\\ \Delta_{3,\varepsilon} &= \mathbb{E}_x\big[u(\tau^\varepsilon \wedge \tau \wedge T, X^\varepsilon_{\tau^\varepsilon\wedge\tau\wedge T}) - u(\tau^\varepsilon \wedge \tau \wedge T, X_{\tau^\varepsilon\wedge\tau\wedge T})\big],\\ \Delta_{4,\varepsilon} &= \mathbb{E}_x\big[u(\tau^\varepsilon \wedge \tau \wedge T, X_{\tau^\varepsilon\wedge\tau\wedge T})\big] - u(0, x).\end{aligned}$$

The first term divided by ε readily converges to $-\mathbb{E}_x[\mathbf{1}_{\tau\leq T}\nabla g\ \Theta(\tau, X_\tau)]$. Simply using the expression of X^ε, we prove that $\Delta_{3,\varepsilon}/\varepsilon$ converge to $\mathbb{E}_x[\mathbf{1}_{\tau\leq T}\nabla u\ \Theta(\tau, X_\tau)]$. The fourth term is equal to 0, due to the martingale property $s \mapsto u(s, X_s)$ up to the exit time τ. The second term divided by ε does not give any contribution at the limit, which is more involved to justify. Since $u = g$ coincide on the boundary ∂D_t, only the normal component of $\nabla(u - g)$ remain. □

Going back to the introduction, the application of this result to (4) gives

$$c_2(T, X, U, \Phi) = -\mathbb{E}_x(\mathbf{1}_{\tau\leq T}\ u'_x(\tau, U)). \tag{14}$$

The boundary sensitivity analysis can be extended also to reflected diffusion processes (connected to PDE with Neuman conditions). For the applications of these boundary sensitivities, we refer the reader to [13].

3 Discretization Error

To the Euler scheme defined in (12), we naturally associate its discrete exit time $\tau^{\Delta} := \inf\{t_i > 0 : X^{\Delta}_{t_i} \notin D_{t_i}\}$. We approximate the functional

$$V_\tau := g(\tau \wedge T, X_{\tau\wedge T}) Z_{\tau\wedge T} + \int_0^{\tau\wedge T} Z_s f(s, X_s)\, ds$$

by

$$V^{\Delta}_{\tau^{\Delta}} := g(\tau^{\Delta} \wedge T, X^{\Delta}_{\tau^{\Delta}\wedge T}) Z^{\Delta}_{\tau^{\Delta}\wedge T} + \int_0^{\tau^{\Delta}\wedge T} Z^{\Delta}_{\varphi(s)} f(\varphi(s), X^{\Delta}_{\varphi(s)}) ds$$

with

$$Z^{\Delta}_t = e^{-\int_0^t k(\varphi(r), X^{\Delta}_{\varphi(r)}) dr}.$$

Note that in $V^{\Delta}_{\tau^{\Delta}}$, on $\{\tau^{\Delta} \leq T\}$ g is a.s. not evaluated on the side part $\bigcup_{0\leq t\leq T}\{t\} \times \partial D_t$ of the boundary. Although this approximation seems to be coarse, it does not affect the convergence rate and really reduces the computational cost with respect to the alternative that would consist in taking the projection on ∂D.

If the process X can be exactly simulated at times $(t_i)_i$, of course we should do so. However, we can show [21] that the main error term is due to the use of the discrete exit time: indeed, replacing X by X^{Δ} yields weak errors of magnitude Δ instead of $\sqrt{\Delta}$.

Now we state a first result, regarding the decomposition of the weak error

$$\mathbb{E}_x[V^{\Delta}_{\tau^{\Delta}} - V_\tau], \tag{15}$$

using the overshoot when the process exits from D at time $\tau^{\Delta} \leq T$:

$$F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}}). \tag{16}$$

Since $F(t, x) > 0$ for $x \in D_t$, the discrete-time process $(F^{-}(t_i, X^{\Delta}_{t_i}))_{i\geq 0}$ is equal to 0 before time τ^{Δ}, and then it is equal to $d(X^{\Delta}_{\tau^{\Delta}}, \partial D_{\tau^{\Delta}})$ (provided that it is smaller than r_0, which occurs with very high probability): this justifies the label of overshoot for (16).

Theorem 2. *[21, Theorem 6] We have*

$$\mathbb{E}_x[V^{\Delta}_{\tau^{\Delta}} - V_\tau] = \mathbb{E}_x(\mathbf{1}_{\tau^{\Delta}\leq T} Z^{\Delta}_{\tau^{\Delta}} (\partial_n u - \partial_n g)(\tau^{\Delta}, \pi_{\partial D_{\tau^{\Delta}}}(X^{\Delta}_{\tau^{\Delta}})) F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}})) + o(\sqrt{\Delta}).$$

Because the increments of X^{Δ} behave like $\sqrt{\Delta}$, we can formally obtain that $F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}})$ is of order $\sqrt{\Delta}$ in L_p. This heuristics is fully justified in [21, Proposition 2]. The first error decomposition of the above type appears in [16]. It has been leveraged in [19] to prove lower and upper bounds for the above weak error. But the full first-order expansion requires the additional knowledge of the asymptotic distribution of the renormalized overshoot $\Delta^{-\frac{1}{2}}F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}})$. This is the purpose of the following result.

Theorem 3. *[21, Theorem 3] Let h be a continuous function with compact support. For all $t \in [0,T]$, $x \in D_0$, $y \geq 0$, we have*

$$\lim_{\Delta \to 0} \mathbb{E}_x[\mathbf{1}_{\tau^{\Delta} \leq t} Z^{\Delta}_{\tau^{\Delta}} h(X^{\Delta}_{\tau^{\Delta}}) \mathbf{1}_{F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}}) \geq y\sqrt{\Delta}}] = \mathbb{E}_x\big[\mathbf{1}_{\tau \leq t} Z_\tau h(X_\tau)\big(1 - H(y/|\nabla F \sigma(\tau, X_\tau)|)\big)\big]$$

with

$$H(y) := (\mathbb{E}_0[s_{\tau^+}])^{-1} \int_0^y \mathbb{P}_0[s_{\tau^+} > z]dz,$$

$s_0 := 0$ and

$$\forall n \geq 1, \quad s_n := \sum_{i=1}^{n} G^i,$$

the G^i being i.i.d. standard centered normal variables, $\tau^+ := \inf\{n \geq 0 : s_n > 0\}$.

In other words, $(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}}, \Delta^{-1/2}F^{-}(\tau^{\Delta}, X^{\Delta}_{\tau^{\Delta}}))$ weakly converges to $(\tau, X_\tau, |\nabla F\sigma(\tau, X_\tau)|Y)$ where Y is a random variable independent of (τ, X_τ), and which cumulative function is equal to H. This is a non-trivial generalization of Siegmund results [29]. Actually, Y has the asymptotic law of the renormalized Brownian overshoot in dimension 1. To complete our discrete time error analysis, in view of Theorems 2 and 3 we need to compute the mean of Y: we set

$$c_0 := \mathbb{E}(Y) = \frac{\mathbb{E}_0[s^2_{\tau^+}]}{2\mathbb{E}_0[s_{\tau^+}]}.$$

The value of c_0 given in (2) is obtained by [29], see also more recently [12].

The proof of Theorem 3 is very technical: it consists in showing that essentially, the phenomenon inherits from the Brownian behavior. This follows from two facts: X^{Δ} locally behaves like a scaled Brownian motion and the boundary is locally flat.

Using an additional uniform integrability property of the renormalized overshoot, we can pass to the limit in Theorem 2 to obtain

Theorem 4. *[21, Theorem 4] For $x \in D_0$, the discrete time error can be expanded at first order w.r.t. $\sqrt{\Delta}$:*

$$\mathbb{E}_x[V^{\Delta}_{\tau^{\Delta}} - V_\tau] = c_0\sqrt{\Delta}\mathbb{E}_x(\mathbf{1}_{\tau \leq T} Z_\tau(\partial_n u - \partial_n g)(\tau, X_\tau)|\nabla F\sigma(\tau, X_\tau)|) + o(\sqrt{\Delta}).$$

Unfortunately, the remainder term is presumably not uniform in x (some particular behavior occurs for x that are close to the boundary ∂D_0 at a distance of order $\sqrt{\Delta}$); this is discussed in [18].

Going back to the introduction, applying this result to (3) gives

$$c_1(T, X, U, \Phi) = -c_0\mathbb{E}_x(\mathbf{1}_{\tau \leq T}\ u'_x(\tau, X_\tau)\sigma) = c_0\sigma c_2(T, X, U, \Phi)$$

by using (14); we have just recovered the identity (5).

Comparing (13) and Theorem 4, observe that both expansions can be perfectly matched by adjusting ε and Θ according to Δ and σ: this is the so-called asymptotic equivalence between boundary perturbation and discrete time error. This is the starting point to design an improved scheme for simulating stopped diffusion processes.

4 Boundary Correction for Simulating Stopped Diffusion Process

Gathering together the main results of two previous sections (and omitting technical details) leads to the following numerical scheme, which is simple and very efficient compared to existing methods. We propose to stop the Euler scheme at its exit of a smaller domain in order to compensate the underestimation of exit time and to achieve an error of order $o(\sqrt{\Delta})$. The smaller domain is defined by its time-section

$$D_t^{\Delta} = \{x \in D_t : d(x, \partial D_t) > c_0\sqrt{\Delta}|n^*\sigma(t,x)|\}$$

where $n(t,x)$ is the inward normal vector at the closest point to x on the boundary ∂D_t.[5] Thus, the associated exit time of the Euler scheme is given by

$$\hat{\tau}^{\Delta} = \inf\{t_i > 0 : X_{t_i}^{\Delta} \notin D_{t_i}^{\Delta}\} \le \tau^{\Delta}.$$

It should be noticed that the simulation of $\hat{\tau}^{\Delta}$ is as easy and as quick as that of τ^{Δ}, which is a major advantage. Then, the functional V_τ is approximated by

$$V_{\hat{\tau}^{\Delta}}^{\Delta} = g(\hat{\tau}^{\Delta} \wedge T, X_{\hat{\tau}^{\Delta}\wedge T}^{\Delta})Z_{\hat{\tau}^{\Delta}\wedge T}^{\Delta} + \int_0^{\hat{\tau}^{\Delta}\wedge T} Z_{\varphi(s)}^{\Delta} f(\varphi(s), X_{\varphi(s)}^{\Delta})ds.$$

Theorem 5. *[21, Theorem 5] For $x \in D_0$, we have*

$$\mathbb{E}_x[V_{\hat{\tau}^{\Delta}}^{\Delta} - V_\tau] = o(\sqrt{\Delta}).$$

Any other boundary correction would have led to an error of magnitude $\sqrt{\Delta}$, instead of $o(\sqrt{\Delta})$.

So far, we have exposed the result for problems defined within a finite time horizon T, but theoretical results extend to stationary problems as well (see [21, Theorem 18]). We complete this presentation by giving a numerical example borrowed to [21], in the infinite horizon situation (elliptic PDE). We take $d = d' = 3$,

[5] The closest point to x may not be unique for points x far from ∂D_t. But since the above definition of D_t^{Δ} involves only points close to the boundary, this does not make any difference.

Table 1 Supremum of the absolute error for the Euler scheme (relative error in % in parenthesis).

Δ	Without correction	In the corrected domain
0.1	0.169 (199%)	0.0220 (24.4%)
0.05	0.114 (133%)	0.0115 (13.1%)
0.01	0.0471 (54.7%)	0.0026 (2.98%)

$$b(x) = (x_2\ x_3\ x_1)^*,$$

$$\sigma(x) = \begin{pmatrix} (1+|x_3|)^{1/2} & 0 & 0 \\ (\frac{1}{2}(1+|x_1|)^{1/2} & (\frac{3}{4}(1+|x_1|))^{1/2} & 0 \\ 0 & \frac{1}{2}(1+|x_2|)^{1/2} & (\frac{3}{4}(1+|x_2|))^{1/2} \end{pmatrix}$$

and $D = \{x \in \mathbb{R}^3 : |x| < 2\}$. Taking $k \equiv 0$,

$$g(x) = x_1x_2x_3$$

and

$$\begin{aligned} -f(x) =& x_2^2x_3 + x_3^2x_1 + x_1^2x_2 + \frac{1}{2}[x_3(1+|x_1|)^{1/2}(1+|x_3|)^{1/2} \\ &+ x_1\left(\frac{3}{4}\right)^{1/2}(1+|x_1|)^{1/2}(1+|x_2|)^{1/2}], \end{aligned}$$

we easily check that the solution is $u(x) = x_1x_2x_3$. For the initial points x, we take $(x_0^i)_{1\le i\le 3} \in \{-0.7, -0.3, 0.3, 0.7\}$. In Table 1, we report the supremum over the previous grid points of the absolute value of the absolute and relative errors for time step $\Delta \in \{0.01, 0.05, 0.1\}$. The number of Monte Carlo simulations are equal to 10^6, so that the width of the 95% confidence interval is about 2×10^{-3}.

It is clear that appropriately shifting the boundary much reduces the simulation bias. In [18] other tests related to option pricing are presented: they confirm the efficiency of this scheme.

Acknowledgements This work has been partly done when the author was affiliated to Grenoble INP-Ensimag. The author is grateful to *Grenoble Institute of Technology* (BQR grant entitled Monte Carlo) and to the Chair *Risques Financiers* of the *Fondation du Risque* for their financial support.

The author also thanks the two referees for their valuable comments and suggestions.

References

1. G. Allaire. *Shape optimization by the homogeneization method.* Springer Verlag, New-York, 1st edition, 2002.
2. L. Andersen and R. Brotherton-Ratcliffe. Exact exotics. *Risk*, 9:85–89, 1996.

3. P. Baldi. Exact asymptotics for the probability of exit from a domain and applications to simulation. *The Annals of Probability*, 23(4):1644–1670, 1995.
4. P. Baldi and L. Caramellino. Asymptotics of hitting probabilities for general one-dimensional diffusions. *Annals of Applied Probability*, 12:1071–1095, 2002.
5. V. Bally and D. Talay. 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.
6. T.R. Bielecki and M. Rutkowski. *Credit risk: modelling, valuation and hedging*. Springer Finance. Springer-Verlag, Berlin, 2002.
7. P.P. Boyle and S.H. Lau. Bumping up against the barrier with the binomial method. *Journal of Derivatives*, 1:6–14, 1994.
8. M. Broadie, P. Glasserman, and S. Kou. A continuity correction for discrete barrier options. *Mathematical Finance*, 7:325–349, 1997.
9. M. Broadie, P. Glasserman, and S. Kou. Connecting discrete and continuous path-dependent options. *Finance and Stochastics*, 3:55–82, 1999.
10. B. Casella and G.O. Roberts. Exact Monte Carlo simulation of killed diffusions. *Adv. in Appl. Probab.*, 40(1):273–291, 2008.
11. P. Cattiaux. Hypoellipticité et hypoellipticité partielle pour les diffusions avec une condition frontière. *Ann. Inst. H. Poincaré Probab. Statist.*, 22(1):67–112, 1986.
12. J.T. Chang and Y. Peres. Ladder heights, Gaussian random walks and the Riemann zeta function. *Ann. Probab.*, 25(2):787–802, 1997.
13. C. Costantini, E. Gobet, and N. El Karoui. Boundary sensitivities for diffusion processes in time dependent domains. *Applied Mathematics and Optimization*, 54(2):159–187, 2006.
14. E.H.A. Dia and D. Lamberton. Continuity correction for barrier options in jump-diffusion models. Technical report, HAL: http://hal.archives-ouvertes.fr/hal-00547668/fr/,2010.
15. C.D. Fuh, S.F. Luo, and J.F. Yen. Pricing discrete path-dependent options under a double exponential jump-diffusion model. Technical report, 2011.
16. E. Gobet. Euler schemes for the weak approximation of killed diffusion. *Stochastic Processes and their Applications*, 87:167–197, 2000.
17. E. Gobet. Euler schemes and half-space approximation for the simulation of diffusions in a domain. *ESAIM: Probability and Statistics*, 5:261–297, 2001.
18. E. Gobet. *Handbook of Numerical Analysis, Vol. XV, Special Volume: Mathematical Modeling and Numerical Methods in Finance*, chapter Advanced Monte Carlo methods for barrier and related exotic options, pages 497–528. Elsevier, Netherlands: North-Holland, 2009.
19. E. Gobet and S. Menozzi. Exact approximation rate of killed hypoelliptic diffusions using the discrete Euler scheme. *Stochastic Processes and their Applications*, 112(2):201–223, 2004.
20. E. Gobet and S. Menozzi. Discrete sampling of functionals of Itô processes. *Séminaire de probabilités XL – Lecture Notes in Mathematics 1899 Springer Verlag*, pages 355–374, 2007.
21. E. Gobet and S. Menozzi. Stopped diffusion processes: boundary corrections and overshoot. *Stochastic Processes and Their Applications*, 120:130–162, 2010.
22. K.M. Jansons and G.D. Lythe. Multidimensional exponential timestepping with boundary test. *SIAM Journal on Scientific Computing*, 27(3):793–808, 2005.
23. C. Labart. *EDSR: analyse de discrétisation et résolution par méthodes de Monte Carlo adaptatives; Perturbation de domaines pour les options américaines*. PhD thesis, Ecole Polytechnique, http://www.cmap.polytechnique.fr/∼labart/index_files/these.pdf, 2007.
24. T.L. Lai, Y. Yao, and F. Aitsahlia. Corrected random walk approximations to free boundary problems in optimal stopping. *Adv. in Appl. Probab.*, 39(3):753–775, 2007.
25. G.M. Lieberman. *Second order parabolic differential equations*. World Scientific Publishing Co. Inc., River Edge, NJ, 1996.
26. D. Nualart. *Malliavin calculus and related topics*. Springer Verlag, second edition, 2006.
27. E.J. Pauwels. Smooth first-passage densities for one-dimensional diffusions. *Journal of applied probability*, 24(2):370–377, 1987.

28. P.V. Shevchenko. Addressing the bias in Monte Carlo pricing of multi-asset options with multiple barriers through discrete sampling. *Journal of Computational Finance*, 6(3):1–20, 2003.
29. D. Siegmund. Corrected diffusion approximations in certain random walk problems. *Adv. in Appl. Probab.*, 11(4):701–719, 1979.
30. D. Siegmund. *Sequential Analysis*. Springer, 1985.

Stochastic Approximation of Functions and Applications

Stefan Heinrich

Abstract We survey recent results on the approximation of functions from Sobolev spaces by stochastic linear algorithms based on function values. The error is measured in various Sobolev norms, including positive and negative degree of smoothness. We also prove some new, related results concerning integration over Lipschitz domains, integration with variable weights, and study tractability of generalized versions of indefinite integration and discrepancy.

1 Introduction and Preliminaries

In this paper we survey and discuss recent results from [5–8] and predecessors thereof, from a unifying point of view of approximation of functions by linear algorithms based on function values. The functions belong to a certain Sobolev space and the error is measured in the norm of another Sobolev space. The emphasis lies on stochastic approximation, but we also include the deterministic counterparts. We discuss upper and lower bounds, hence the complexity of approximation, and compare the deterministic and randomized setting. The algorithms that reach the optimal rates are explained in detail.

The paper also contains a number of new results which are related to the known ones surveyed here. This includes the optimal order of the error of randomized integration of functions from Sobolev classes over general bounded Lipschitz domains, weighted integration with variable weights from Sobolev classes, approximation in certain spaces of functions with dominating mixed derivatives, and a result on the dimension-dependence (tractability) of generalized versions of indefinite integration and discrepancy.

S. Heinrich
Department of Computer Science, University of Kaiserslautern, D-67653 Kaiserslautern, Germany
e-mail: heinrich@informatik.uni-kl.de

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_5, © Springer-Verlag Berlin Heidelberg 2012

Let $d \in \mathbb{N} := \{1, 2, \dots\}$, $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$, let $\mathbb{K}$ stand for the field of reals $\mathbb{R}$ or complex numbers $\mathbb{C}$. We always consider $\mathbb{K}$-valued functions and linear spaces over $\mathbb{K}$, with $\mathbb{K}$ being the same for all the spaces involved. For a Banach space X the unit ball $\{x \in X : \|x\| \le 1\}$ is denoted by $\mathscr{B}_X$ and the dual space by X^*. Given another Banach space Y, the space of bounded linear operators from X to Y is denoted by $\mathscr{L}(X, Y)$. Throughout the paper log means $\log_2$. Furthermore, we often use the same symbol $c, c_1, \dots$ for possibly different positive constants, also when they appear in a sequence of relations.

Let $(G, \mathscr{G}, \mu)$ be a measure space. For $1 \le p \le \infty$, let $L_p(G, \mu)$ be the space of $\mathbb{K}$-valued p-integrable functions, equipped with the usual norm

$$\|f\|_{L_p(G,\mu)} = \left(\int_G |f(x)|^p d\mu(x)\right)^{1/p}$$

if $p < \infty$, and

$$\|f\|_{L_\infty(G,\mu)} = \operatorname{ess\,sup}_{x \in G} |f(x)|.$$

Let $Q \subset \mathbb{R}^d$ be a bounded Lipschitz domain, i.e., for $d = 1$ a finite union of bounded open intervals with disjoint closure, and for $d \ge 2$ a bounded open set with locally Lipschitz boundary. If μ is the Lebesgue measure on Q, we write $L_p(Q)$ instead of $L_p(Q, \mu)$. Let $C(\bar{Q})$ denote the space of continuous functions on the closure $\bar{Q}$ of Q, equipped with the supremum norm. For $r \in \mathbb{N}_0$ and $1 \le p \le \infty$ we introduce the Sobolev space

$$W_p^r(Q) = \{f \in L_p(Q) : \; D^\alpha f \in L_p(Q), \; |\alpha| \le r\},$$

where $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$, $|\alpha| := \sum_{j=1}^d \alpha_j$, and $D^\alpha f$ is the generalized partial derivative. The norm on $W_p^r(Q)$ is defined as

$$\|f\|_{W_p^r(Q)} = \left(\sum_{|\alpha| \le r} \|D^\alpha f\|_{L_p(Q)}^p\right)^{1/p}$$

if $p < \infty$, and

$$\|f\|_{W_\infty^r(Q)} = \max_{|\alpha| \le r} \|D^\alpha f\|_{L_\infty(Q)}.$$

Observe that for $r = 0$, $W_p^0(Q)$ is just $L_p(Q)$.

For basic notions concerning the randomized setting of information-based complexity – the framework we use here – we refer to [4, 14, 20]. The particular notation applied here can be found in [6].

First we consider deterministic algorithms. Let G be a nonempty set, let $\mathscr{F}(G)$ denote the linear space of all $\mathbb{K}$-valued functions on G and let Y be a Banach space. Given a nonempty subset $F \subseteq \mathscr{F}(G)$, the class of linear deterministic algorithms $\mathscr{A}_n^{\det}(F, Y)$ consists of all linear operators from $\mathscr{F}(G)$ to Y of the form

$$Af = \sum_{i=1}^{n} f(x_i)\psi_i$$

with $x_i \in G$ and $\psi_i \in Y$. Let $S : F \to Y$ be any mapping. The error of $A \in \mathscr{A}_n^{\mathrm{det}}(F, Y)$ as an approximation of S is defined as

$$e(S, A, F, Y) = \sup_{f \in F} \|Sf - Af\|_Y$$

and the deterministic n-th minimal error as

$$e_n^{\mathrm{det}}(S, F, Y) = \inf_{A \in \mathscr{A}_n^{\mathrm{det}}(F,Y)} e(S, A, F, Y).$$

Hence, no deterministic linear algorithm that uses at most n function values can provide a smaller error than $e_n^{\mathrm{det}}(S, F, Y)$. The quantities $e_n^{\mathrm{det}}(S, F, Y)$ were also called linear sampling numbers [15].

Next we introduce linear randomized sampling algorithms. This is a little more technical since we want these algorithms to act also on spaces of equivalence classes of functions, where function values itself may not be defined. Here we let, in addition to the above, $\mathscr{G}$ be a σ-algebra of subsets of G, μ a nonnegative, σ-additive, σ-finite measure on $(G, \mathscr{G})$ with $\mu(G) > 0$. Let $F \subseteq L_0(G, \mu)$ be a nonempty subset, where $L_0(G, \mu)$ is the linear space of equivalence classes of $\mathscr{G}$-measurable functions on G, with the usual equivalence of being equal except on a set of μ-measure zero.

For $n \in \mathbb{N}$ we consider the following class of randomized linear algorithms from F to Y. An element

$$A \in \mathscr{A}_n^{\mathrm{ran}}(F, Y)$$

is a tuple

$$A = ((\Omega, \Sigma, \mathbb{P}), (A_\omega)_{\omega \in \Omega}),$$

where $(\Omega, \Sigma, \mathbb{P})$ is a probability space,

$$A_\omega \in \mathscr{A}_n^{\mathrm{det}}(\mathscr{F}(G), Y) \quad (\omega \in \Omega),$$

thus

$$A_\omega f = \sum_{i=1}^{n} f(x_{i\omega})\psi_{i\omega} \quad (\omega \in \Omega),$$

and the following two properties hold:

1. (Consistency) If f_0 and f_1 are representatives of the same class $f \in F$, then

$$A_\omega f_0 = A_\omega f_1 \quad (\mathbb{P} - \text{almost surely}). \tag{1}$$

2. (Measurability) For each $f \in F$ and each representative f_0 of f, the mapping

$$\omega \in \Omega \to A_\omega f_0 \in Y \quad \text{is } \Sigma\text{-to-Borel measurable} \tag{2}$$

and essentially separably valued, i.e., there is a separable subspace $Y_0 \subseteq Y$ such that

$$A_\omega f_0 \in Y_0 \quad (\mathbb{P} - \text{almost surely}). \tag{3}$$

Let again $S : F \to Y$ be any mapping. The error of an algorithm $A \in \mathscr{A}_n^{\mathrm{ran}}(F, Y)$ as an approximation to S on F is defined as

$$e(S, A, F, Y) = \sup_{f \in F} \mathbb{E}\, \|Sf - A_\omega f\|_Y .$$

The randomized n-th minimal error of S is defined as

$$e_n^{\mathrm{ran}}(S, F, Y) = \inf_{A \in \mathscr{A}_n^{\mathrm{ran}}(F,Y)} e(S, A, F, Y).$$

It follows that no randomized linear algorithm that uses at most n function values can have a smaller error than $e_n^{\mathrm{ran}}(S, F, Y)$. Note that the definition involves the first moment. This way lower bounds have the strongest form, because respective bounds for higher moments follow by Hölder's inequality. Upper bounds for concrete algorithms are stated in a form which includes possible estimates of higher moments.

We need some notions and facts from probability theory in Banach spaces. Let $1 \le p \le 2$. An operator $T \in \mathscr{L}(X, Y)$ is said to be of type p if there is a constant $c > 0$ such that for all $n \in \mathbb{N}$ and all sequences $(g_i)_{i=1}^n \subset X$,

$$\mathbb{E} \left\| \sum_{i=1}^{n} \varepsilon_i T g_i \right\|^p \le c^p \sum_{i=1}^{n} \|g_i\|^p, \tag{4}$$

where (ε_i) is a sequence of independent random variables on some probability space $(\Omega, \Sigma, \mathbb{P})$ with $\mathbb{P}\{\varepsilon_i = 1\} = \mathbb{P}\{\varepsilon_i = -1\} = 1/2$. The type p constant $\tau_p(T)$ of the operator T is defined to be the smallest $c > 0$ such that (4) holds. We put $\tau_p(T) = \infty$ if T is not of type p. Each operator is of type 1. A Banach space X is of type p iff the identity operator of X is of type p. We write $\tau_p(X)$ for the type p constant of the identity operator of X. For $1 \le p < \infty$ the spaces ℓ_p^n are uniformly of type $\min(p, 2)$, meaning that there is a constant $c(p) > 0$ such that for all $n \in \mathbb{N}$ we have $\tau_{\min(p,2)}(\ell_p^n) \le c(p)$. For $p = \infty$ there is a constant $c(\infty) > 0$ such that $\tau_2(\ell_\infty^n) \le c(\infty)(\log n + 1)^{1/2}$ for all $n \in \mathbb{N}$. We refer to [12], Chap. 9 for definitions and basic facts on the type of Banach spaces, from which the operator analogues easily follow.

We will use the following result, see [8], Lemma 3.2. (the Banach space case of which with $p_1 = p$ is contained in Proposition 9.11 of [12]).

Lemma 1. *Let $1 \le p \le 2$, $p \le p_1 < \infty$. Then there is a constant $c = c(p, p_1) > 0$ such that for all Banach spaces X, Y, each operator $T \in \mathscr{L}(X, Y)$ of type p, each $n \in \mathbb{N}$ and each sequence of independent, mean zero X-valued random variables $(\eta_i)_{i=1}^n$ with $\mathbb{E}\,\|\eta_i\|^{p_1} < \infty$ $(1 \le i \le n)$ the following holds:*

$$\left(\mathbb{E}\left\|\sum_{i=1}^n T\eta_i\right\|^{p_1}\right)^{1/p_1} \le c\tau_p(T)\left(\sum_{i=1}^n \left(\mathbb{E}\,\|\eta_i\|^{p_1}\right)^{p/p_1}\right)^{1/p}.$$

2 Approximation of the Embedding $J : W_p^r(Q) \to W_q^s(Q)$ with $s \ge 0$

In this section we consider approximation of the embedding $J : W_p^r(Q) \to W_q^s(Q)$. By the Sobolev embedding theorem, [1], Theorem 5.4, J is continuous if

$$\left.\begin{array}{llll} & 1 \le q < \infty & \text{and} & \frac{r-s}{d} \ge \left(\frac{1}{p} - \frac{1}{q}\right)_+ \\ \text{or} & & & \\ & q = \infty, \quad 1 < p < \infty, & \text{and} & \frac{r-s}{d} > \frac{1}{p} \\ \text{or} & & & \\ & q = \infty, \quad p \in \{1, \infty\}, & \text{and} & \frac{r-s}{d} \ge \frac{1}{p}. \end{array}\right\} \tag{5}$$

We shall study $e_n^{\det}(J, \mathscr{B}_{W_p^r(Q)}, W_p^s(Q))$ and $e_n^{\text{ran}}(J, \mathscr{B}_{W_p^r(Q)}, W_p^s(Q))$, so we want to approximate functions from $W_p^r(Q)$ in the norm of $W_q^s(Q)$ by deterministic or randomized linear sampling algorithms based on n function values.

We also need the so-called embedding condition, ensuring that $W_p^r(Q)$ is continuously embedded into $C(\bar{Q})$ (meaning that each equivalence class contains a continuous representative). This holds if and only if

$$\left.\begin{array}{llll} & p = 1 & \text{and} & r/d \ge 1 \\ \text{or} & & & \\ & 1 < p \le \infty & \text{and} & r/d > 1/p, \end{array}\right\} \tag{6}$$

see [1], Chap. 5. In these cases function values at points of Q are well-defined and deterministic algorithms as introduced in Sect. 1 make sense.

In its full generality, the following was shown in [6], Theorems 3.1 and 4.2.

Theorem 1. *Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$, let Q be a bounded Lipschitz domain and assume that (5) is satisfied. Then there are constants $c_{1-6} > 0$ such that for all $n \in \mathbb{N}$ the following holds. In the deterministic setting, if the embedding condition (6) is fulfilled, then*

$$c_1 n^{-\frac{r-s}{d}+\left(\frac{1}{p}-\frac{1}{q}\right)_+} \le e_n^{\det}(J, \mathscr{B}_{W_p^r(Q)}, W_q^s(Q)) \le c_2 n^{-\frac{r-s}{d}+\left(\frac{1}{p}-\frac{1}{q}\right)_+},$$

and if the embedding condition is not fulfilled, then

$$c_3 \le e_n^{\mathrm{det}}(J, \mathscr{B}_{W_p^r(Q)} \cap C(\bar{Q}), W_q^s(Q)) \le c_4.$$

In the randomized setting we have

$$c_5 n^{-\frac{r-s}{d}+\left(\frac{1}{p}-\frac{1}{q}\right)_+} \le e_n^{\mathrm{ran}}(J, \mathscr{B}_{W_p^r(Q)}, W_q^s(Q)) \le c_6 n^{-\frac{r-s}{d}+\left(\frac{1}{p}-\frac{1}{q}\right)_+},$$

independently of the embedding condition.

To explain the occurring exponent in a few words: we can consider $n^{-(r-s)/d}$ as a 'reward' for decay in smoothness by going from $W_p^r(Q)$ to $W_q^s(Q)$, while $n^{1/p-1/q}$ is the 'price' we have to pay for the improvement of summability from p to q if $p < q$.

In various particular aspects and special cases Theorem 1 has many authors.

1. Deterministic setting, the embedding condition (6) holds:
   For simple domains as $Q = (0, 1)^d$ and $s = 0$, the bounds are classical approximation theory. For $Q = (0, 1)^d$ and $s > 0$, see Vybíral [22]. The general case of Lipschitz domains for $s = 0$ is due to Novak and Triebel [15]. The case of Lipschitz domains for $s > 0$ was obtained in [6], solving Problem 18 posed by Novak and Woźniakowski in [16].
2. Deterministic setting, the embedding condition (6) does not hold:
   This means, function values are not well-defined, so, formally, deterministic algorithms make no sense. However, we may just slightly restrict the class by considering $\mathscr{B}_{W_p^r(Q)} \cap C(\bar{Q})$ to make function values well-defined. Note that by considering $\mathscr{B}_{W_p^r(Q)} \cap C(\bar{Q})$ we do not impose a $C(\bar{Q})$ norm restriction, we only demand that the function is continuous, but it can have an arbitrary large $C(\bar{Q})$ norm. Such functions are dense in $B_{W_p^r(Q)}$ in the norm of $W_p^r(Q)$ (see [1], Theorem 3.18).

   Although function values are now well-defined, the result above shows that no deterministic algorithm can have an error converging to zero. This result was already proved in [5] for the cube.
3. Randomized setting, the embedding condition (6) holds:
   The upper bound follows from the deterministic setting. The lower bound was shown by Wasilkowski in [23] ($p = q = \infty$), Novak [14] ($1 \le p \le \infty$, $q = \infty$), and Mathé [13] ($1 \le p, q \le \infty$). It follows that in the case of the embedding condition deterministic and stochastic algorithms have the same optimal rate, that is, randomization does not provide a speedup.
4. Randomized setting, the embedding condition (6) does not hold:
   This was shown in [6]. Comparing deterministic and randomized setting we see that in this case randomization can give a speedup of up to $n^{-\beta}$ for any β with $0 < \beta < 1$. Indeed, for $p = q = 1$, $s = 0$, the maximal exponent of the speedup is r/d, which can be arbitrarily close to 1.

Let us describe the algorithm behind Theorem 1, following essentially the exposition in [6]. Fix parameters $\rho \in \mathbb{N}_0$, $\rho \geq r-1$, and $0 \leq \delta < 1$, let $\mathscr{P}_\rho$ denote the space of polynomials on $\mathbb{R}^d$ of degree not exceeding ρ, and let $P : \mathscr{F}(\mathbb{R}^d) \to \mathscr{F}(\mathbb{R}^d)$ be the d-fold tensor product of Lagrange interpolation on $[0, 1-\delta]$ of degree ρ, hence

$$Pf = \sum_{j=1}^{\kappa} f(y_j)\psi_j,$$

with $(y_j)_{j=1}^{\kappa} \in [0, 1-\delta]^d$ and $(\psi_j)_{j=1}^{\kappa}$ the respective Lagrange polynomials. We have

$$Pg = g \quad (g \in \mathscr{P}_\rho). \tag{7}$$

Let $\xi = \xi(\omega)$ $(\omega \in \Omega)$ be a uniformly distributed on $[0,1]^d$ random variable on a complete probability space $(\Omega, \Sigma, \mathbb{P})$. For $\omega \in \Omega$ define the operator $P_\omega : \mathscr{F}([0,1]^d) \to \mathscr{F}(\mathbb{R}^d)$ by setting for $f \in \mathscr{F}([0,1]^d)$

$$(P_\omega f)(x) = \sum_{j=1}^{\kappa} f(y_j + \delta\xi(\omega))\, \psi_j(x - \delta\xi(\omega)) \quad (x \in \mathbb{R}^d). \tag{8}$$

Note that if $\delta = 0$, then P_ω is deterministic, i.e., does not depend on ω. It follows from (7) that

$$P_\omega g = g \qquad (g \in \mathscr{P}_\rho,\ \omega \in \Omega). \tag{9}$$

We include Q into any axis-parallel cube $\tilde{Q}$,

$$Q \subset \tilde{Q} = x_0 + [0,b]^d,$$

and partition $\tilde{Q}$ into closed subcubes of sidelength $b2^{-l}$ and of disjoint interior

$$\tilde{Q} = \bigcup_{i=1}^{2^{dl}} Q_{li}.$$

For $l \in \mathbb{N}_0$ we define the scaling operators $E_{li}, R_{li} : \mathscr{F}(\mathbb{R}^d) \to \mathscr{F}(\mathbb{R}^d)$ for $f \in \mathscr{F}(\mathbb{R}^d)$ and $x \in \mathbb{R}^d$ by

$$(E_{li} f)(x) = f(x_{li} + b2^{-l}x)$$

and

$$(R_{li} f)(x) = f(b^{-1}2^l(x - x_{li})),$$

where x_{li} denote the point in Q_{li} with minimal coordinates. Note that E_{li} scales functions with support in Q_{li} to functions with support in $[0,1]^d$, and R_{li} is the inverse of E_{li}.

Define

$$\mathscr{I}_l = \{i \,:\, 1 \le i \le 2^{dl},\ Q_{li} \subseteq Q\},$$

the set if indices of cubes completely contained in Q, and

$$\mathscr{K}_l = \{k \,:\, 1 \le k \le 2^{dl},\ Q_{lk} \cap Q \neq \emptyset\},$$

the set of indices of cubes intersecting Q. So we have

$$\bigcup_{i \in \mathscr{I}_l} Q_{li} \subset Q \subset \bigcup_{k \in \mathscr{K}_l} Q_{lk}. \tag{10}$$

Let $B(x,\rho)$ denote the closed and $B^0(x,\rho)$ the open Euclidean ball of radius ρ around $x \in \mathbb{R}^d$. Based on the geometry of the Lipschitz property of Q the following was shown in [7], Lemma 3.1, see also [6], Lemma 3.2.

Lemma 2. *There are constants $a > b\sqrt{d}$ and $l_0 \in \mathbb{N}_0$ such that for all $l \ge l_0$*

$$\bigcup_{k \in \mathscr{K}_l} Q_{lk} \subseteq \bigcup_{i \in \mathscr{I}_l} B(x_{li}, a2^{-l}).$$

Using this lemma one can construct a suitable partition of unity on Q. Let $\sigma \in \mathbb{N}_0$, $\sigma \ge s$, and denote the space of functions possessing continuous, bounded partial derivatives up to order σ on $\mathbb{R}^d$ by $C^\sigma(\mathbb{R}^d)$. Let $\eta \in C^\sigma(\mathbb{R}^d)$ be such that supp $(\eta) \subseteq B^0(0, 2a/b)$, $\eta \ge 0$, and $\eta > 0$ on $B(0, a/b)$. We can choose η to be a polynomial on some ball around 0, for example

$$\eta(x) = \begin{cases} \left(\dfrac{9a^2}{4b^2} - \displaystyle\sum_{i=1}^{d} x_i^2\right)^{\sigma+1} & \text{if} \quad |x| \le \frac{3a}{2b} \\ 0 & \text{otherwise.} \end{cases}$$

By Lemma 2, there exists a constant $c > 0$ such that for $l \ge l_0$

$$\sum_{j \in \mathscr{I}_l} R_{lj}\eta(x) \ge c \qquad (x \in Q).$$

Define the functions η_{li} $(i \in \mathscr{I}_l, l \ge l_0)$ on Q by

$$\eta_{li}(x) = \frac{R_{li}\eta(x)}{\sum_{j \in \mathscr{I}_l} R_{lj}\eta(x)} \qquad (x \in Q).$$

These functions satisfy

$$\eta_{li}(x) = 0 \quad (x \in Q \setminus B(x_{li}, a2^{-l+1}))$$

and

$$\sum_{i\in\mathscr{I}_l}\eta_{li}(x)=1\qquad(x\in Q).$$

Now we define for $l \ge l_0$ and $\omega \in \Omega$ the operator $P_{l,\omega}^{(0)}:\mathscr{F}(Q)\to C^\sigma(Q)$ by

$$P_{l,\omega}^{(0)}f=\sum_{i\in\mathscr{I}_l}\eta_{li}(R_{li}P_\omega E_{li}f)|_Q\qquad(f\in\mathscr{F}(Q)).$$

Setting for $l \ge l_0$, $i \in \mathscr{I}_l$, $1 \le j \le \kappa$, and $\omega \in \Omega$

$$y_{lij\omega}=x_{li}+b\,2^{-l}(y_j+\delta\xi(\omega)) \tag{11}$$

and

$$\psi_{lij\omega}(x)=\psi_j(b^{-1}2^l(x-x_{li})-\delta\xi(\omega)), \tag{12}$$

we can finally write $P_{l,\omega}^{(0)}f$ as

$$P_{l,\omega}^{(0)}f=\sum_{i\in\mathscr{I}_l}\sum_{j=1}^{\kappa}f(y_{lij\omega})\eta_{li}\psi_{lij\omega}.$$

This completes the description of the algorithm leading to the upper bound in Theorem 1.

The algorithm above uses the partition of unity for smoothing the local approximations. In the case $s = 0$ the target space is $L_q(Q)$ and we do not need smoothing. In view of the application to integration given in the next section, we want to discuss this case in more detail and introduce a simpler algorithm. Using Lemma 2, we choose for $l \ge l_0$ any partition

$$\mathscr{K}_l=\bigcup_{i\in\mathscr{I}_l}\mathscr{K}_{li} \tag{13}$$

with

$$i\in\mathscr{K}_{li}\quad(i\in\mathscr{I}_l), \tag{14}$$

$$Q_{lk}\subseteq B(x_{li},a2^{-l})\quad(k\in\mathscr{K}_{li}), \tag{15}$$

$$\mathscr{K}_{li}\cap\mathscr{K}_{lj}=\emptyset\quad(i,j\in\mathscr{I}_l,\ i\ne j). \tag{16}$$

In other words, each cube Q_{lk} which intersects Q is associated with some cube Q_{li} which is not far from Q_{lk} and lies completely inside Q. The union of all cubes associated with Q_{li} is denoted by

$$\tilde{Q}_{li}=\bigcup_{k\in\mathscr{K}_{li}}Q_{lk}. \tag{17}$$

Now we proceed as follows. We apply approximating operators locally to the Q_{li} with $i \in \mathscr{I}_l$ and use the result (which is a polynomial defined on all of $\mathbb{R}^d$) for all the associated cubes Q_{lk} with $k \in \mathscr{K}_{li}$, that is, for the region $\tilde{Q}_{li}$. For $l \geq l_0$ and $\omega \in \Omega$ we define $P_{l,\omega}^{(1)} : \mathscr{F}(Q) \to L_q(Q)$ by

$$P_{l,\omega}^{(1)} f = \sum_{i \in \mathscr{I}_l} \chi_{\tilde{Q}_{li} \cap Q} R_{li} P_\omega E_{li} f \qquad (f \in \mathscr{F}(Q)), \tag{18}$$

which we can write as

$$P_{l,\omega}^{(1)} f = \sum_{i \in \mathscr{I}_l} \sum_{j=1}^{\kappa} f(y_{lij\omega}) \chi_{\tilde{Q}_{li} \cap Q} \psi_{lij\omega}, \tag{19}$$

with the $y_{lij\omega}$ and $\psi_{lij\omega}$ given by (11) and (12). Consistency (1) of $\big(P_{l,\omega}^{(1)}\big)_{\omega \in \Omega}$ is readily checked. As to measurability, note that we can represent

$$\begin{aligned} \psi_{lij\omega}(x) &= \psi_j(b^{-1} 2^l (x - x_{li}) - \delta \xi(\omega)) \\ &= \sum_{m=1}^{M} a_{jm}(\delta \xi(\omega)) \varphi_m(b^{-1} 2^l (x - x_{li})) \end{aligned} \tag{20}$$

with suitable $M \in \mathbb{N}$ and polynomials a_{jm}, φ_m $(1 \leq j \leq \kappa,\ 1 \leq m \leq M)$, from which (2) and (3) directly follow. So we have

$$\big(P_{l,\omega}^{(1)}\big)_{\omega \in \Omega} \in \mathscr{A}_{n_l}^{\text{ran}}(W_p^r(Q), L_q(Q)) \quad \text{with} \quad n_l = \kappa |\mathscr{I}_l|. \tag{21}$$

The following result generalizes Proposition 1 of [5] by combining the approach of Proposition 3.3 in [6] with that of Lemma 3.2 in [7]. It will be used for variance reduction in Sect. 3.

Proposition 1. *Let $d \in \mathbb{N}$, $r \in \mathbb{N}_0$, $1 \leq p, q \leq \infty$, let Q be a bounded Lipschitz domain, and assume that (5) is satisfied with $s = 0$. Let $(P_{l,\omega}^{(1)})_{\omega \in \Omega}$ for $l \geq l_0$ be given by (19), with parameters $\rho \in \mathbb{N}_0$, $\rho \geq r - 1$ and $0 \leq \delta < 1$. Moreover, if the embedding condition (6) does not hold, we assume $\delta > 0$. Then there is a constant $c > 0$ such that for all $l \geq l_0$ and $f \in W_p^r(Q)$ the following hold.*

If $q < \infty$, then

$$(\mathbb{E} \, \| f - P_{l,\omega}^{(1)} f \|_{L_q(Q)}^q)^{1/q} \leq c \, 2^{-rl + \max(1/p - 1/q, 0) dl} \| f \|_{W_p^r(Q)}, \tag{22}$$

and if $q = \infty$, then

$$\operatorname{ess\,sup}_{\omega \in \Omega} \| f - P_{l,\omega}^{(1)} f \|_{L_\infty(Q)} \leq c \, 2^{-rl + dl/p} \| f \|_{W_p^r(Q)}. \tag{23}$$

Proof. We put $B = B^0(0, 2a/b)$. By assumption, (5) holds for $s = 0$, so we have

$$\|f\|_{L_q(B)} \le c\|f\|_{W_p^r(B)} \quad (f \in W_p^r(B)). \tag{24}$$

Assume $q < \infty$. First we show that for $f \in W_p^r(B)$

$$\left(\mathbb{E}\,\|P_\omega f\|_{L_q(B)}^q\right)^{1/q} \le c\|f\|_{W_p^r(B)}. \tag{25}$$

Indeed, by (8) we have

$$\begin{aligned}
&\left(\mathbb{E}\,\|P_\omega f\|_{L_q(B)}^q\right)^{1/q} \\
&\le \left(\mathbb{E}\left(\sum_{j=1}^{\kappa} |f(y_j + \delta\xi(\omega))| \|\psi_j(\cdot - \delta\xi(\omega))\|_{L_q(B)}\right)^q\right)^{1/q} \\
&\le c\sum_{j=1}^{\kappa} (\mathbb{E}\,|f(y_j + \delta\xi(\omega))|^q)^{1/q}.
\end{aligned} \tag{26}$$

If $\delta > 0$, it follows from (24) that

$$\begin{aligned}
\sum_{j=1}^{\kappa} (\mathbb{E}\,|f(y_j + \delta\xi(\omega))|^q)^{1/q} &= \sum_{j=1}^{\kappa}\left(\delta^{-d}\int_{[0,\delta]^d} |f(y_j + z)|^q dz\right)^{1/q} \\
&\le c\|f\|_{L_q(B)} \le c\|f\|_{W_p^r(B)},
\end{aligned}$$

which together with (26) gives (25). If $\delta = 0$, which, by assumption, is only admitted if the embedding condition (6) holds, we have

$$\sum_{j=1}^{\kappa} (\mathbb{E}\,|f(y_j + \delta\xi(\omega))|^q)^{1/q} = \sum_{j=1}^{\kappa} |f(y_j)| \le \kappa\|f\|_{C(\bar{B})} \le c\|f\|_{W_p^r(B)},$$

which combined with (26) again implies (25). Using Theorem 3.1.1 of [2], it follows that there is a constant $c > 0$ such that for all $f \in W_p^r(B)$

$$\inf_{g \in \mathscr{P}_\rho} \|f - g\|_{W_p^r(B)} \le c|f|_{r,p,B}, \tag{27}$$

where

$$|f|_{r,p,B} = \left(\sum_{|\alpha|=r} \|D^\alpha f\|_{L_p(B)}^p\right)^{1/p}$$

if $p < \infty$ and

$$|f|_{r,\infty,B} = \max_{|\alpha|=r} \|D^\alpha f\|_{L_\infty(B)}.$$

We get from (9), (24), (25), and (27)

$$\begin{aligned}(\mathbb{E}\,\|f - P_\omega f\|_{L_q(B)}^q)^{1/q} &= \inf_{g\in\mathscr{P}_\rho} \Big(\mathbb{E}\,\|(f-g) - P_\omega(f-g)\|_{L_q(B)}^q\Big)^{1/q} \\ &\le c \inf_{g\in\mathscr{P}_\rho} \|f-g\|_{W_p^r(B)} \le c|f|_{r,p,B}. \qquad (28)\end{aligned}$$

Now let $f \in W_p^r(Q)$ and let $\tilde{f} \in W_p^r(\mathbb{R}^d)$ be an extension of f with

$$\|\tilde{f}\|_{W_p^r(\mathbb{R}^d)} \le c\|f\|_{W_p^r(Q)}$$

(see [19]). Then (10), (13), (16), and (17) imply

$$\begin{aligned}&(\mathbb{E}\,\|f - P_{l,\omega}^{(1)} f\|_{L_q(Q)}^q)^{1/q} \\ &= \left(\mathbb{E}\left\|\sum_{i\in\mathscr{I}_l} \chi_{\tilde{Q}_{li}\cap Q}(f - R_{li}P_\omega E_{li} f)\right\|_{L_q(Q)}^q\right)^{1/q} \\ &= \left(\sum_{i\in\mathscr{I}_l} \mathbb{E}\,\|f - R_{li}P_\omega E_{li} f\|_{L_q(\tilde{Q}_{li}\cap Q)}^q\right)^{1/q}. \qquad (29)\end{aligned}$$

Furthermore, from (15) and (28),

$$\begin{aligned}&\Big(\mathbb{E}\,\|f - R_{li}P_\omega E_{li} f\|_{L_q(\tilde{Q}_{li}\cap Q)}^q\Big)^{1/q} \\ &\le \Big(\mathbb{E}\,\|\tilde{f} - R_{li}P_\omega E_{li}\tilde{f}\|_{L_q(B(x_{li},a2^{-l}))}^q\Big)^{1/q} \\ &= b^{d/q}2^{-dl/q}\Big(\mathbb{E}\,\|E_{li}\tilde{f} - P_\omega E_{li}\tilde{f}\|_{L_q(B)}^q\Big)^{1/q} \\ &\le c\,2^{-dl/q}|E_{li}\tilde{f}|_{r,p,B}. \qquad (30)\end{aligned}$$

Using Hölder's inequality, we get for $p < \infty$

$$\begin{aligned}&\left(2^{-dl}\sum_{i\in\mathscr{I}_l} |E_{li}\tilde{f}|_{r,p,B}^q\right)^{1/q} \\ &\le c\,2^{\max(1/p-1/q,0)dl}\left(2^{-dl}\sum_{i\in\mathscr{I}_l} |E_{li}\tilde{f}|_{r,p,B}^p\right)^{1/p}\end{aligned}$$

$$
\begin{aligned}
&\le c\,2^{-rl+\max(1/p-1/q,0)dl}\left(\sum_{i\in\mathscr{I}_l}|\tilde{f}|^p_{r,p,B(x_{li},a2^{-l})}\right)^{1/p}\\
&\le c\,2^{-rl+\max(1/p-1/q,0)dl}\,|\tilde{f}|_{r,p,\mathbb{R}^d}\\
&\le c\,2^{-rl+\max(1/p-1/q,0)dl}\,\|f\|_{W_p^r(Q)}. \qquad (31)
\end{aligned}
$$

The case $p=\infty$ follows in the same way with the respective changes. Joining (29)–(31) proves (22). For $q=\infty$, relation (23) follows analogously, with the usual modifications, replacing everywhere $(\mathbb{E}\,\|\cdot\|^q)^{1/q}$ by $\operatorname{ess\,sup}_{\omega\in\Omega}\|\cdot\|$ etc. □

3 Integration Over Lipschitz Domains

Let Q be a bounded Lipschitz domain as in the previous section and let $I: W_p^r(Q)\to\mathbb{K}$ be the integration operator

$$If=\int_Q f(x)dx.$$

Theorem 2. *Let $r\in\mathbb{N}_0$, $d\in\mathbb{N}$, $1\le p\le\infty$, $\bar{p}=\min(p,2)$. Then there exist constants $c_{1-6}>0$ such that in the deterministic setting, if the embedding condition (6) holds, then*

$$c_1 n^{-r/d}\le e_n^{\rm det}(I,\mathscr{B}_{W_p^r(Q)},\mathbb{K})\le c_2 n^{-r/d},$$

and if the embedding condition does not hold, then

$$c_3\le e_n^{\rm det}(I,\mathscr{B}_{W_p^r(Q)}\cap C(\bar{Q}),\mathbb{K})\le c_4.$$

In the randomized setting we have, independently of the embedding condition,

$$c_5 n^{-r/d-1+1/\bar{p}}\le e_n^{\rm ran}(I,\mathscr{B}_{W_p^r(Q)},\mathbb{K})\le c_6 n^{-r/d-1+1/\bar{p}}.$$

In the deterministic case with the embedding condition the upper bound is a direct consequence of [15], see also [21], Theorem 5.4. It also follows from Proposition 1 by integrating the deterministic approximation for $\delta=0$ (see (32) and (33) below, where this appears as part of the variance reduction). The lower bound for general Lipschitz domains is easily derived from that for the cube, which is well-known, see [14]. The lower bound in the deterministic case without the embedding condition follows from the proof of Theorem 5.2 in [7] (the upper bound is trivial, it is just the boundedness of I).

Let us turn to the randomized case. For the cube, this result is due to Novak for those r, d, p for which $W_p^r(Q)$ is embedded into $L_2(Q)$ (meaning that $p \geq 2$ or $(p < 2 \wedge r/d \geq 1/p - 1/2)$), see [14], Sect. 2.2.9. The remaining cases were settled for the cube in [5]. As in the deterministic case, the lower bound for general Lipschitz domains follows from the known one for the cube, see [14] and [4]. The extension of the upper bound to general Lipschitz domains is new and we give a proof here.

We start by introducing a randomized algorithm. Similar to [5], we use an approximation for variance reduction by separation of the main part, which we combine here with stratified sampling. We use $P_{l,\omega_1}^{(1)}$ for $l \geq l_0$, see relations (11), (12), and (19) for its definition, with l_0 the constant from Lemma 2. For the purpose of the present proof we have changed the notation of the underlying probability space to $(\Omega_1, \Sigma_1, \mathbb{P}_1)$. Again we assume $\delta > 0$ if the embedding condition (6) does not hold. For $f \in \mathscr{F}(Q)$ we have

$$\begin{aligned} IP_{l,\omega_1}^{(1)} f &= \sum_{i \in \mathscr{I}_l} \sum_{j=1}^{\kappa} f(y_{lij\omega_1}) \int_{\tilde{Q}_{li} \cap Q} \psi_{lij\omega_1}(x)dx \\ &= \sum_{i \in \mathscr{I}_l} \sum_{j=1}^{\kappa} \alpha_{lij\omega_1} f(y_{lij\omega_1}) \end{aligned} \tag{32}$$

with

$$\alpha_{lij\omega_1} = \int_{\tilde{Q}_{li} \cap Q} \psi_{lij\omega_1}(x)dx = \sum_{k \in \mathscr{K}_{li}} \int_{Q_{lk} \cap Q} \psi_{lij\omega_1}(x)dx. \tag{33}$$

Observe that for $\delta > 0$, this is a stochastic quadrature, with the only element of randomness being ξ, while for $\delta = 0$ it is deterministic (compare (11) and (12)).

Now let $\zeta_k = \zeta_k(\omega_2)$ $(k \in \mathscr{K}_l)$ be independent, uniformly distributed on Q_{lk} random variables over a complete probability space $(\Omega_2, \Sigma_2, \mathbb{P}_2)$. Define a stratified sampling algorithm $A_{l,\omega_2}^{(2)}$ by setting for $g \in \mathscr{F}(Q)$ and $\omega_2 \in \Omega_2$

$$A_{l,\omega_2}^{(2)} g = b^d 2^{-dl} \sum_{k \in \mathscr{K}_l} \chi_{Q_{lk} \cap Q}(\zeta_k(\omega_2)) g(\zeta_k(\omega_2)),$$

where we recall that $|Q_{lk}| = b^d 2^{-dl}$. It follows readily that (1)–(3) hold, so

$$\left(A_{l,\omega_2}^{(2)}\right)_{\omega_2 \in \Omega_2} \in \mathscr{A}_{m_l}^{\text{ran}}(L_p(Q), \mathbb{K}) \quad \text{with} \quad m_l = |\mathscr{K}_l|.$$

Moreover, for $g \in L_1(Q)$

$$\mathbb{E}\, A_{l,\omega_2}^{(2)} g = \sum_{k \in \mathscr{K}_l} \int_{Q_{lk}} \chi_{Q_{lk} \cap Q}(x) g(x)dx = \int_Q g(x)dx.$$

First we show an error estimate for $A^{(2)}_{l,\omega_2}$. The case $p < 2$ seems to be new. Moreover, in the case $p > 2$ we estimate higher moments than the usual second moment.

Lemma 3. *Let* $1 \le p \le \infty$*,* $p_1 \le p$*,* $p_1 < \infty$*. Then there is a constant* $c > 0$ *such that for* $l \ge l_0$ *and* $g \in L_p(Q)$

$$\left(\mathbb{E}_{\omega_2}|Ig - A^{(2)}_{l,\omega_2}g|^{p_1}\right)^{1/p_1} \le c2^{-(1-1/\bar{p})dl}\|g\|_{L_p(Q)}.$$

Proof. We can assume $\bar{p} \le p_1$, the other cases follow from Hölder's inequality. Setting for $k \in \mathscr{K}_l$

$$\theta_k = b^d 2^{-dl}\chi_{Q_{lk}\cap Q}(\zeta_k)g(\zeta_k),$$

we have

$$A^{(2)}_{l,\omega_2}g - Ig = \sum_{k\in\mathscr{K}_l}(\theta_k - \mathbb{E}\,\theta_k). \tag{34}$$

From Lemma 1, taking into account that $\mathbb{K}$ is of type 2, hence also of type $\bar{p}$, we get

$$\begin{aligned}
&\left(\mathbb{E}\Big|\sum_{k\in\mathscr{K}_l}(\theta_k - \mathbb{E}\,\theta_k)\Big|^{p_1}\right)^{1/p_1} \\
&\le c\left(\sum_{k\in\mathscr{K}_l}\left(\mathbb{E}\,|\theta_k - \mathbb{E}\,\theta_k|^{p_1}\right)^{\bar{p}/p_1}\right)^{1/\bar{p}} \\
&\le c|\mathscr{K}_l|^{1/\bar{p}-1/p_1}\left(\sum_{k\in\mathscr{K}_l}\mathbb{E}\,|\theta_k - \mathbb{E}\,\theta_k|^{p_1}\right)^{1/p_1}.
\end{aligned} \tag{35}$$

Furthermore,

$$\begin{aligned}
(\mathbb{E}\,|\theta_k - \mathbb{E}\,\theta_k|^{p_1})^{1/p_1} &\le 2(\mathbb{E}\,|\theta_k|^{p_1})^{1/p_1} \\
&= 2(b^d 2^{-dl})^{1-1/p_1}\left(\int_{Q_{lk}\cap Q}|g(x)|^{p_1}dx\right)^{1/p_1}.
\end{aligned} \tag{36}$$

Combining (34)–(36) and using $p_1 \le p$, we obtain

$$\begin{aligned}
&\left(\mathbb{E}\,|A^{(2)}_{l,\omega_2}g - Ig|^{p_1}\right)^{1/p_1} \\
&\le c|\mathscr{K}_l|^{1/\bar{p}-1/p_1}(b^d 2^{-dl})^{1-1/p_1}\left(\int_Q |g(x)|^{p_1}dx\right)^{1/p_1} \\
&\le c2^{-(1-1/\bar{p})dl}\|g\|_{L_p(Q)}.
\end{aligned}$$

□

Now we put

$$(\Omega, \Sigma, \mathbb{P}) = (\Omega_1, \Sigma_1, \mathbb{P}_1) \times (\Omega_2, \Sigma_2, \mathbb{P}_2)$$

and define the final algorithm $(A_{l,\omega})_{\omega\in\Omega}$ for $\omega = (\omega_1, \omega_2)$ and $f \in \mathscr{F}(Q)$ by setting

$$A_{l,\omega} f = I P^{(1)}_{l,\omega_1} f + A^{(2)}_{l,\omega_2}(f - P^{(1)}_{l,\omega_1} f), \tag{37}$$

thus, we separated the main part $P^{(1)}_{l,\omega_1} f$, integrated it exactly and applied stratified sampling to the remaining function $f - P^{(1)}_{l,\omega_1} f$. Writing this in more detail, we obtain

$$\begin{aligned} A_{l,\omega} f &= \sum_{i\in\mathscr{I}_l} \sum_{j=1}^{\kappa} \alpha_{lij\omega_1} f(y_{lij\omega_1}) \\ &\quad + b^d 2^{-dl} \sum_{k\in\mathscr{K}_l} \chi_{Q_{lk}\cap Q}(\zeta_k)\Big(f(\zeta_k) - (P^{(1)}_{l,\omega_1} f)(\zeta_k)\Big). \end{aligned}$$

We have

$$\begin{aligned} (P^{(1)}_{l,\omega_1} f)(\zeta_k) &= \sum_{i_1\in\mathscr{I}_l} \sum_{k_1\in\mathscr{K}_{li_1}} \sum_{j=1}^{\kappa} f(y_{li_1 j\omega_1}) \chi_{Q_{lk_1}\cap Q}(\zeta_k) \psi_{li_1 j\omega_1}(\zeta_k) \\ &= \sum_{j=1}^{\kappa} f(y_{l\iota(k)j\omega_1}) \chi_{Q_{lk}\cap Q}(\zeta_k) \psi_{l\iota(k)j\omega_1}(\zeta_k) \end{aligned}$$

for almost all $\omega_1 \in \Omega_1$, where $\iota(k)$ is the unique $i \in \mathscr{I}_l$ with $k \in \mathscr{K}_{li}$. Consequently,

$$\begin{aligned} A_{l,\omega} f = \sum_{i\in\mathscr{I}_l} \sum_{k\in\mathscr{K}_{li}} \Bigg(& \sum_{j=1}^{\kappa} f(y_{lij\omega_1}) \int_{Q_{lk}\cap Q} \psi_{lij\omega_1}(x) dx \\ & + b^d 2^{-dl} \chi_{Q_{lk}\cap Q}(\zeta_k) \Big(f(\zeta_k) - \sum_{j=1}^{\kappa} f(y_{lij\omega_1}) \psi_{lij\omega_1}(\zeta_k) \Big) \Bigg), \end{aligned}$$

with the $y_{lij\omega_1}$ and $\psi_{lij\omega_1}$ given by (11) and (12) and equality holding $\mathbb{P}$-almost surely. We have

$$\big(A_{l,\omega}\big)_{\omega\in\Omega} \in \mathscr{A}^{\mathrm{ran}}_{n_l}(W^r_p(Q), \mathbb{K}) \quad \text{with} \quad n_l = \kappa|\mathscr{I}_l| + |\mathscr{K}_l| \le c2^{dl}, \tag{38}$$

which can be checked in a similar way as (21), using (20).

Proposition 2. *Let $1 \le p \le \infty$, $p_1 \le p$, $p_1 < \infty$. Then there is a constant $c > 0$ such that for $l \ge l_0$ and $f \in W_p(Q)$*

$$(\mathbb{E}\,|If - A_{l,\omega} f|^{p_1})^{1/p_1} \le c2^{-rl-(1-1/\bar{p})dl} \|f\|_{W_p^r(Q)}.$$

Proof. We have

$$If - A_{l,\omega} f = I(f - P_{l,\omega_1}^{(1)} f) - A_{l,\omega_2}^{(2)}(f - P_{l,\omega_1}^{(1)} f).$$

Using Fubini's theorem, Lemma 3, and Proposition 1 for $q = p$, we get for $p < \infty$

$$\begin{aligned}
&(\mathbb{E}\,|If - A_{l,\omega} f|^{p_1})^{1/p_1} \\
&= \left(\mathbb{E}_{\omega_1}\mathbb{E}_{\omega_2}\left|I\left(f - P_{l,\omega_1}^{(1)} f\right) - A_{l,\omega_2}^{(2)}\left(f - P_{l,\omega_1}^{(1)} f\right)\right|^{p_1}\right)^{1/p_1} \\
&\le c2^{-(1-1/\bar{p})dl}\left(\mathbb{E}_{\omega_1}\left\|f - P_{l,\omega_1}^{(1)} f\right\|_{L_p(Q)}^{p_1}\right)^{1/p_1} \\
&\le c2^{-(1-1/\bar{p})dl}\left(\mathbb{E}_{\omega_1}\left\|f - P_{l,\omega_1}^{(1)} f\right\|_{L_p(Q)}^{p}\right)^{1/p} \qquad (39) \\
&\le c2^{-(1-1/\bar{p})dl-rl}\|f\|_{W_p^r(Q)}.
\end{aligned}$$

This also holds for $p = \infty$, if we replace in (39) $\left(\mathbb{E}_{\omega_1}\left\|f - P_{l,\omega_1}^{(1)} f\right\|_{L_p(Q)}^{p}\right)^{1/p}$ by $\operatorname{ess\,sup}_{\omega_1 \in \Omega_1}\|f - P_{l,\omega_1}^{(1)} f\|_{L_\infty(Q)}$, concluding the proof.

□

Now the upper bound in the randomized case of Theorem 2 is a direct consequence of Proposition 2 and (38).

4 Approximation of Embeddings into Spaces with Negative Degree of Smoothness

Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$. Let q^* be the dual index to q, given by $1/q + 1/q^* = 1$. Denote by $\tilde{W}_{q^*}^s(Q)$ the closure in the norm of $W_{q^*}^s(Q)$ of the set of C^∞ functions whose support is contained in Q and let $U : \tilde{W}_{q^*}^s(Q) \to W_{q^*}^s(Q)$ be the identical embedding. We consider two embedding operators

$$J : W_p^r(Q) \to W_{q^*}^s(Q)^*$$

given for $f \in W_p^r(Q)$ by

$$(Jf)(g) = \int_Q f(x)g(x)dx \quad (g \in W_{q^*}^s(Q))$$

and

$$\tilde{J} = U^*J : W_p^r(Q) \xrightarrow{J} W_{q^*}^s(Q)^* \xrightarrow{U^*} \tilde{W}_{q^*}^s(Q)^*. \tag{40}$$

We note that by definition, see [1], Sect. 3.11, for $1 < q \le \infty$ and $s > 0$

$$\tilde{W}_{q^*}^s(Q)^* = W_q^{-s}(Q). \tag{41}$$

Let us formulate conditions, under which J (and hence $\tilde{J}$) is well-defined and continuous. First let us state two auxiliary conditions.

$$r = 0,\ p = 1,\ 1 < q < \infty, \tag{42}$$

$$s = 0,\ q = \infty,\ 1 < p < \infty. \tag{43}$$

Now $J : W_p^r(Q) \to W_{q^*}^s(Q)^*$ is well-defined and continuous if

$$\left.\begin{array}{ll} & (42) \text{ holds and } \frac{s}{d} > \frac{1}{q^*}, \\ \text{or} & \\ & (43) \text{ holds and } \frac{r}{d} > \frac{1}{p}, \\ \text{or} & \\ & (42) \text{ and } (43) \text{ do not hold, and } \frac{r+s}{d} \ge \left(\frac{1}{p} - \frac{1}{q}\right)_+ . \end{array}\right\} \tag{44}$$

This follows from the Sobolev embedding theorem (5), see also [7], Sect. 4.

Next we want to give some motivation why to consider spaces with negative degree of smoothness $W_q^{-s}(Q)$. The space $W_2^{-s}(Q)$ plays a central role in the theory of elliptic partial differential equations, in connection with the weak form. Let $m \in \mathbb{N}$ and consider the bilinear form a on $\tilde{W}_2^m(Q)$, defined by

$$a(u, v) = \sum_{|\alpha|,|\beta| \le m} \int_Q a_{\alpha\beta}(x) D^\alpha u(x) D^\beta v(x) dx \quad (u, v \in \tilde{W}_2^m(Q)),$$

where $a_{\alpha\beta} \in C(\bar{Q})$. We assume that a is $\tilde{W}_2^m(Q)$-elliptic, meaning that

$$|a(u, v)| \le c_1 \|u\|_{W_2^m(Q)} \|v\|_{W_2^m(Q)}$$
$$a(u, u) \ge c_2 \|u\|^2_{W_2^m(Q)}$$

for $u, v \in \tilde{W}_2^m(Q)$. The weak elliptic problem associated with a is the following. Given $f \in W_2^{-m}(Q)$, find $u \in \tilde{W}_2^m(Q)$ such that for all $v \in \tilde{W}_2^m(Q)$

$$a(u, v) = f(v). \tag{45}$$

By ellipticity, the problem has a unique solution $S_0 f \in \tilde{W}_2^m(Q)$, and

$$S_0 : W_2^{-m}(Q) \to \tilde{W}_2^m(Q)$$

is an isomorphism. For $r \in \mathbb{N}_0$ we seek to solve the weak problem for $f \in W_2^r(Q)$. The solution operator, that is, the operator, which maps the problem instance $f \in W_2^r(Q)$ to the solution u of (45) is

$$S^{\mathrm{ell}} = S_0 \tilde{J} : W_2^r(Q) \xrightarrow{\tilde{J}} W_2^{-m}(Q) \xrightarrow{S_0} \tilde{W}_2^m(Q). \tag{46}$$

Since S_0 is an isomorphism, we immediately derive from (46) the connection to approximation of $\tilde{J}$:

Corollary 1. *Let $m \in \mathbb{N}$. Then there are constants $c_{1-4} > 0$ such that*

$$\begin{aligned} c_1 e_n^{\det}(\tilde{J}, \mathscr{B}_{W_2^r(Q)}, W_2^{-m}(Q)) &\le e_n^{\det}(S^{\mathrm{ell}}, \mathscr{B}_{W_2^r(Q)}, \tilde{W}_2^m(Q)) \\ &\le c_2 e_n^{\det}(\tilde{J}, \mathscr{B}_{W_2^r(Q)}, W_2^{-m}(Q)) \end{aligned}$$

and

$$\begin{aligned} c_3 e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{W_2^r(Q)}, W_2^{-m}(Q)) &\le e_n^{\mathrm{ran}}(S^{\mathrm{ell}}, \mathscr{B}_{W_2^r(Q)}, \tilde{W}_2^m(Q)) \\ &\le c_4 e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{W_2^r(Q)}, W_2^{-m}(Q)). \end{aligned}$$

We also consider approximation in the more general space $W_{q^*}^s(Q)^*$, because by (40) upper bounds are stronger, while the lower bound methods from [7] work equally for both cases $\tilde{W}_{q^*}^s(Q)^*$ and $W_{q^*}^s(Q)^*$.

Moreover, let us also point out an interesting connection to a problem of weighted integration, not with a fixed weight, but simultaneous integration over Sobolev classes of weights. We discuss this only briefly, leaving the detailed exploration open to future research.

First we consider the deterministic case. Let $A \in \mathscr{A}_n^{\det}(W_p^r(Q), W_{q^*}^s(Q)^*)$,

$$Af = \sum_{i=1}^n f(x_i)\psi_i,$$

with $x_i \in Q$ and $\psi_i \in W_{q^*}^s(Q)^*$ $(i = 1, \dots, n)$. We have

$$\begin{aligned}
&e(J, A, \mathscr{B}_{W_p^r(Q)}, W_{q^*}^s(Q)^*)\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \|Jf - Af\|_{W_{q^*}^s(Q)^*}\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \left\| Jf - \sum_{i=1}^n f(x_i)\psi_i \right\|_{W_{q^*}^s(Q)^*}\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \sup_{w \in \mathscr{B}_{W_{q^*}^s(Q)}} \left| \int_Q f(x)w(x)dx - \sum_{i=1}^n f(x_i)(\psi_i, w) \right|.
\end{aligned}$$

This way we approximate the weighted integral $\int_Q f(x)w(x)dx$ by a quadrature $\sum_{i=1}^n (\psi_i, w) f(x_i)$. The quadrature weights depend on the integration weight w only through n linear functionals, and the error is taken uniformly over the integrands f and weights w.

In the randomized case we let $A \in \mathscr{A}_n^{\text{ran}}(W_p^r(Q), W_{q^*}^s(Q)^*)$,

$$A = ((\Omega, \Sigma, \mathbb{P}), (A_\omega)_{\omega \in \Omega}),$$

$$A_\omega f = \sum_{i=1}^n f(x_{i,\omega})\psi_{i,\omega} \quad (\omega \in \Omega),$$

with $x_{i,\omega} \in Q$ and $\psi_{i,\omega} \in W_{q^*}^s(Q)^*$ $(i = 1, \dots, n, \omega \in \Omega)$. Then we have

$$\begin{aligned}
&e(J, A, \mathscr{B}_{W_p^r(Q)}, W_{q^*}^s(Q)^*)\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \mathbb{E} \|Jf - A_\omega f\|_{W_{q^*}^s(Q)^*}\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \mathbb{E} \left\| Jf - \sum_{i=1}^n f(x_{i,\omega})\psi_{i,\omega} \right\|_{W_{q^*}^s(Q)^*}\\
&= \sup_{f \in \mathscr{B}_{W_p^r(Q)}} \mathbb{E} \sup_{w \in \mathscr{B}_{W_{q^*}^s(Q)}} \left| \int_Q f(x)w(x)dx - \sum_{i=1}^n f(x_{i,\omega})(\psi_{i,\omega}, w) \right|.
\end{aligned}$$

Thus, similar to the deterministic case, we approximate $\int_Q f(x)w(x)dx$ by a quadrature, this time a stochastic one, and the quadrature weights depend on the integration weight w through n random linear functionals. Moreover, observe that the error criterion is different from the usual one in the randomized setting, namely, it is uniform over the class of weights.

After these motivations let us state the main results on approximation. In the deterministic case, we have the following.

Theorem 3. *Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$ and assume that (44) holds. Then there are constants $c_{1-4} > 0$ such that for all $n \in \mathbb{N}$ with $n \ge 2$, if the embedding condition (6) holds, then*

$$\begin{aligned} c_1 n^{-\gamma_1} &\le e_n^{\mathrm{det}}(\tilde{J}, \mathscr{B}_{W_p^r(Q)}, \tilde{W}_{q^*}^s(Q)^*) \\ &\le e_n^{\mathrm{det}}(J, \mathscr{B}_{W_p^r(Q)}, W_{q^*}^s(Q)^*) \ \le \ c_2 n^{-\gamma_1} (\log n)^{\nu_1}, \end{aligned}$$

where

$$\gamma_1 = \min\left(\frac{r+s}{d} - \left(\frac{1}{p} - \frac{1}{q}\right)_+, \frac{r}{d}\right), \tag{47}$$

$$\nu_1 = \begin{cases} 1 \ \textit{if} \quad \dfrac{s}{d} = \dfrac{1}{q^*},\ p = 1,\ 1 < q < \infty, \\ 0 \ \textit{otherwise}, \end{cases} \tag{48}$$

and if the embedding condition (6) does not hold, we have

$$\begin{aligned} c_3 &\le e_n^{\mathrm{det}}(\tilde{J}, \mathscr{B}_{W_p^r(Q)} \cap C(\bar{Q}), \tilde{W}_{q^*}^s(Q)^*) \\ &\le e_n^{\mathrm{det}}(J, \mathscr{B}_{W_p^r(Q)} \cap C(\bar{Q}), W_{q^*}^s(Q)^*) \le c_4. \end{aligned}$$

The case of the embedding condition is essentially due to Vybíral [22], based on results of Novak and Triebel [15], with the exception of the case $s/d = 1/p - 1/q$ with $1 \le p < q \le \infty$, which was shown in [7]. The result of Theorem 3, for the case that the embedding condition does not hold, was proved in [7].

To state the next result put $\bar{p} = \min(p, 2)$,

$$\theta = \frac{s}{d} - \left(\frac{1}{p} - \frac{1}{q}\right)_+, \quad \tau = 1 - \frac{1}{\bar{p}},$$

$$\nu_2' = \begin{cases} 0 & \text{if} \quad \theta > \tau \\ 1 & \text{if} \quad \theta = \tau \text{ and } p \le q < \infty \\ 2 - 1/\bar{p} & \text{if} \quad \theta = \tau \text{ and } p < q = \infty \\ 2 & \text{if} \quad \theta = \tau \text{ and } p = q = \infty \\ 1 & \text{if} \quad \theta = \tau \text{ and } p > q \\ 0 & \text{if} \quad \theta < \tau \text{ and } \min(p, q) < \infty \\ \theta & \text{if} \quad \theta < \tau \text{ and } p = q = \infty. \end{cases} \tag{49}$$

The main approximation result in the randomized case is

Theorem 4. *Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$ and assume that (44) holds. Then there are constants $c_1, c_2 > 0$ such that for all $n \in \mathbb{N}$ with $n \ge 2$*

$$\begin{aligned} c_1 n^{-\gamma_2} &\le e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{W_p^r(Q)}, \tilde{W}_{q^*}^s(Q)^*) \\ &\le e_n^{\mathrm{ran}}(J, \mathscr{B}_{W_p^r(Q)}, W_{q^*}^s(Q)^*) \ \le \ c_2 n^{-\gamma_2}(\log n)^{\nu_2}, \end{aligned}$$

where

$$\gamma_2 = \min\left(\frac{r+s}{d} - \left(\frac{1}{p} - \frac{1}{q}\right)_+, \frac{r}{d} + 1 - \frac{1}{\bar{p}}\right), \tag{50}$$

$$\nu_2 = \begin{cases} \nu_2' \text{ if } & \gamma_2 > 0, \\ 0 \ \text{ if } & \gamma_2 = 0, \end{cases} \tag{51}$$

and ν_2' is given by (49).

This result is proved in [7]. Together with the randomized case of Theorem 1 it solved a problem posed by Novak and Woźniakowski, see [16], Sect. 4.3.3, Problem 25. Even the case $p = q = 2$, $Q = (0,1)$ of Theorem 4 was new. The algorithm realizing the optimal rate is discussed in the next section.

For the weak elliptic problem we conclude (see also [7], Corollary 7.1 for a slightly more general statement)

Corollary 2. *Let $r \in \mathbb{N}_0$, $m \in \mathbb{N}$. Then there are constants $c_{1-6} > 0$ such that for all $n \in \mathbb{N}$ with $n \ge 2$, if the embedding condition (6) holds,*

$$c_1 n^{-\frac{r}{d}} \le e_n^{\det}(S^{\mathrm{ell}}, \mathscr{B}_{W_2^r(Q)}, \tilde{W}_2^m(Q)) \le c_2 n^{-\frac{r}{d}},$$

if the embedding condition (6) does not hold,

$$c_3 \le e_n^{\det}(S^{\mathrm{ell}}, \mathscr{B}_{W_2^r(Q)}, \tilde{W}_2^m(Q)) \le c_4,$$

and, independently of the embedding condition,

$$c_5 n^{-\frac{r}{d} - \min\left(\frac{m}{d}, \frac{1}{2}\right)} \le e_n^{\mathrm{ran}}(S^{\mathrm{ell}}, \mathscr{B}_{W_2^r(Q)}, \tilde{W}_2^m(Q)) \le c_6 n^{-\frac{r}{d} - \min\left(\frac{m}{d}, \frac{1}{2}\right)}(\log n)^{\nu_3}$$

with

$$\nu_3 = \begin{cases} 1 \text{ if } & \frac{m}{d} = \frac{1}{2}, \\ 0 \text{ otherwise}. \end{cases}$$

For the problem of integration with variable weights we obtain

Corollary 3. *Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$ and assume that (44) and the embedding condition (6) hold. Then there are constants $c_1, c_2 > 0$ such that for all $n \in \mathbb{N}$ with $n \ge 2$*

$$\begin{aligned}
& c_1 n^{-\gamma_1} \\
& \le \inf_{(x_i,\psi_i)} \sup_{f\in\mathscr{B}_{W_p^r(Q)},\, w\in\mathscr{B}_{W_{q^*}^s(Q)}} \left| \int_Q f(x)w(x)dx - \sum_{i=1}^n f(x_i)(\psi_i, w) \right| \\
& \le c_2 n^{-\gamma_1} (\log n)^{\nu_1},
\end{aligned}$$

where γ_1 and ν_1 are given by (47) and (48), and the infimum is taken over all families $(x_i)_{1\le i\le n} \subset Q$, $(\psi_i)_{1\le i\le n} \subset W_{q^*}^s(Q)^*$.

Corollary 4. *Let $r, s \in \mathbb{N}_0$, $1 \le p, q \le \infty$ and assume that (44) holds. Then there are constants $c_1, c_2 > 0$ such that for all $n \in \mathbb{N}$ with $n \ge 2$*

$$\begin{aligned}
& c_1 n^{-\gamma_2} \\
& \le \inf_{(x_{i,\omega},\psi_{i,\omega})} \sup_{f\in\mathscr{B}_{W_p^r(Q)}} \mathbb{E} \sup_{w\in\mathscr{B}_{W_{q^*}^s(Q)}} \left| \int_Q f(x)w(x)dx - \sum_{i=1}^n f(x_{i,\omega})(\psi_{i,\omega}, w) \right| \\
& \le c_2 n^{-\gamma_2} (\log n)^{\nu_2},
\end{aligned}$$

where γ_2 and ν_2 are given by (50) and (51), and the infimum is taken over all families $(x_{i,\omega})_{1\le i\le n,\,\omega\in\Omega} \subset Q$ and $(\psi_{i,\omega})_{1\le i\le n,\,\omega\in\Omega} \subset W_{q^*}^s(Q)^*$ satisfying conditions (1)–(3).

Given $1 \le p \le \infty$ and $r \in \mathbb{N}_0$, let us put $q = p$ and choose any $s \in \mathbb{N}$ satisfying

$$\frac{s}{d} > 1 - \frac{1}{p},$$

hence (44) holds, $\gamma_1 = \frac{r}{d}$, $\nu_1 = 0$, $\gamma_2 = \frac{r}{d}$ and $\nu_2 = 0$. Now setting $w(x) \equiv 1$, we recover from Corollaries 3 and 4 the upper bounds of Theorem 2. However, the resulting algorithm (see the next section) is more complicated than the one presented in Sect. 3.

5 Approximation of $J : W_p^r(Q) \to W_{q^*}^s(Q)^*$ – The Algorithm

In this section we want to explain some ideas of the construction of the algorithm from [7] which gives the upper bound in Theorem 4. If (44) holds, then, as shown in [7], proof of Proposition 4.1, we can find a number $1 \le u \le \infty$ such that both embeddings

$$J_1 : W_p^r(Q) \to L_u(Q)$$

and

$$J_{2,0} : W^s_{q^*}(Q) \to L_{u^*}(Q)$$

are continuous. Let

$$V_u : L_q(Q) \to L_{q^*}(Q)^*$$

be the embedding given by

$$(V_u f, g) = (f, g) \quad (f \in L_q(Q),\, g \in L_{q^*}(Q)), \tag{52}$$

which, in fact, is just the identity operator on $L_q(Q)$ for $1 < q \le \infty$ and the canonical embedding of $L_1(Q)$ into $L_\infty(Q)^* = L_1(Q)^{**}$ for $q = 1$. Hence with

$$J_2 = J^*_{2,0} V_u : L_u(Q) \to W^s_{q^*}(Q)^*, \tag{53}$$

the embedding J can be factorized as

$$J : W^r_p(Q) \xrightarrow{J_1} L_u(Q) \xrightarrow{J_2} W^s_{q^*}(Q)^*.$$

For the approximation of J_1 we use the algorithm from Proposition 1, see below. The key part of the approximation of J is that of J_2. We use the duality (53). Let us note the following to explain the next steps. We want to approximate $J^*_{2,0} V_q$ by operators based on function values. We know how to do this for $J_{2,0}$ (Proposition 1), but this does not help for the dual $J^*_{2,0}$, because then the delta functionals would appear at the wrong end. Moreover, we need deterministic error estimates to pass them to the dual. Thus, we start with a deterministic linear approximation of $J_{2,0}$.

Let φ_j $(j = 1, \dots, \kappa)$ be any orthonormal in $L_2([0,1]^d)$ basis of the space $\mathscr{P}_\rho$ of polynomials of degree at most ρ and let $P : L_1([0,1]^d) \to \mathscr{P}_\rho$ be defined by

$$Pg = \sum_{j=1}^{\kappa} (g, \varphi_j)\varphi_j \quad (g \in L_1([0,1]^d)).$$

For $l \ge l_0$, with l_0 the constant from Lemma 2, we define, similarly to (18), an operator $P_l : W^s_{q^*}(Q) \to L_{u^*}(Q)$ by setting for $g \in W^s_{q^*}(Q)$

$$\begin{aligned} P_l g &= \sum_{i \in \mathscr{I}_l} \chi_{\tilde{Q}_{li} \cap Q} R_{li} P E_{li} g \\ &= b^{-d} 2^{dl} \sum_{i \in \mathscr{I}_l} \sum_{k \in \mathscr{K}_{li}} \sum_{j=1}^{\kappa} (g, \chi_{Q_{li}} R_{li} \varphi_j)\, \chi_{Q_{lk} \cap Q} R_{li} \varphi_j . \end{aligned}$$

Then the dual operator

$$P_l^* f = b^{-d} 2^{dl} \sum_{i \in \mathscr{I}_l} \sum_{k \in \mathscr{K}_{li}} \sum_{j=1}^{\kappa} (f, \chi_{Q_{lk} \cap Q} R_{li} \varphi_j) \chi_{Q_{li}} R_{li} \varphi_j$$

approximates $J^*_{2,0}$. The next idea would be to use simultaneous Monte Carlo integration for the approximation of the weighted integrals $(f, \chi_{Q_{lk}\cap Q} R_{li}\varphi_j)$. This, however, does not give the optimal rate. So we resort to a multilevel splitting. We fix $L \in \mathbb{N}_0$, $L \geq l_0$, and write P_L as

$$P_L = \sum_{l=l_0}^{L}(P_l - P_{l-1}), \qquad P_{l_0-1} := 0.$$

We can represent (see [7], proof of the first part of Lemma 3.3, for details)

$$(P_l - P_{l-1})g = \sum_{k\in\mathscr{K}_l} \sum_{j=1}^{\kappa}(g, h_{lkj})\, \chi_{Q_{lk}\cap Q} R_{lk}\varphi_j, \tag{54}$$

where the h_{lkj} are defined in the following way. For $l \geq l_0$ and $k \in \mathscr{K}_l$ let $\iota(l,k)$ be the unique index $i \in \mathscr{I}_l$ with $Q_{lk} \subset \tilde{Q}_{li}$, see (13)–(17) for the definitions. Let α_{lkjm} be given by

$$\chi_{Q_{lk}} R_{l,\iota(l,k)}\varphi_j = \sum_{m=1}^{\kappa} \alpha_{lkjm}\chi_{Q_{lk}} R_{lk}\varphi_m,$$

which is a correct definition since $(R_{lk}\varphi_j)_{j=1}^{\kappa}$ is a basis of the polynomials $\mathscr{P}_\rho(Q_{lk})$ on Q_{lk}. For the case $l = l_0$ we set for $k \in \mathscr{K}_{l_0}$, $m = 1,\dots,\kappa$

$$h_{l_0km} = b^{-d}2^{dl_0}\chi_{Q_{l_0,\iota(l_0,k)}} R_{l_0,\iota(l_0,k)} \sum_{j=1}^{\kappa} \alpha_{l_0kjm}\varphi_j .$$

Furthermore, for $l \geq l_0 + 1$ and $k \in \mathscr{K}_l$ let $\upsilon(l,k)$ be the unique $i \in \mathscr{I}_{l-1}$ with $Q_{lk} \subset \tilde{Q}_{l-1,i}$. Let $\beta_{lkjm} \in \mathbb{K}$ be such that

$$\chi_{Q_{lk}} R_{l-1,\upsilon(l,k)}\varphi_j = \sum_{m=1}^{\kappa} \beta_{lkjm}\chi_{Q_{lk}} R_{lk}\varphi_m.$$

We put for $l \geq l_0 + 1$, $k \in \mathscr{K}_l$, $m = 1,\dots,\kappa$

$$\begin{aligned} h_{lkm} = \;& b^{-d}2^{dl}\chi_{Q_{l,\iota(l,k)}} R_{l,\iota(l,k)} \sum_{j=1}^{\kappa} \alpha_{lkjm}\varphi_j \\ & -b^{-d}2^{d(l-1)}\chi_{Q_{l-1,\upsilon(l,k)}} R_{l-1,\upsilon(l,k)} \sum_{j=1}^{\kappa} \beta_{lkjm}\varphi_j . \end{aligned}$$

Passing to the dual, we get from (54)

$$(P_l - P_{l-1})^* f = \sum_{k\in\mathcal{K}_l} \sum_{j=1}^{\kappa} (f, \chi_{Q_{lk}\cap Q} R_{lk}\varphi_j) h_{lkj}. \tag{55}$$

Now fix any numbers $N_l \in \mathbb{N}$ $(l = l_0, \dots, L)$ and let $(\xi_{li})_{l=l_0, i=1}^{L,N_l}$ be independent uniformly distributed on $[0,1]^d$ random variables on some complete probability space $(\Omega_2, \Sigma_2, \mathbb{P}_2)$. Put

$$\xi_{lki} = x_{lk} + b2^{-l}\xi_{li},$$

where we recall that x_{lk} is the point in Q_{lk} with minimal coordinates, so $(\xi_{lki})_{i=1}^{N_l}$ are independent, uniformly distributed on Q_{lk} random variables. We approximate the scalar products in (55) by the standard Monte Carlo method

$$\begin{aligned}&(f, \chi_{Q_{lk}\cap Q} R_{lk}\varphi_j)\\ &\approx N_l^{-1}|Q_{lk}| \sum_{i=1}^{N_l} \tilde{f}(\xi_{lki})\,(R_{lk}\varphi_j)(\xi_{lki})\\ &= N_l^{-1} b^d 2^{-dl} \sum_{i=1}^{N_l} \tilde{f}(x_{lk} + b2^{-l}\xi_{li})\varphi_j(\xi_{li}).\end{aligned}$$

Here $\tilde{f}$ is defined by

$$\tilde{f}(x) = \begin{cases} f(x) & \text{if} \quad x \in Q \\ 0 & \text{if} \quad x \in Q_l \setminus Q, \end{cases} \tag{56}$$

where

$$Q_l = \bigcup_{k\in\mathcal{K}_l} Q_{lk}.$$

This leads to the following approximations. For $f \in L_u(Q)$, $\omega_2 \in \Omega_2$,

$$\begin{aligned}&(P_l - P_{l-1})^* f \approx A_{l,\omega_2}^{(2)} f\\ &= b^d 2^{-dl} N_l^{-1} \sum_{k\in\mathcal{K}_l} \sum_{j=1}^{\kappa} \sum_{i=1}^{N_l} \tilde{f}\big(x_{lk} + b2^{-l}\xi_{li}(\omega_2)\big)\,\varphi_j\big(\xi_{li}(\omega_2)\big) h_{lkj},\end{aligned} \tag{57}$$

and, summing over the levels,

$$\begin{aligned}J_2 f &\approx P_L^* f\\ &\approx A_{\omega_2}^{(2)} f = b^d \sum_{l=l_0}^{L} 2^{-dl} N_l^{-1} \sum_{k\in\mathcal{K}_l} \sum_{j=1}^{\kappa} \sum_{i=1}^{N_l} \tilde{f}(x_{lk} + b2^{-l}\xi_{li})\,\varphi_j(\xi_{li}) h_{lkj}.\end{aligned}$$

We are ready to define the final algorithm $(A_{\omega_0})_{\omega_0 \in \Omega_0}$ for the approximation of $J : W_p^r(Q) \to W_{q^*}^s(Q)^*$. For $L_1 \in \mathbb{N}_0$, $L_1 \geq l_0$ let $\big(P^{(1)}_{L_1,\omega_1}\big)_{\omega_1 \in \Omega_1}$ be the algorithm defined in (19) with $(\Omega_1, \Sigma_1, \mathbb{P}_1)$ the associated probability space. We put

$$(\Omega_0, \Sigma_0, \mathbb{P}_0) = (\Omega_1, \Sigma_1, \mathbb{P}_1) \times (\Omega_2, \Sigma_2, \mathbb{P}_2)$$

and use $P^{(1)}_{L_1,\omega_1}$ for the approximation of J_1 – which is a way of variance reduction similar to that in the integration algorithm (37) in Sect. 3. Then $A^{(2)}_{\omega_2}$ is applied to the difference $f - P^{(1)}_{L_1,\omega_1} f$. Hence we set for $\omega_0 = (\omega_1, \omega_2)$ and $f \in W_p^r(Q)$

$$A_{\omega_0} f = P^{(1)}_{L_1,\omega_1} f + A^{(2)}_{\omega_2}(f - P^{(1)}_{L_1,\omega_1} f).$$

We refer to [7] for the appropriate choice of the parameters and the error analysis giving the upper estimate of Theorem 4.

6 Indefinite Integration and Approximation in Spaces of Functions with Dominating Mixed Derivatives

This chapter is based on [8], where indefinite integration was studied. Here, however, we mainly explore the connection to approximation in certain Sobolev spaces of functions with dominating mixed derivatives, which has not been considered in [8].

In this section we put

$$Q = [0, 1]^d.$$

Let $1 \leq p \leq \infty$, $\bar{1} = (1, 1, \dots, 1) \in \mathbb{N}^d$, and define

$$\hat{W}_p^{\bar{1}}(Q) = \Big\{ f \in \mathscr{F}(Q) : \ \exists g \in L_p(Q), \ f(x) = \int_{[x,\bar{1}]} g(t)dt \ \ (x \in Q) \Big\},$$

where for $x = (x_1, \dots, , x_d)$ we put $[x, \bar{1}] = [x_1, 1] \times \dots \times [x_d, 1]$. The space $\hat{W}_p^{\bar{1}}(Q)$ is equipped with the norm

$$\|f\|_{\hat{W}_p^{\bar{1}}(Q)} = \|D^{\bar{1}} f\|_{L_p(Q)} = \|g\|_{L_p(Q)}.$$

So $\hat{W}_p^{\bar{1}}(Q)$ is a space of functions with dominating mixed smoothness and boundary conditions (these functions vanish for all $x \in Q$ with at least one coordinate equal to 1). Let $\tilde{W}_p^{\bar{1}}(Q)$ be the closure in $\hat{W}_p^{\bar{1}}(Q)$ of the set of infinitely differentiable functions with support in the interior of Q. Let

$$U_p : \tilde{W}_p^{\bar{1}}(Q) \to \hat{W}_p^{\bar{1}}(Q)$$

be the identical embedding. We define for $1 < p \le \infty$

$$W_p^{-\bar{1}}(Q) = \tilde{W}_{p^*}^{\bar{1}}(Q)^*.$$

Similarly to Sect. 4, our goal is to study stochastic approximation of

$$J : L_p(Q) \to \hat{W}_{q^*}^{\bar{1}}(Q)^*$$

and

$$\tilde{J} = U_{q^*}^* J : L_p(Q) \to \tilde{W}_{q^*}^{\bar{1}}(Q)^* \tag{58}$$

for $1 \le p, q \le \infty$, where J is defined by

$$(Jf)(g) = \int_Q f(x)g(x)dx \quad \big(f \in L_p(Q),\, g \in \hat{W}_{q^*}^{\bar{1}}(Q)\big). \tag{59}$$

It is easily verified that J and $\tilde{J}$ are continuous injections. We shall relate the embedding J to indefinite integration, investigated in [8]. For this purpose we introduce the operator $S : L_p(Q) \to L_q(Q)$ of indefinite integration by setting for $f \in L_p(Q)$ and $x = (x_1, \dots, x_d) \in Q$

$$(Sf)(x) = \int\limits_{[0,x]} f(t)dt = \int_0^{x_1} \dots \int_0^{x_d} f(t_1, \dots, t_d)dt.$$

Clearly, S is continuous for all $1 \le p, q \le \infty$. To establish the connection to J we also introduce a related operator $S_0 : L_p(Q) \to L_q(Q)$ by

$$(S_0 f)(x) = \int_{[x,\bar{1}]} f(t)dt.$$

For $f, g \in L_1(Q)$ we have

$$(Sf, g) = (f, S_0 g). \tag{60}$$

Furthermore, the operator S_0 is an isometric isomorphism from $L_{q^*}(Q)$ to $\hat{W}_{q^*}^{\bar{1}}(Q)$ (meaning that S_0 maps $L_{q^*}(Q)$ onto $\hat{W}_{q^*}^{\bar{1}}(Q)$ with preservation of the norm). Hence, the dual operator

$$S_0^* : \hat{W}_{q^*}^{\bar{1}}(Q)^* \to L_{q^*}(Q)^*$$

and its inverse

$$(S_0^*)^{-1} : L_{q^*}(Q)^* \to \hat{W}_{q^*}^{\bar{1}}(Q)^*$$

are isometric isomorphisms, as well. Next we show that J can be represented as

$$J : L_p(Q) \xrightarrow{S} L_q(Q) \xrightarrow{V_q} L_{q^*}(Q)^* \xrightarrow{(S_0^*)^{-1}} \hat{W}^{\bar{1}}_{q^*}(Q)^*, \tag{61}$$

where V_q is the canonical embedding, see (52). Indeed, let $f \in L_p(Q)$, $g \in \hat{W}^{\bar{1}}_{q^*}(Q)$. Then, using (60) and (52),

$$((S_0^*)^{-1} V_q S f, g) = (V_q S f, S_0^{-1} g) = (Sf, S_0^{-1} g) = (f, g),$$

from which (61) follows. Since $(S_0^*)^{-1}$ is an isometric isomorphism and, for $1 < q \le \infty$, V_q is the identity of $L_q(Q)$, we conclude in this case

$$e_n^{\mathrm{ran}}(J, \mathscr{B}_{L_p(Q)}, \hat{W}^{\bar{1}}_{q^*}(Q)^*) = e_n^{\mathrm{ran}}(S, \mathscr{B}_{L_p(Q)}, L_q(Q)). \tag{62}$$

This relation also holds for $q = 1$, because then V_1 is an isometric embedding of $L_1(Q)$ into $L_1(Q)^{**}$ such that the range $V_1(L_1(Q))$ admits a projection of norm 1 from $L_1(Q)^{**}$, see, e.g., [11], Par. 17, Theorem 3 (in combination with Par. 3, Theorem 7 and Par. 15, Theorem 3). Taking into account (58) and $\|U_q\| = 1$, it follows from (62) that

$$e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{L_p(Q)}, \tilde{W}^{\bar{1}}_{q^*}(Q)^*) \le e_n^{\mathrm{ran}}(J, \mathscr{B}_{L_p(Q)}, \hat{W}^{\bar{1}}_{q^*}(Q)^*). \tag{63}$$

The respective counterparts of (62) and (63) for the deterministic minimal error e_n^{det} also hold. We are ready to apply the following result from [8].

Theorem 5. *Let $d \in \mathbb{N}$, $1 \le p \le \infty$ and $\bar{p} = \min(p, 2)$. Then there are constants $c_1, c_2 > 0$ such that*

$$c_1 n^{-1+1/\bar{p}} \le e_n^{\mathrm{ran}}(S, \mathscr{B}_{L_p(Q)}, L_\infty(Q)) \le c_2 n^{-1+1/\bar{p}}.$$

Using this theorem, we can derive the respective results for the embedding operators J and $\tilde{J}$ as well as an easy generalization of Theorem 5 itself.

Corollary 5. *Let $d \in \mathbb{N}$, $1 \le p, q \le \infty$ and $\bar{p} = \min(p, 2)$. Then there are constants $c_{1-4} > 0$ such that*

$$c_1 n^{-1+1/\bar{p}} \le e_n^{\mathrm{ran}}(S, \mathscr{B}_{L_p(Q)}, L_q(Q)) \le c_2 n^{-1+1/\bar{p}} \tag{64}$$

$$\begin{aligned} c_3 n^{-1+1/\bar{p}} &\le e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{L_p(Q)}, \tilde{W}^{\bar{1}}_{q^*}(Q)^*) \\ &\le e_n^{\mathrm{ran}}(J, \mathscr{B}_{L_p(Q)}, \hat{W}^{\bar{1}}_{q^*}(Q)^*) \le c_4 n^{-1+1/\bar{p}}. \end{aligned} \tag{65}$$

Proof. The upper bound in (64) follows from Theorem 5 and the continuity of the embedding $L_\infty(Q) \to L_q(Q)$. The upper bound of (65) is a consequence of (62), (63), and the upper bound of (64).

The lower bound of (65) is shown by a reduction to integration. Let ψ be a C^∞-function with support in Q satisfying $\psi \geq 0$ and $\psi \not\equiv 0$. Define $S_1 : L_p(Q) \to \mathbb{K}$ as

$$S_1 f = \int_Q f(x)\psi(x)dx \quad (f \in L_p(Q)).$$

By (59),

$$(\tilde{J} f, \psi) = S_1 f,$$

thus

$$e_n^{\mathrm{ran}}(S_1, \mathscr{B}_{L_p(Q)}, \mathbb{K}) \leq \|\psi\|_{\tilde{W}_{q^*}^{\bar{1}}(Q)}\, e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{L_p(Q)}, \tilde{W}_{q^*}^{\bar{1}}(Q)^*),$$

while it is well-known that

$$e_n^{\mathrm{ran}}(S_1, \mathscr{B}_{L_p(Q)}, \mathbb{K}) \geq cn^{-1+1/\bar{p}},$$

see [14]. Finally, the lower bound of (64) follows from (62), (63), and the lower bound of (65). □

Let us mention that in the deterministic case there is no convergence to zero of the minimal error. This is easily shown by reduction to integration, in the same way as in the proof of Corollary 5. Thus, we have

Corollary 6. *Let $d \in \mathbb{N}$, $1 \leq p, q \leq \infty$. Then there are constants $c_{1-4} > 0$ such that*

$$c_1 \leq e_n^{\det}(S, \mathscr{B}_{L_p(Q)}, L_q(Q)) \leq c_2$$
$$c_3 \leq e_n^{\det}(\tilde{J}, \mathscr{B}_{L_p(Q)}, \tilde{W}_{q^*}^{\bar{1}}(Q)^*) \leq e_n^{\det}(J, \mathscr{B}_{L_p(Q)}, \hat{W}_{q^*}^{\bar{1}}(Q)^*) \leq c_4.$$

So far the constants in the estimates could depend in an arbitrary way on the dimension. Now we take a closer look at the upper bounds with the goal of establishing polynomial dependence of the constants on the dimension, hence tractability, see [16, 17] for this notion and the theory thereof. We restrict our considerations to the case $q = \infty$, since in this case the problems S and J are normalized, meaning that

$$\begin{aligned}\|S : L_p(Q) \to L_\infty(Q)\| &= \|J : L_p(Q) \to \hat{W}_1^{\bar{1}}(Q)^*\| \\ &= \|\tilde{J} : L_p(Q) \to W_\infty^{-\bar{1}}(Q)\| = 1,\end{aligned}$$

so tractability with respect to the absolute and relative error criterion (see [16, 17]) coincide.

Most tractability results were established for weighted problems, that is, with a decreasing dependence on subsequent dimensions. Here we show tractability for certain unweighted embedding operators. We again use the connection to indefinite integration (62) and a respective result from [8]. For this sake we introduce the simple sampling algorithm. Let $(\xi_i)_{i=1}^n$ be independent, uniformly distributed on Q random variables on a complete probability space $(\Omega, \Sigma, \mathbb{P})$. We approximate the indefinite integration operator S by

$$(Sf)(x) = \int_Q \chi_{[0,x]}(t) f(t) dt$$
$$\approx (A_{n,\omega} f)(x) = \frac{1}{n} \sum_{i=1}^n \chi_{[0,x]}(\xi_i(\omega))\, f(\xi_i(\omega)) \quad (x \in Q, \omega \in \Omega),$$

thus

$$Sf \approx A_{n,\omega} f = \frac{1}{n} \sum_{i=1}^n f(\xi_i) \chi_{[\xi_i, \bar{1}]}.$$

We note that this algorithm satisfies consistency (1), but does not possess the measurability properties (2) and (3). However, for each $f \in L_p(Q)$ the mapping

$$\omega \in \Omega \to \|Sf - A_{n,\omega} f\|_{L_\infty(Q)}$$

is Σ-measurable, see [8] for these facts and also for another algorithm with the same approximation properties, but fulfilling (1)–(3).

The following was shown in [8]. A proof of a generalization of (66) is given in Sect. 7.

Theorem 6. *Let* $1 \le p \le \infty$, $1 \le p_1 < \infty$, $p_1 \le p$, *and* $\bar{p} = \min(p, 2)$. *Then there is a constant* $c > 0$ *such that for all* $d, n \in \mathbb{N}$, $Q = [0, 1]^d$, $f \in L_p(Q)$,

$$\left(\mathbb{E}\, \|Sf - A_{n,\omega} f\|_{L_\infty(Q)}^{p_1}\right)^{1/p_1} \le c d^{1-1/\bar{p}} n^{-1+1/\bar{p}} \|f\|_{L_p(Q)}, \tag{66}$$

and moreover,

$$e_n^{\text{ran}}(S, \mathscr{B}_{L_p(Q)}, L_\infty(Q)) \le c d^{1-1/\bar{p}} n^{-1+1/\bar{p}}. \tag{67}$$

Let us define a related algorithm on $L_p(Q)$ with values in $\hat{W}_1^{\bar{1}}(Q)^*$ by setting for $f \in L_p(Q)$ and $\omega \in \Omega$

$$A_{n,\omega}^{(1)} f = \sum_{i=1}^n f(\xi_i(\omega)) \delta_{\xi_i(\omega)}$$

with the ξ_i as above and $\delta_x \in \hat{W}_1^{\bar{1}}(Q)^*$ given for $x \in Q$ by

$$(g, \delta_x) = g(x) \quad (g \in \hat{W}_1^{\bar{1}}(Q)).$$

A corresponding algorithm $\tilde{A}_{n,\omega}^{(1)}$ with values in $\tilde{W}_1^{\bar{1}}(Q)^* = W_\infty^{-\bar{1}}(Q)$ is defined by

$$\tilde{A}^{(1)}_{n,\omega} f = U_1^* A^{(1)}_{n,\omega} f = \sum_{i=1}^n f(\xi_i(\omega))\tilde{\delta}_{\xi_i(\omega)}, \tag{68}$$

with $\tilde{\delta}_x$ standing for δ_x, interpreted as a functional on the subspace $\tilde{W}_1^{\bar{1}}(Q)$. We use Theorem 6 to derive the following error estimates for the algorithms $A_n^{(1)}$ and $\tilde{A}_n^{(1)}$.

Proposition 3. *Let* $1 \le p \le \infty$, $1 \le p_1 < \infty$, $p_1 \le p$, *and* $\bar{p} = \min(p, 2)$. *Then there is a constant* $c > 0$ *such that for all* $d, n \in \mathbb{N}$, $Q = [0,1]^d$, $f \in L_p(Q)$,

$$\mathbb{E}\left(\|\tilde{J} f - \tilde{A}^{(1)}_{n,\omega} f\|^{p_1}_{W_\infty^{-\bar{1}}(Q)}\right)^{1/p_1} \le \mathbb{E}\left(\|J f - A^{(1)}_{n,\omega} f\|^{p_1}_{\hat{W}_1^{\bar{1}}(Q)^*}\right)^{1/p_1} \tag{69}$$

$$\le c d^{1-1/\bar{p}} n^{-1+1/\bar{p}} \|f\|_{L_p(Q)}, \tag{70}$$

and moreover,

$$\begin{aligned} e_n^{\mathrm{ran}}(\tilde{J}, \mathscr{B}_{L_p(Q)}, W_\infty^{-\bar{1}}(Q)) &\le e_n^{\mathrm{ran}}(J, \mathscr{B}_{L_p(Q)}, \hat{W}_1^{\bar{1}}(Q)^*) \\ &\le c d^{1-1/\bar{p}} n^{-1+1/\bar{p}}. \end{aligned} \tag{71}$$

Proof. Inequality (69) follows from (58) and (68). To show (70), we first note that for $g \in \hat{W}_{q^*}^{\bar{1}}(Q)$ and $x \in Q$ we have

$$\begin{aligned} (g, (S_0^*)^{-1}\chi_{[x,\bar{1}]}) &= (S_0^{-1} g, \chi_{[x,\bar{1}]}) = (S_0(S_0^{-1} g))(x) \\ &= g(x) = (g, \delta_x), \end{aligned}$$

thus

$$(S_0^*)^{-1}\chi_{[x,\bar{1}]} = \delta_x. \tag{72}$$

Consequently, using (61) (noting that V_∞ is the identity of $L_\infty(Q)$), (72), and (66) of Theorem 6, we get for $f \in L_p(Q)$

$$\begin{aligned} &\mathbb{E}\left(\|J f - A^{(1)}_{n,\omega} f\|^{p_1}_{\hat{W}_1^{\bar{1}}(Q)^*}\right)^{1/p_1} \\ &= \mathbb{E}\left(\left\|J f - \sum_{i=1}^n f(\xi_i)\delta_{\xi_i}\right\|^{p_1}_{\hat{W}_1^{\bar{1}}(Q)^*}\right)^{1/p_1} \\ &= \mathbb{E}\left(\left\|(S_0^*)^{-1} S f - \sum_{i=1}^n f(\xi_i)\,(S_0^*)^{-1}\chi_{[\xi_i,\bar{1}]}\right\|^{p_1}_{\hat{W}_1^{\bar{1}}(Q)^*}\right)^{1/p_1} \\ &= \mathbb{E}\left(\left\|S f - \sum_{i=1}^n f(\xi_i)\chi_{[\xi_i,\bar{1}]}\right\|^{p_1}_{L_\infty(Q)}\right)^{1/p_1} \end{aligned}$$

$$= \mathbb{E}\left(\left\| Sf - A_{n,\omega} f \right\|_{L_\infty(Q)}^{p_1}\right)^{1/p_1} \le c d^{1-1/\bar{p}} n^{-1+1/\bar{p}}.$$

Finally, (71) follows from (67), (62), and (63). □

The results in this section are very specific, leaving much room for further investigations, e.g., of smoothness different from $\bar{1}$, of other source spaces than $L_p(Q)$, and of more general domains Q. In the latter direction a generalization of the first part of Theorem 6 is given in the next section.

7 A Generalization of Indefinite Integration and Tractability of Discrepancy

Let $(G, \mathscr{G})$ be a measurable space, that is, G is a nonempty set and $\mathscr{G}$ a σ-algebra of subsets of G. Let $\mathscr{C} \subseteq \mathscr{G}$ be a family of measurable subsets of G. Recall that the Vapnik-Červonenkis dimension $v(\mathscr{C})$ is defined to be the smallest $m \in \mathbb{N}_0$ such that for each set $B \subseteq G$ with $m + 1$ elements the following holds

$$|\{B \cap C : C \in \mathscr{C}\}| < 2^{m+1},$$

if there is such an m, and $v(\mathscr{C}) = \infty$, if there is none. If $v(\mathscr{C}) < \infty$, the family $\mathscr{C}$ is called a Vapnik-Červonenkis class. Let μ be a probability measure on $(G, \mathscr{G})$ and define the following generalization of the indefinite integration operator

$$S_{\mathscr{C}} : L_p(G, \mu) \to \ell_\infty(\mathscr{C})$$

by setting for $f \in L_p(G, \mu)$ and $C \in \mathscr{C}$

$$(S_{\mathscr{C}} f)(C) = \int_C f(t) d\mu(t).$$

Note that here we have again weighted integration. This time the weight is fixed, but we seek to approximate simultaneously over a family of integration domains.

We shall study randomized approximation of $S_{\mathscr{C}}$ for Vapnik-Červonenkis classes $\mathscr{C}$. For this purpose we define the analogue of the simple sampling algorithm. Let $(\xi_i)_{i=1}^n$ be independent random variables on some probability space $(\Omega, \Sigma, \mathbb{P})$ with values in G and distribution μ. For $f \in L_1(G, \mu)$, $C \in \mathscr{C}$, and $\omega \in \Omega$ put

$$(A_{n,\omega} f)(C) = \frac{1}{n} \sum_{i=1}^n \chi_C(\xi_i(\omega)) f(\xi_i(\omega)).$$

This algorithm satisfies consistency (1), but may fail the measurability properties (2) and (3), even for countable $\mathscr{C}$. We refer to [8], Sect. 6.3 for an argument which is easily seen to cover also the present situation. On the other hand, it is easy to verify that for countable $\mathscr{C}$ we have again the following weaker measurability property. For each $f \in L_p(Q)$

$$\|S_{\mathscr{C}} f - A_{n,\omega} f\|_{\ell_\infty(\mathscr{C})}$$

is Σ-measurable.

The next result generalizes Theorem 6. We adopt the proof of [8], Lemma 3.3 to this general setting. How to pass to the uncountable class involved in Theorem 6 is discussed below.

Theorem 7. *Let $1 \le p \le \infty$, $1 \le p_1 < \infty$, $p_1 \le p$, and $\bar{p} = \min(p, 2)$. Then there is a constant $c > 0$ such that the following holds. For any $(G, \mathscr{G}, \mu)$ and $(\xi_i)_{i=1}^n$ as above, any countable family $\mathscr{C} \subseteq \mathscr{G}$ and any $f \in L_p(G, \mu)$*

$$\left(\mathbb{E}\,\|S_{\mathscr{C}} f - A_{n,\omega} f\|_{\ell_\infty(\mathscr{C})}^{p_1}\right)^{1/p_1} \le c v(\mathscr{C})^{1-1/\bar{p}} n^{-1+1/\bar{p}} \|f\|_{L_p(G,\mu)}. \tag{73}$$

Proof. We fix $f \in L_p(G, \mu)$. Let $\mathscr{C}_0 \subseteq \mathscr{C}$ be any finite nonempty subset and let $\mathscr{G}_0$ be the algebra of subsets of G generated by $\mathscr{C}_0$. Let $\mathscr{M}(G, \mathscr{G}_0)$ denote the Banach space of signed measures on $\mathscr{G}_0$, equipped with the total variation norm. Introduce an operator $J_{\mathscr{C}} : \mathscr{M}(G, \mathscr{G}_0) \to \ell_\infty(\mathscr{C}_0)$ defined by

$$J_{\mathscr{C}_0} \mu = (\mu(C))_{C \in \mathscr{C}_0}.$$

According to a result of Pisier [18], Theorem 1 and Remark 6, there is a constant $c > 0$ depending only on $\bar{p}$ such that the type $\bar{p}$ constant of $J_{\mathscr{C}_0}$, recall the definition (4), satisfies

$$\tau_{\bar{p}}(J_{\mathscr{C}_0}) \le c v(\mathscr{C}_0)^{1-1/\bar{p}} \le c v(\mathscr{C})^{1-1/\bar{p}}. \tag{74}$$

Define independent, zero mean, $\mathscr{M}(G, \mathscr{G}_0)$-valued random variables $(\eta_i)_{i=1}^n$ as follows. For $B \in \mathscr{G}_0$ we set

$$\eta_i(B) = \int_B f(t) d\mu(t) - \chi_B(\xi_i) f(\xi_i).$$

We have

$$\begin{aligned}\left(\mathbb{E}\,\|\eta_i\|_{\mathscr{M}(G,\mathscr{G}_0)}^{p_1}\right)^{1/p_1} &\le \left(\mathbb{E}\left(\int_G |f(t)| d\mu(t) + |f(\xi_i)|\right)^{p_1}\right)^{1/p_1} \\ &\le 2\|f\|_{L_{p_1}(G,\mu)}.\end{aligned} \tag{75}$$

Next we apply Lemma 1. We assume that $p_1 \ge \bar{p}$, the other case then follows from Hölder's inequality. Using (74) and (75), we get

$$\left(\mathbb{E}\max_{C\in\mathscr{C}_0}\left|\int_C f(t)d\mu(t)-\frac{1}{n}\sum_{i=1}^n\chi_C(\xi_i)f(\xi_i)\right|^{p_1}\right)^{1/p_1}$$
$$=n^{-1}\left(\mathbb{E}\left\|\sum_{i=1}^n J_{\mathscr{C}_0}\eta_i\right\|^{p_1}_{\ell_\infty(\mathscr{C}_0)}\right)^{1/p_1}$$
$$\leq c\tau_{\bar{p}}(J_{\mathscr{C}_0})n^{-1}\left(\sum_{i=1}^n\left(\mathbb{E}\,\|\eta_i\|^{p_1}_{\mathscr{M}(G,\mathscr{G}_0)}\right)^{\bar{p}/p_1}\right)^{1/\bar{p}}$$
$$\leq cv(\mathscr{C})^{1-1/\bar{p}}n^{-1+1/\bar{p}}\|f\|_{L_p(G,\mu)},$$

from which (73) follows by Fatou's lemma. □

For $G=[0,1]^d$, $\mathscr{G}$ the σ-algebra of Lebesgue measurable sets, μ the Lebesgue measure, and

$$\mathscr{C}=\mathscr{C}^{(0)}=\{[0,x]:x\in[0,1]^d\cap\mathbb{Q}^d\},$$

where $\mathbb{Q}$ denotes the set of rationals, we have $v(\mathscr{C}^{(0)})=d$, see, e.g., [3], Corollary 9.2.15. Moreover, for $f\in L_1([0,1]^d)$ and $t_1,\dots,t_n\in[0,1]^d$

$$\sup_{x\in[0,1]^d\cap\mathbb{Q}^d}\left|\int_{[0,x]}f(t)dt-\frac{1}{n}\sum_{i=1}^n\chi_{[0,x]}(t_i)f(t_i)\right|$$
$$=\sup_{x\in[0,1]^d}\left|\int_{[0,x]}f(t)dt-\frac{1}{n}\sum_{i=1}^n\chi_{[0,x]}(t_i)f(t_i)\right|. \tag{76}$$

This is an immediate consequence of 'right'-continuity

$$\lim_{y\to x,\,y\geq x}\chi_{[0,y]}(t)=\chi_{[0,x]}(t)\quad(t\in[0,1]^d). \tag{77}$$

Now Theorem 6 follows from Theorem 7.

Given a point set $\{t_1,\dots,t_n\}\subset[0,1]^d$, the star discrepancy is defined as

$$d^*_\infty(t_1,\dots,t_n)=\sup_{x\in[0,1]^d}\left||[0,x)|-\frac{1}{n}\sum_{i=1}^n\chi_{[0,x)}(t_i)\right|.$$

The main result of [9] established tractability of the star-discrepancy, meaning an estimate which has a negative power in n and a constant which depends polynomially on d:

Theorem 8. *There is a constant $c>0$ such that for all $d,n\in\mathbb{N}$ there exist $t_1,\dots,t_n\in[0,1]^d$ with*

$$d^*_\infty(t_1,\dots,t_n)\leq cd^{1/2}n^{-1/2}.$$

It turns out that we can recover this result – even in a much more general form – as a direct consequence of Theorem 7. For this purpose, let us introduce the following generalization of the star-discrepancy. Let $(G, \mathscr{G}, \mu)$ be a probability space, $\mathscr{C} \subset \mathscr{G}$ any subfamily, let $f \in L_1(G, \mu)$ be a function (not an equivalence class) and set for $\{t_1, \ldots, t_n\} \subset G$

$$d_\infty^{\mathscr{C},\mu,f}(t_1, \ldots, t_n) = \sup_{C \in \mathscr{C}} \left| \int_C f(t)d\mu(t) - \frac{1}{n} \sum_{i=1}^n f(t_i)\chi_C(t_i) \right|.$$

So this discrepancy measures how well the quasi-Monte Carlo method defined by the point set $\{t_1, \ldots, t_n\}$ approximates the integral of a function f with respect to a distribution μ, uniformly over all sets C of a given family $\mathscr{C}$. From Theorem 7 we obtain

Corollary 7. *Let $1 < p \le \infty$ and $\bar{p} = \min(p, 2)$. Then there is a constant $c > 0$ such that for for any probability space $(G, \mathscr{G}, \mu)$, countable $\mathscr{C} \subseteq \mathscr{G}$, and any function $f \in L_p(G, \mu)$ there exist $t_1, \ldots, t_n \in G$ with*

$$d_\infty^{\mathscr{C},\mu,f}(t_1, \ldots, t_n) \le c v(\mathscr{C})^{1-1/\bar{p}} n^{-1+1/\bar{p}} \|f\|_{L_p(G,\mu)}.$$

If we choose $f \equiv 1$ and write $d_\infty^{\mathscr{C},\mu}$ instead of $d_\infty^{\mathscr{C},\mu,1}$, we have

$$d_\infty^{\mathscr{C},\mu}(t_1, \ldots, t_n) = \sup_{C \in \mathscr{C}} \left| \mu(C) - \frac{1}{n} \sum_{i=1}^n \chi_C(t_i) \right|.$$

Corollary 7 with $p = \infty$ implies

Corollary 8. *There is a constant $c > 0$ such that for any probability space $(G, \mathscr{G}, \mu)$ and countable $\mathscr{C} \subseteq \mathscr{G}$ there exist $t_1, \ldots, t_n \in G$ with*

$$d_\infty^{\mathscr{C},\mu}(t_1, \ldots, t_n) \le c v(\mathscr{C})^{1/2} n^{-1/2}.$$

Note that this result was also obtained in [9], Theorem 4, by slightly different tools. Theorem 8 follows from Corollary 8 by taking $G = [0, 1]^d$, μ the Lebesgue measure, and

$$\mathscr{C} = \mathscr{C}^{(1)} = \{[0, x) : x \in [0, 1]^d \cap \mathbb{Q}^d\}.$$

Then we have again $v(\mathscr{C}^{(1)}) = d$ and the analogue of (76) holds, which follows from ‘left’-continuity in place of (77).

In this section we only considered upper bounds. For results on d-dependent lower bounds we refer to [8–10].

References

1. R. A. Adams, Sobolev Spaces, Academic Press, New York, 1975.
2. P. G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland, Amsterdam, 1978.
3. R. M. Dudley, A course on empirical processes (École d'Été de Probabilités de Saint-Flour XII-1982). Lecture Notes in Mathematics 1097, 2–141, Springer-Verlag, New York, 1984.
4. S. Heinrich, Random approximation in numerical analysis, in: K. D. Bierstedt, A. Pietsch, W. M. Ruess, D. Vogt (Eds.), Functional Analysis, Marcel Dekker, New York, 1993, 123–171.
5. S. Heinrich, Randomized approximation of Sobolev embeddings, in: Monte Carlo and Quasi-Monte Carlo Methods 2006 (A. Keller, S. Heinrich, H. Niederreiter, eds.), Springer, Berlin, 2008, 445–459.
6. S. Heinrich, Randomized approximation of Sobolev embeddings II, J. Complexity 25 (2009), 455–472.
7. S. Heinrich, Randomized approximation of Sobolev embeddings III, J. Complexity 25 (2009), 473–507.
8. S. Heinrich, B. Milla, The randomized complexity of indefinite integration, J. Complexity 27 (2011), 352–382.
9. S. Heinrich, E. Novak, G. W. Wasilkowski, H. Woźniakowski, The inverse of the star-discrepancy depends linearly on the dimension, Acta Arithmetica 96 (2001), 279–302.
10. A. Hinrichs, Covering numbers, Vapnik-Červonenkis classes and bounds for the star-discrepancy, J. Complexity 20 (2004), 477–483.
11. H. E. Lacey, The Isometric Theory of Classical Banach Spaces, Springer, Berlin-Heidelberg -New York, 1974.
12. M. Ledoux, M. Talagrand, Probability in Banach Spaces, Springer, Berlin-Heidelberg -New York, 1991.
13. P. Mathé, Random approximation of Sobolev embeddings, J. Complexity 7 (1991), 261–281.
14. E. Novak, Deterministic and Stochastic Error Bounds in Numerical Analysis, Lecture Notes in Mathematics **1349**, Springer-Verlag, Berlin, 1988.
15. E. Novak, H. Triebel, Function spaces in Lipschitz domains and optimal rates of convergence for sampling, Constr. Approx. **23** (2006), 325–350.
16. E. Novak, H. Woźniakowski, Tractability of Multivariate Problems, Volume 1, Linear Information, European Math. Soc., Zürich, 2008.
17. E. Novak, H. Woźniakowski, Tractability of Multivariate Problems, Volume 2, Standard Information for Functionals, European Math. Soc., Zürich, 2010.
18. G. Pisier, Remarques sur les classes de Vapnik-Červonenkis, Ann. Inst. Henri Poincaré, Probab. Stat. 20 (1984), 287–298.
19. E. M. Stein, Singular Integrals and Differentiability Properties of Functions, Princeton University Press, Princeton, 1970.
20. J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, Information-Based Complexity, Academic Press, 1988.
21. H. Triebel, Bases in Function Spaces, Sampling, Discrepancy, Numerical Integration, European Math. Soc., Zürich, 2010.
22. J. Vybíral, Sampling numbers and function spaces, J. Complexity 23 (2007), 773–792.
23. G. W. Wasilkowski, Randomization for continuous problems, J. Complexity 5 (1989), 195–218.

On Figures of Merit for Randomly-Shifted Lattice Rules

Pierre L'Ecuyer and David Munger

Abstract Randomized quasi-Monte Carlo (RQMC) can be seen as a variance reduction method that provides an unbiased estimator of the integral of a function f over the s-dimensional unit hypercube, with smaller variance than standard Monte Carlo (MC) under certain conditions on f and on the RQMC point set. When f is smooth enough, the variance converges faster, asymptotically, as a function of the number of points n, than the MC rate of $\mathscr{O}(1/n)$. The RQMC point sets are typically constructed to minimize a given parameterized measure of discrepancy between their empirical distribution and the uniform distribution. These parameters can give different weights to the different subsets of coordinates (or lower-dimensional projections) of the points, for example. The ideal parameter values (to minimize the variance) depend very much on the integrand f and their choice (or estimation) is far from obvious in practice. In this paper, we survey this question for randomly-shifted lattice rules, an important class of RQMC point sets, and we explore the practical issues that arise when we want to use the theory to construct lattices for applications. We discuss various ways of selecting figures of merit and for estimating their ideal parameters (including the weights), we examine how they can be implemented in practice, and we compare their performance on examples inspired from real-life problems. In particular, we look at how much improvement (variance reduction) can be obtained, on some examples, by constructing the points based on function-specific figures of merit compared with more traditional general-purpose lattice-rule constructions.

P. L'Ecuyer · D. Munger
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; mungerd@gmail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_6, © Springer-Verlag Berlin Heidelberg 2012

1 Introduction: Monte Carlo and Randomized Quasi-Monte Carlo

We are concerned with the problem of estimating the integral of a function $f : [0,1)^s \to \mathbb{R}$ over the s-dimensional unit hypercube $[0,1)^s = \{\mathbf{u} = (u_1,\ldots,u_s) : 0 \le u_j < 1 \text{ for all } j\}$, by evaluating f at n points in this hypercube and taking the average. The integral can be written as

$$\mu = \mu(f) = \int_{[0,1)^s} f(\mathbf{u})\,d\mathbf{u} = \mathbb{E}[f(\mathbf{U})] \tag{1}$$

where $\mathbb{E}$ denotes the mathematical expectation and $\mathbf{U} = (U_1,\ldots,U_s) \sim U(0,1)^s$ (a random vector uniformly distributed over $(0,1)^s$). A large class of applications fit this framework [14, 18, 19].

The standard *Monte Carlo* (MC) method generates n *independent* realizations of $\mathbf{U}$, say $\mathbf{U}_0,\ldots,\mathbf{U}_{n-1}$, and estimates μ by

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=0}^{n-1} f(\mathbf{U}_i).$$

This estimator is unbiased and its variance converges as $\mathcal{O}(n^{-1})$ when $n \to \infty$.

Randomized quasi-Monte Carlo (RQMC) employs an estimator of the same form,

$$\hat{\mu}_{n,\text{rqmc}} = \frac{1}{n}\sum_{i=0}^{n-1} f(\mathbf{U}_i) \tag{2}$$

where $\mathbf{U}_i \sim U[0,1)^s$ for each i, so $\mathbb{E}[\hat{\mu}_{n,\text{rqmc}}] = \mu$ (the estimator is unbiased), but the $\mathbf{U}_i$'s are no longer independent. The aim is to have $\text{Var}[\hat{\mu}_{n,\text{rqmc}}] < \text{Var}[\hat{\mu}_n]$. For this, the randomized points are constructed so that $P_n = \{\mathbf{U}_0,\ldots,\mathbf{U}_{n-1}\} \subset [0,1)^s$ covers $[0,1)^s$ more evenly than typical independent random points, in the sense that some selected (expected) measure of discrepancy between the empirical distribution of P_n and the uniform distribution over $[0,1)^s$ is smaller. Two popular classes of RQMC point sets are randomly-shifted lattices and digitally-shifted nets. For background on RQMC methods, including lattice rules, see [14, 16, 18–21] and the references given there.

In this paper, we focus on randomly-shifted lattice rules, where P_n is the intersection of a randomly-shifted lattice with $[0,1)^s$ [21]. For any given square-integrable f (that is, for which $\text{Var}[f(\mathbf{U}_i)] < \infty$), $\text{Var}[\hat{\mu}_{n,\text{rqmc}}]$ can be written explicitly in terms of the square Fourier coefficients of f and on the lattice. Conceptually, one could compute the optimal lattice for f by solving an optimization problem that minimizes this variance expression with respect to the lattice parameters. However, this is impractical because these Fourier coefficients are usually unknown, and there are infinitely many, so we have to rely on suboptimal strategies. The variance expression is usually replaced by a figure of merit with fewer parameters, and those parameters are selected by heuristic methods that take into account the class of functions f to be considered.

The aim of this paper is twofold. First, we give a partial overview of current knowledge on randomly-shifted lattice rules from a practical viewpoint. Then, we examine the issues that arise when we want to exploit this theoretical knowledge in applications. In particular, we explore the impact of the choice of figure of merit, the choice of weights given to the different subsets of coordinates in discrepancy measures, we compare empirical performances of these choices in terms of the RQMC variance, we compare the convergence rate for the variance that are typically observed empirically (for reasonable values of n) to the theoretical asymptotic rates (when n goes to infinity) which are based on bounds that are usually not tight, and see what can be observed in the common situation where the integrand is discontinuous or unbounded. We always assume that s is fixed. We do not consider complexity and tractability issues.

The remainder is organized as follows. In Sect. 2, we recall basic definitions and known results on randomly-shifted lattice rules and the corresponding explicit variance expressions. In Sect. 3, we discuss how we could conceivably select a lattice adaptively to reduce the variance expression if the Fourier coefficients of f were known, or could be estimated easily when needed. The main purpose is to show the difficulty of doing this. We describe and implement a selection method that starts with a large set of lattices and eliminates them one by one, by visiting a sequence of important terms in the variance expression and by keeping, at each step, only the lattices that eliminate those large variance terms. The procedure is very effective on the small examples on which we try it, where the Fourier coefficients are known. But for typical real-life problems, the Fourier coefficients are unknown and estimating them would be too time-consuming, so we need other heuristics. In Sect. 4, we examine previously-proposed figures of merit defined as discrepancies that assign a weight to each subset of coordinates (or projection), using a functional ANOVA decomposition of f, and we suggest ways of specifying the weights as functions of the ANOVA variance components, for Sobolev classes of integrands with square integrable partial derivatives up a given order. When s is large, having a different weight for each projection may give a model with too many parameters. Parameterized choices of weights with fewer parameters are discussed in Sect. 5. They include order-dependent weights and product weights, in particular, and we examine ways of setting those weights. In Sect. 6, we discuss figures of merit defined in terms of the lengths of shortest nonzero vectors in dual lattices. In Sect. 7, we briefly recall the algorithms we have used to search for good lattices with respect to the selected figures of merit. Then, in the following sections, we perform empirical experiments with some examples. Our goal is to estimate the convergence rate of the RQMC variance as a function of n and the variance reduction compared with standard MC, in the practical range of values of n, and to assess the impact of the choice of figure of merit (and weights) on this variance, at least for our selected examples. Motivated by the fact that discontinuous integrands are very frequent in applications, we start in Sect. 9 with simple indicator functions. We give examples where lattice rules are still effective, but where a standard measure of discrepancy can be (sometimes) a very bad predictor of the performance. This illustrates the difficulty of defining good and robust figures of merit in general. In one case, we make the integrand continuous by taking a conditional expectation with respect to one random variable (after generating the other ones) and we examine the effect of this. In Sect. 10, we consider a stochastic activity network example inspired from a real-life application, where the integrand is also an indicator function, and we extend the study made in Sect. 9 to this slightly more complicated setting. The examples of Sect. 9 were constructed as simplifications of that of Sect. 10, to try to better understand the behavior of randomly-shifted lattice rules in those situations. Finally, in Sect. 11, we consider the pricing of Asian-style options, with and without a barrier. Our examples have been inspired from real-life problems, and as it turns out, none of them satisfies the smoothness conditions that guarantee a fast convergence of the variance (such as $\mathcal{O}(n^{-4})$) with the best lattice constructions. This seems to correspond to many typical real-life problems. An online appendix provides detailed results of our experiments.

The good news is that in the great majority of cases, most reasonable choices of figures of merit and weights provide lattices that perform well, for those examples, provided that none of the relevant weights is zero and the irrelevant weights do not dominate too much. This means that there is no need to work hard to estimate the ANOVA variances accurately. Faced with an important application, one may want to spend a small fraction of the available computing budget at the beginning to estimate ANOVA components very roughly, or to just explore a few choices of weights and compare the variances in pilot runs. Also, the convergence rate of the variance observed empirically for reasonable values of n (up to a few millions) is slower than the asymptotic rates proved theoretically for smooth functions. On the other hand, this observed rate is always better than $\mathcal{O}(1/n)$, even for discontinuous and unbounded integrands, in our examples.

2 Randomly-Shifted Lattice Rules

An *integration lattice* is a discrete vector space defined by

$$L_s = \left\{ \mathbf{v} = \sum_{j=1}^{s} z_j \mathbf{v}_j \text{ such that each } z_j \in \mathbb{Z} \right\},$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent over $\mathbb{R}$ and L_s contains $\mathbb{Z}^s$, the integer vectors. A *lattice rule* approximates μ by the average of $f(\mathbf{u}_0), \ldots, f(\mathbf{u}_{n-1})$, where $P_n^0 = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\} = L_s \cap [0,1)^s$. Almost all lattice rules used in practice have *rank 1*, which means that the points of P_n^0 can be enumerated as $\mathbf{u}_i = i\mathbf{v}_1 \bmod 1$ for $i = 0, \ldots, n-1$, where $n\mathbf{v}_1 = \mathbf{a}_1 = (a_1, \ldots, a_s) \in \{0, 1, \ldots, n-1\}$ is the *generating vector*. We have a *Korobov rule* if $\mathbf{a}_1$ has the form $\mathbf{a}_1 = (1, a, a^2 \bmod n, \dots)$ for some integer $a \in \mathbb{Z}_n$. For more details on lattice rules, see [4, 11, 19, 21]. For any subset of coordinates $\mathfrak{u} \subseteq \{1, \dots, s\}$, the *projection* $L_s(\mathfrak{u})$ of L_s over the subspace determined by $\mathfrak{u}$ is also a lattice, in $|\mathfrak{u}|$ dimensions. In this paper, we assume that all lattices are of rank 1 and that the coordinates $a_1, \dots, a_s$ of the generating vector are *all* relatively prime to n (when n is prime, this is automatic). When the latter holds for the first coordinate, the lattice rule is called a rank-1 simple rule [4]. Here we are assuming more: our assumption implies that the projection of P_n^0 over the subspace determined by any nonempty subset of coordinates contains exactly n points and this projection is always $\{0, 1/n, \dots, (n-1)/n\}$ in the case of a single coordinate. Therefore, there is no need to measure the uniformity of these one-dimensional projections.

The point set P_n^0 can be turned into an RQMC point set P_n by a *random shift modulo 1*, defined as follows [5, 21]: Generate a single random point $\mathbf{U}$ uniformly over $(0, 1)^s$ and add it to each point of P_n^0, modulo 1, coordinate-wise: $\mathbf{U}_i = (\mathbf{u}_i + \mathbf{U}) \bmod 1$. Then, each $\mathbf{U}_i$ is uniformly distributed over $[0, 1)^s$ and $\hat{\mu}_{n,\mathrm{rqmc}}$ is an unbiased estimator of μ, while P_n inherits the lattice structure of P_n^0.

A key issue is whether (and when) $\hat{\mu}_{n,\mathrm{rqmc}}$ has smaller variance than the MC estimator $\hat{\mu}_n$. An exact expression for the variance can be obtained in terms of the Fourier coefficients of the integrand f, as follows. If f has Fourier expansion

$$f(\mathbf{u}) = \sum_{\mathbf{h} \in \mathbb{Z}^s} \hat{f}(\mathbf{h}) e^{2\pi\sqrt{-1}\mathbf{h}^{\mathrm{t}}\mathbf{u}},$$

then (see [15])

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} |\hat{f}(\mathbf{h})|^2, \tag{3}$$

where $L_s^* = \{\mathbf{h} \in \mathbb{R}^s : \mathbf{h}^{\mathrm{t}}\mathbf{v} \in \mathbb{Z} \text{ for all } \mathbf{v} \in L_s\} \subseteq \mathbb{Z}^s$ is the *dual lattice* to L_s. This variance depends on f and on L_s. For any given f, an optimal lattice L_s from the viewpoint of variance reduction would minimize $D^2(P_n^0) = \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}]$. This suggests a figure of merit of the general form

$$\mathscr{M}_w(P_n^0) = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} w(\mathbf{h}), \tag{4}$$

with weights $w(\mathbf{h})$ that mimic the anticipated behavior of the $|\hat{f}(\mathbf{h})|^2$. It may be tempting to refer to (4) as a measure of discrepancy. However it does not necessarily measure a departure from the uniform distribution. For certain functions f, the best lattice does not necessarily cover the space very evenly.

It is known [21] that if f has square-integrable mixed partial derivatives up to order α (an integer), and the periodic continuations of its derivatives up to order $\alpha - 2$ are *continuous* across the unit cube boundaries, then

$$|\hat{f}(\mathbf{h})|^2 = \mathscr{O}((\max(1, h_1), \dots, \max(1, h_s))^{-2\alpha}). \tag{5}$$

Moreover, there is a rank-1 lattice with $\mathbf{v}_1 = \mathbf{v}_1(n)$ such that

$$\mathscr{P}_{2\alpha} = \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha} = \mathscr{O}(n^{-2\alpha+\delta}) \tag{6}$$

for any $\delta > 0$. Note that $\mathscr{P}_{2\alpha}$ in (6) is the RQMC variance for a *worst-case* f having

$$|\hat{f}(\mathbf{h})|^2 = (\max(1, h_1), \ldots, \max(1, h_s))^{-2\alpha},$$

so the convergence order in (6) applies when (5) holds. This worst-case f is not necessarily representative of functions encountered in applications, and therefore, $\mathscr{P}_{2\alpha}$ is not necessarily the most appropriate figure of merit.

For the preceding bound to hold with $\alpha \geq 2$, the periodic continuation of f must be continuous. When it is not, which is often the case, f can be transformed into a function $\tilde{f}$ having the same integral and a continuous periodic continuation, by compressing the graph of f horizontally along each coordinate and then making a mirror copy with respect to 1/2. This gives $\tilde{f}(u_1, \ldots, u_s) = f(v_1, \ldots, v_s)$ where $v_j = 2u_j$ for $u_j < 1/2$ and $v_j = 2(1 - u_j)$ for $u_j \geq 1/2$. In practice, instead of changing f, we would stretch the (randomized) points by a factor of 2 along each coordinate, and fold them back. This is equivalent. That is, each coordinate $U_{i,j}$ of $\mathbf{U}_i$ is replaced by $2U_{i,j}$ if $U_{i,j} < 1/2$ and by $2(1 - U_{i,j})$ otherwise. This is the *baker's transformation*. When f is sufficiently smooth, this can reduce the RQMC variance from $\mathscr{O}(n^{-2+\delta})$ to $\mathscr{O}(n^{-4+\delta})$ [12].

3 Adaptive Search for Lattices that Avoid the Large Fourier Coefficients

Searching for a lattice that minimizes the variance expression (3) for each f that we want to integrate is certainly impractical, because the Fourier coefficients are usually unknown and there are infinitely many. If we estimate them, we would have to do it for each f and this is likely to take more time than applying RQMC to estimate μ. We nevertheless explore empirically, in this section, what we could do if we knew (or could estimate on-demand, at low cost) those Fourier coefficients and how much we could gain by exploiting this knowledge (or by finding the optimal lattice for the problem at hand in any other way). In situations where the gain can be significant, it may be worthwhile to investigate ways of identifying the most important Fourier coefficients.

We start with a simple function for which we know the Fourier expansion. But even in that case, the figure of merit (the variance) in (3) involves an infinite number of terms. Heuristic ways of handling this could be to truncate the sum to a finite subset $B \subset \mathbb{Z}^s$,

$$\sum_{\mathbf{0} \neq \mathbf{h} \in L_s^* \cap B} |\hat{f}(\mathbf{h})|^2,$$

Algorithm 1 : Dual-Space Exploration

```
Require: a set of lattices $\mathscr{L}$ and a weight function $w$
  $\mathscr{Q} \leftarrow \mathscr{N}(\mathbf{0})$    // vectors $\mathbf{h}$ to be visited, sorted by decreasing weight $w(\mathbf{h})$
  $\mathscr{M} \leftarrow \mathscr{N}(\mathbf{0})$    // vectors $\mathbf{h}$ that have already entered $\mathscr{Q}$
  while $|\mathscr{L}| > 1$ do
    $\mathbf{h} \leftarrow$ remove first vector from $\mathscr{Q}$
    for all lattices $L_s \in \mathscr{L}$ such that $\mathbf{h} \in L_s^*$ do
      remove $L_s$ from $\mathscr{L}$
      if $|\mathscr{L}| = 1$ then
        return the single lattice $L_s \in \mathscr{L}$ and exit
      end if
    end for
    for all $\mathbf{h}' \in \mathscr{N}(\mathbf{h}) \setminus \mathscr{M}$ do
      add $\mathbf{h}'$ to $\mathscr{M}$ and to $\mathscr{Q}$ with priority (weight) $w(\mathbf{h}')$
    end for
  end while
```

or to the largest q square coefficients $|\hat{f}(\mathbf{h})|^2$. But this is hard to implement. The following heuristic truncates the sum adaptively by exploring the dual space. It makes sense in the situation where the $|\hat{f}(\mathbf{h})|^2$ tend to decrease with each $|h_j|$. It starts with a large set $\mathscr{L}$ of lattices (or a large set of generating vectors $\mathbf{v}_1$, for a given n). Then the method searches for vectors $\mathbf{h}$ with large weights $w(\mathbf{h}) = |\hat{f}(\mathbf{h})|^2$, via a neighborhood search starting at $\mathbf{h} = \mathbf{0}$, keeping a sorted list (as in Dijkstra's shortest path algorithm), and eliminates successively from $\mathscr{L}$ the lattices whose dual contains $\mathbf{h}$ for the next largest $w(\mathbf{h})$ found so far, until a single lattice remains. It is stated as Algorithm 1 (the scope of the **while** and **for** statements are specified by the indentation). The ordered set $\mathscr{Q}$ can be implemented as a priority queue. This algorithm requires a definition of neighborhood in the space $\mathbb{Z}^s$ of vectors $\mathbf{h}$. For example, one can define the neighborhood of $\mathbf{h}$, $\mathscr{N}(\mathbf{h})$, as the set of vectors that differ from $\mathbf{h}$ by only one coordinate, and by one unit only. When the $|\hat{f}(\mathbf{h})|^2$ are unknown, we may think of estimating them whenever they are needed in the algorithm, dynamically.

One can also define a *component-by-component* (CBC) version of this construction algorithm, as follows. For each coordinate j, $j = 1, \ldots, s$, we apply the algorithm for a set $\mathscr{L}$ of j-dimensional lattices with common (fixed) $j - 1$ first coordinates, determined in the previous steps, and we select the jth coordinate by visiting all j-dimensional vectors $\mathbf{h}$ having nonzero jth coordinate, as in Algorithm 1.

Example 1. To experiment with this algorithm, we consider the product V-shaped function

$$f(\mathbf{u}) = \prod_{j=1}^{s} \frac{|4u_j - 2| + c_j}{1 + c_j},$$

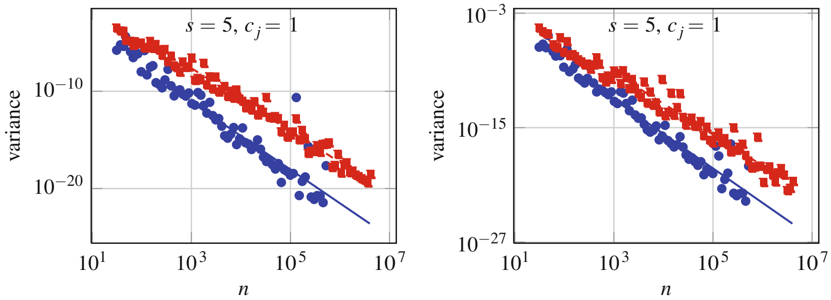


Fig. 1 Estimated variance vs. n, for $s = 5$, in log-log scale, with $c_j = 1$ (*left*) and $c_j = j$ (*right*), using lattices constructed with the dual-space exploration algorithm (●), and the CBC algorithm with the $\mathscr{P}_{\gamma,2}$ criterion (■).

for which

$$\hat{f}(\mathbf{h}) = \prod_{\{j : h_j \text{ is odd}\}} \frac{4}{(1 + c_j)\pi^2 h_j^2}.$$

We take $s = 5$ dimensions, first with $c_j = 1$ and then with $c_j = j$. We applied the CBC version of the dual-space exploration algorithm for prime values of n ranging from $2^5 - 1 = 31$ to $2^{19} - 1 = 524, 287$, to construct a 5-dimensional lattice for each n, then we estimated the RQMC variance for this lattice by the empirical variance with 100 independent random shifts.

Figure 1 shows the empirical variance as a function of n, in the lower (dark) line. The upper line represents the RQMC variance with lattices obtained by a CBC construction using the criteria $\mathscr{P}_{\gamma,2}$ defined in (9), with weights selected based on estimated ANOVA variance components as explained in Sect. 4. This is arguably the best available construction method for general applications among those that we have tried in our experiments. The figure shows that for this small example, our dual-space exploration method does much better. The reason is that by constructing the lattice in terms of a figure of merit that takes into account the individual Fourier coefficients, we can be more accurate in selecting the vectors $\mathbf{h}$ that we want to eliminate from the dual lattice, and thus kick out more of the important terms from the variance expression (3), than if we use a criterion such as $\mathscr{P}_{\gamma,2}$ that just put weights on subsets of coordinates.

For the dual-space exploration, with $n = 2^{16} + 1$, the variance was reduced by a factor of 1.7×10^{14} for $c_j = 1$ and 3.0×10^{16} for $c_j = j$, compared with MC. Empirically, the variance decreases approximately as $\mathscr{O}(n^{-3.46})$ for $c_j = 1$ and $\mathscr{O}(n^{-3.61})$ for $c_j = j$. (There is one outlying value, for $n = 2^{17} - 1 = 131, 071$, where the algorithm did poorly for $c_j = 1$, as can be seen in Fig. 1.) For the lattice constructions based on $\mathscr{P}_{\gamma,2}$, on the other hand, the variance was reduced by only 1.8×10^{12} in the best case, and decreased empirically (approximately) as $\mathscr{O}(n^{-3.24})$.

We also tried with the $\mathscr{M}'_{\gamma,2}$ criterion defined in (14), and the results were worse than with $\mathscr{P}_{\gamma,2}$.

The dual-space exploration algorithm performs much better, for this small example, than the other methods discussed in forthcoming sections. However, in typical situations, the Fourier coefficients are unknown, not always monotonously decreasing with the components of **h**, have to be estimated during the exploration, and the dimension can be much larger than 5. Then, this search approach is unlikely to remain practical and effective. We will discuss alternatives in the following.

4 ANOVA Decomposition and Projection-Dependent Weights

Given that the Fourier expansion and the sum (3) have too many terms to be convenient figures of merit for selecting the lattice parameters, one could seek decompositions of f into a smaller number of terms than in (3), and define measures that take into account the relative importance of those terms. A popular one is the functional ANOVA decomposition [8, 18, 20], where $f(\mathbf{u}) = f(u_1, \ldots, u_s)$ is written as

$$f(\mathbf{u}) = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} f_{\mathfrak{u}}(\mathbf{u}) = \mu + \sum_{i=1}^{s} f_{\{i\}}(u_i) + \sum_{i,j=1:j\neq i}^{s} f_{\{i,j\}}(u_i, u_j) + \cdots$$

where

$$f_{\mathfrak{u}}(\mathbf{u}) = \int_{[0,1)^{|\bar{\mathfrak{u}}|}} f(\mathbf{u})\, \mathrm{d}\mathbf{u}_{\bar{\mathfrak{u}}} - \sum_{\mathfrak{v}\subset\mathfrak{u}} f_{\mathfrak{v}}(\mathbf{u}_{\mathfrak{v}})$$

$\bar{\mathfrak{u}}$ is the complement of $\mathfrak{u}$, and $\mathbf{u}_{\mathfrak{v}}$ refers to the projection of $\mathbf{u}$ on the subspace determined by $\mathfrak{v}$. The *MC variance* then decomposes as

$$\sigma^2 = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \sigma_{\mathfrak{u}}^2, \quad \text{where } \sigma_{\mathfrak{u}}^2 = \mathrm{Var}[f_{\mathfrak{u}}(\mathbf{U})].$$

The variance components $\sigma_{\mathfrak{u}}^2$ can be estimated by the algorithm given in [25], using either MC or (preferably) RQMC to estimate the integrals.

For any $\mathbf{h} \in \mathbb{Z}^s$, let

$$\mathfrak{u}(\mathbf{h}) = \mathfrak{u}(h_1, \ldots, h_s) = \{j \in \{1, \ldots, s\} : h_j \neq 0\}.$$

The RQMC variance with a randomly-shifted lattice rule decomposes as:

$$\mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}] = \sum_{\emptyset\neq\mathfrak{u}\subseteq\{1,\ldots,s\}} \sum_{\mathbf{h}\in L_s^*:\mathfrak{u}(\mathbf{h})=\mathfrak{u}} |\hat{f}(\mathbf{h})|^2 = \sum_{\mathfrak{u}\subseteq\{1,\ldots,s\}} \mathrm{Var}[\hat{\mu}_{n,\mathrm{rqmc}}(f_{\mathfrak{u}})]. \quad (7)$$

The idea here is to adopt a criterion as in (4), but with weights $w(\mathbf{h})$ that depend on a smaller number of parameters, namely one parameter per projection $\mathfrak{u}$. For this, following [7] and others, we take

$$w(\mathbf{h}) = \gamma_{\mathfrak{u}(\mathbf{h})} \prod_{j \in \mathfrak{u}} h_j^{-2\alpha} \tag{8}$$

for all $\mathbf{h} \in \mathbb{Z}^s$, where α is a positive integer to be selected, and the $\gamma_{\mathfrak{u}}$ are arbitrary positive real numbers which we call *projection-dependent weights*. Some authors call them *general weights* [6, 7], although their form is much less general than the arbitrary weights $w(\mathbf{h})$ in (4). With the weights (8), the figure of merit (4) becomes the *weighted* $\mathscr{P}_{2\alpha}$ criterion [7]:

$$\begin{aligned}\mathscr{P}_{\gamma,2\alpha}(P_n^0) &= \sum_{\mathbf{0} \neq \mathbf{h} \in L_s^*} \gamma_{\mathfrak{u}(\mathbf{h})} (\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha} \\ &= \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,s\}} \frac{1}{n} \sum_{i=0}^{n-1} \gamma_{\mathfrak{u}} \left[\frac{-(-4\pi^2)^\alpha}{(2\alpha)!} \right]^{|\mathfrak{u}|} \prod_{j \in \mathfrak{u}} B_{2\alpha}(u_{i,j}), \end{aligned} \tag{9}$$

where $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}) = i\mathbf{v}_1 \bmod 1$ is the ith lattice point before the shift, $|\mathfrak{u}|$ is the cardinality of $\mathfrak{u}$, and $B_{2\alpha}$ is the Bernoulli polynomial of order 2α.

This criterion comes naturally in the following setting. Let $\mathscr{F}_\alpha$ be the class of functions $f : [0,1)^s \to \mathbb{R}$ for which for each $\mathfrak{u} \subseteq \{1,\ldots,s\}$, the partial derivative of order α with respect to $\mathfrak{u}$ is square integrable, and (if $\alpha \geq 2$) the partial derivatives of orders 0 to $\alpha - 2$ of the periodic continuation of f over $\mathbb{R}^s$ are continuous. Define the square variation of $f \in \mathscr{F}_\alpha$ by

$$V_\gamma^2(f) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} V_\gamma^2(f_{\mathfrak{u}}) = \sum_{\emptyset \neq \mathfrak{u} \subseteq \{1,\ldots,s\}} \frac{1}{\gamma_{\mathfrak{u}} (4\pi^2)^{\alpha|\mathfrak{u}|}} \int_{[0,1]^{|\mathfrak{u}|}} \left| \frac{\partial^{\alpha|\mathfrak{u}|}}{\partial \mathbf{u}^\alpha} f_{\mathfrak{u}}(\mathbf{u}) \right|^2 d\mathbf{u} \tag{10}$$

(which depends on the $\gamma_{\mathfrak{u}}$'s). Then, for any constant $K > 0$, the largest RQMC variance over the class of functions $f \in \mathscr{F}_\alpha$ for which $V_\gamma^2(f) \leq K$ is equal to $K\mathscr{P}_{\gamma,2\alpha}(P_n^0)$, and the maximum is reached for a worst-case function whose square Fourier coefficients are

$$|\hat{f}(\mathbf{h})|^2 = K\gamma_{\mathfrak{u}(\mathbf{h})} (\max(1,h_1),\ldots,\max(1,h_s))^{-2\alpha}.$$

See [6, 14] for the details. The constant K is just a scale factor that can be incorporated in the weights $\gamma_{\mathfrak{u}}$, so we can assume that $K = 1$. The worst-case function can then be written as

$$f_\alpha^*(\mathbf{u}) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \sqrt{\gamma_{\mathfrak{u}}} \prod_{j \in \mathfrak{u}} \frac{(2\pi)^\alpha}{\alpha!} B_\alpha(u_j).$$

The ANOVA variance components for this function are

$$\sigma_{\mathfrak{u}}^2 = \gamma_{\mathfrak{u}} \left[\mathrm{Var}[B_\alpha(U)] \frac{(4\pi^2)^\alpha}{(\alpha!)^2} \right]^{|\mathfrak{u}|} = \gamma_{\mathfrak{u}} \left[|B_{2\alpha}(0)| \frac{(4\pi^2)^\alpha}{(2\alpha)!} \right]^{|\mathfrak{u}|} \overset{\text{def}}{=} \gamma_{\mathfrak{u}} (\kappa(\alpha))^{-|\mathfrak{u}|} \tag{11}$$

where $\kappa(\alpha)$ is a constant that depends on α. In particular, we have

$$\kappa(1) = \frac{3}{\pi^2} \approx 0.30396, \quad \kappa(2) = \frac{45}{\pi^4} \approx 0.46197, \quad \kappa(3) \approx 0.49148,$$

and $\kappa(\alpha)$ increases with α for $\alpha \geq 1$ and converges to $1/2$ when $\alpha \to \infty$.

To be consistent with our choice of $\mathscr{P}_{\gamma,2\alpha}$ as a criterion, we can select the weights $\gamma_{\mathfrak{u}}$ as if the function f that we want to integrate has the same form as f_α^*. That is, we take the weights given by the formula

$$\gamma_{\mathfrak{u}} = \sigma_{\mathfrak{u}}^2 (\kappa(\alpha))^{|\mathfrak{u}|},$$

in which the variance components $\sigma_{\mathfrak{u}}^2$ are replaced by estimates. These estimates can be obtained with the algorithm of [25], for example. This formula can be generalized slightly to

$$\gamma_{\mathfrak{u}} = \sigma_{\mathfrak{u}}^2 \rho^{|\mathfrak{u}|}, \tag{12}$$

where $0 < \rho \leq 1$ is a constant to be selected. In view of the behavior of $\kappa(\alpha)$, it makes sense to take $\rho \leq 0.5$, and smaller when we think that f is less smooth.

It is known that for any $\alpha > 1$, any $\delta > 0$, and any choices of weights $\gamma_{\mathfrak{u}}$, there are rank-1 lattices for which $\mathscr{P}_{\gamma,2\alpha}(P_n^0) = \mathscr{O}(n^{-2\alpha+\delta})$, and the corresponding vectors $\mathbf{v}_1$ can be constructed explicitly one coordinate at a time, by a component-by-component construction method [6].

5 Further Heuristics for Choosing the Weights

In (9), there are $2^s - 1$ parameters $\gamma_{\mathfrak{u}}$ to specify, which is too many when s is large. It is also hard to estimate these $\sigma_{\mathfrak{u}}^2$ with reasonable relative error when they are small compared with σ^2 and this typically occurs for most subsets $\mathfrak{u}$ when $|\mathfrak{u}|$ increases. This motivates the introduction of more parsimonious models for the weights, with fewer parameters. As mentioned in the first paragraph of Sect. 2, the one-dimensional projections are all the same under our assumptions, so the weights of the one-dimensional subsets $|\mathfrak{u}|$ are irrelevant and we can set them to zero in the selection criterion; that is, restrict the sum in (9) to the subsets $\mathfrak{u}$ of cardinality $|\mathfrak{u}| \geq 2$. We always do that in our experiments when searching for good lattices. Note that multiplying all weights by the same constant has no impact on the selection of $\mathbf{v}_1$, since it does not change the relative importance of the projections, so we can fix one of them (the largest one, for example) to 1. But there still remains $2^s - s - 2$ projections weights to specify.

One way to reduce the number of parameters in (9) (and the likelihood of overfitting) is to bundle (partition) the subsets $\mathfrak{u}$ in subgroups, and force the same $\gamma_{\mathfrak{u}}$ within each subgroup. A well-known example of this is to take *order-dependent weights*, where $\gamma_{\mathfrak{u}}$ depends only on the cardinality of $\mathfrak{u}$, say $\gamma_{\mathfrak{u}} = \gamma_r$ when $|\mathfrak{u}| = r$, for $r = 2, \ldots, s$. To specify those γ_r, we can estimate each $\sigma_r^2 = \sum_{\{\mathfrak{u}:|\mathfrak{u}|=r\}} \sigma_{\mathfrak{u}}^2$, which represents the total variance captured by the $\binom{s}{r}$ projections of order r, and plug it in the formula

$$\gamma_r = C\rho^r \sigma_r^2 \binom{s}{r}^{-1},$$

where $C > 0$ is an arbitrary scaling constant. This gives $s-1$ parameters to estimate.

In one special case, we can simply assume that $\gamma_r = \gamma^{r-2}$ for all $r \geq 2$, for some constant γ, and estimate γ by least-squares fitting of the linear regression model

$$\ln C + r \ln \rho + 2 \ln \sigma_r - \ln \binom{s}{r} = (r-2) \ln \gamma + \varepsilon_r$$

(for example), by finding $\ln C$ and $\ln \gamma$ that minimize $\sum_{r=2}^{\infty} \varepsilon_r^2$. The resulting weights are *geometric order-dependent weights*. With *constant order-truncated weights*, one simply takes $\gamma_{\mathfrak{u}} = 1$ for $|\mathfrak{u}| \leq d$ and $\gamma_{\mathfrak{u}} = 0$ otherwise, for a given integer $d \geq 2$. Wang [26] suggests this with $d = 2$.

A different type of parameterization, used in [10, 11, 24], assigns a weight γ_j to each coordinate j and uses the *product weights* $\gamma_{\mathfrak{u}} = \prod_{j \in \mathfrak{u}} \gamma_j$. Again, we can estimate the parameters γ_j by matching the ANOVA variances, ignoring the one-dimensional projections. One way of doing this is to fit the weights (12) where the variance components are estimated, over all *two-dimensional* projections, via a least-squares procedure. Then we rescale all the weights by a constant factor to match the ratio of average estimated weights (12) over the *three-dimensional* projections to that over the two-dimensional projections.

More specifically, we first minimize

$$R = \sum_{k=1}^{s} \sum_{j=1}^{k-1} \left(\tau_j \tau_k - \rho^2 \sigma_{\{j,k\}}^2 \right)^2$$

with respect to $\tau_1, \ldots, \tau_s$, where τ_j can be viewed as the unscaled weight for projection j, and where the variance components $\sigma_{\mathfrak{u}}^2$ for $|\mathfrak{u}| = 2$ are replaced by their estimates. Differentiating this expression with respect to τ_j and equaling to 0, we obtain, for each j,

$$\tau_j \sum_{k=1, k \neq j}^{s} \tau_k^2 = \sum_{k=1, k \neq j}^{s} \tau_k \rho^2 \sigma_{\{j,k\}}^2.$$

We solve this (heuristically) by an iterative fixed-point algorithm:

$$\tau_j^{(0)} = \max_{k,l=1,\ldots,s} \rho\sigma_{\{k,l\}}, \qquad \tau_j^{(i+1)} = \frac{\sum_{k=1,k\neq j}^{s} \tau_k^{(i)} \rho^2 \sigma_{\{j,k\}}^2}{\sum_{k=1,k\neq j}^{s} \left(\tau_k^{(i)}\right)^2},$$

for $i = 1, 2, \ldots$. We then rescale the weights via $\gamma_j = c\tau_j$ where the constant c satisfies

$$\frac{\sum_{k=1}^{s} \sum_{j=1}^{k-1} \tau_j \tau_k}{\sum_{k=1}^{s} \sum_{j=1}^{k-1} \sum_{l=1}^{j-1} \tau_j \tau_k \tau_l} = c\, \frac{\sum_{k=1}^{s} \sum_{j=1}^{k-1} \rho^2 \sigma_{\{j,k\}}^2}{\sum_{k=1}^{s} \sum_{j=1}^{k-1} \sum_{l=1}^{j-1} \rho^3 \sigma_{\{j,k,l\}}^2}$$

in which the sum of weights of order 3 is again replaced by an estimate.

6 Figures of Merit Based on the Spectral Test

In view of the variance expression (3) and its decomposition (7), and because we normally expect the square Fourier coefficients $|\hat{f}(\mathbf{h})|^2$ to decrease with the size of $\mathbf{h}$ (when f is smooth, we know from (5) that these coefficients must converge at the given rate), it seems to make sense to define a criterion that penalizes the short non-zero vectors $\mathbf{h}$ in the dual lattice L_s^*, as well as in the dual lattices $(L_s(\mathfrak{u}))^*$ to the projections $L_s(\mathfrak{u})$. Note that $(L_s(\mathfrak{u}))^*$ is the projection over $\mathfrak{u}$ of $\{\mathbf{h} \in L_s^* : \mathfrak{u}(\mathbf{h}) \subseteq \mathfrak{u}\}$, but not the projection of L_s^* over $\mathfrak{u}$.

For each $\mathfrak{u}$, one can compute the Euclidean length $\ell_{\mathfrak{u}}$ of a shortest nonzero vector in $(L_s(\mathfrak{u}))^*$. There is a known tight theoretical upper bound $\ell_r^*(n)$ on the length of a shortest nonzero vector in a lattice with n points per unit of volume in r dimensions [3, 15], and we can divide $\ell_{\mathfrak{u}}$ by $\ell_{|\mathfrak{u}|}^*(n)$ to obtain a standardized measure between 0 and 1, and raise it to some power $\beta > 0$, for each $\mathfrak{u}$, or take the reciprocal to obtain a measure of non-uniformity larger or equal to 1. To give more weight to more important projections, this measure can in turn be multiplied by some weight $\gamma_{\mathfrak{u}}$, for each $\mathfrak{u}$. Then we can take either the sum or the minimum (worst case) over a selected class $\mathscr{J}$ of nonempty subsets $\mathfrak{u}$ of $\{1, \ldots, s\}$. The role of β is to amplify or reduce the relative importance of bad projections (those having a small value of $\ell_{\mathfrak{u}}/\ell_{|\mathfrak{u}|}^*(n)$) in the criterion. This gives the following figures of merit

$$\mathscr{M}_{\gamma,\beta}(P_n^0) = \sum_{\mathfrak{u}\in\mathscr{J}} \gamma_{\mathfrak{u}} \left(\frac{\ell_{\mathfrak{u}}}{\ell_{|\mathfrak{u}|}^*(n)}\right)^{\beta}, \tag{13}$$

$$\mathscr{M}'_{\gamma,\beta}(P_n^0) = \sum_{\mathfrak{u}\in\mathscr{J}} \gamma_{\mathfrak{u}} \left(\frac{\ell_{|\mathfrak{u}|}^*(n)}{\ell_{\mathfrak{u}}}\right)^{\beta}, \tag{14}$$

$$\widetilde{\mathscr{M}}_{\gamma,\beta}(P_n^0) = \min_{\mathfrak{u}\in\mathscr{J}} \gamma_{\mathfrak{u}} \left(\frac{\ell_{\mathfrak{u}}}{\ell^*_{|\mathfrak{u}|}(n)}\right)^{\beta}, \quad \text{and} \tag{15}$$

$$\widetilde{\mathscr{M}}'_{\gamma,\beta}(P_n^0) = \max_{\mathfrak{u}\in\mathscr{J}} \gamma_{\mathfrak{u}} \left(\frac{\ell^*_{|\mathfrak{u}|}(n)}{\ell_{\mathfrak{u}}}\right)^{\beta}. \tag{16}$$

The criteria (13) and (15) are to be maximized, whereas (14) and (16) are to be minimized. In (15) and (16), only the quality of the worst-case projections matters, and we do not care about the quality of the other ones, whereas in (13) and (14), the quality of all the projections contributes to the sum, so these criteria encourage improvements on all projections, not only the worst ones. The two variants (15) and (16) are equivalent in terms of which lattice maximizes or minimizes them, if we invert the weights (although we do not invert the weights in our examples). On the other hand, (13) and (14) are really different. For a fixed β and fixed weights, in (13) the bad projections have a small importance in the sum (they only "fail to score high") whereas in (14) they have more importance because they bring a large penalty.

The computing time of $\ell_{\mathfrak{u}}$ increases only very slowly with n (roughly at a logarithmic rate), in contrast to that of $\mathscr{P}_{\gamma,2\alpha}$, but on the other hand it is exponential in s in the worst case. In practice, it can be computed reasonably quickly for s up to 30 or so, and n as large as we want. A computational advantage of the criteria (15) and (16) is that poor lattices can be eliminated quickly (on average) without having to compute all the $\ell_{\mathfrak{u}}$'s. For example, in (15), the lattice can be eliminated as soon as we have a small enough upper bound on $\gamma_{\mathfrak{u}}\ell_{\mathfrak{u}}/\ell^*_{|\mathfrak{u}|}(n)$ for some $\mathfrak{u}$ (for this, we do not even need to know $\ell_{\mathfrak{u}}$ exactly). For all the criteria based on a sum to be minimized, we can also stop and eliminate the lattice whenever the sum exceeds a given value (e.g., if it exceeds the best value found so far).

A special case of (15) was used in [15], with $\beta = 1$ and

$$\begin{aligned}\mathscr{J} &= \mathscr{J}(t_1,\ldots,t_d)\\ &= \{\mathfrak{u} = \{1,\ldots,r\} \text{ for } 2 \le r \le t_1\}\\ &\quad \cup\{\mathfrak{u} = \{j_1,\ldots,j_r\} \text{ such that } 1 = j_1 < \cdots < j_r \le t_r \text{ and } 2 \le r \le d\}.\end{aligned}$$

This was inspired by criteria used for random number generators having a lattice structure [13]. The main drawback of this criterion is that many projections are not considered at all; they can be very bad and this is not reflected by the figure of merit.

All these criteria can also be defined based on the lengths of the shortest vectors in the *primal lattices* $L_s(\mathfrak{u})$, instead of their dual lattices $(L_s(\mathfrak{u}))^*$, and permuting minimization for maximization. This makes little difference in terms of the uniformity of retained lattices. The length of the shortest vector in $L_s(\mathfrak{u})$ represents the minimal distance between any two lattice points, and we want this distance to be as large as possible.

7 Searching for Lattice Parameters

Once we have selected a discrepancy measure (or figure of merit) and specified the weights, the next step is to search for lattices that minimize this measure, for a given n. In our experiments, we will use (and compare) the following strategies.

In the case of Korobov lattices, there is a single parameter that can take at most $n - 1$ values, so we will simply make an exhaustive search for the best vector $\mathbf{a}_1 = (1, a, a^2, \ldots, \ldots)$ over all admissible integers a.

For general rank-1 lattices, under our assumptions, there could be up to $(n-1)^{s-1}$ combinations and an exhaustive search is usually out of the question (for example, this happens as soon as s exceeds a few units if n is around a million, which is not unusual). A standard construction method in this context is the *component by component (CBC) construction algorithm*, which works as follows [22, 23]:

```
Let $a_1 = 1$;
For $j = 2, \ldots, s$, find $a_j \in \{1, \ldots, n-1\}$, $\gcd(a_j, n) = 1$, such that
    $(a_1, \ldots, a_{j-1}, a_j)$ minimizes the selected figure of merit for the first $j$
    dimensions.
```

We will also use the following streamlined search method, which replaces the exhaustive search over a_j at each step by a search over a small number of different random candidates a_j (the number r of candidates can be from 20 to 100, for example, depending on the computing budget that we are willing to devote to this).

```
Let $a_1 = 1$;
For $j = 2, \ldots, s$, try $r$ random $a_j \in \{1, \ldots, n-1\}$, $\gcd(a_j, n) = 1$,
    and retain the one for which $(a_1, \ldots, a_{j-1}, a_j)$ minimizes the selected
    figure of merit for the first $j$ dimensions.
```

8 Experimental Methodology

We summarize our experimental setting for the empirical results reported in the following sections. For each example where this is relevant, we first estimate the ANOVA variance components of the integrand by the method of [25], using RQMC with $2^{20} - 3 = 1{,}048{,}573$ lattice points and 1,000 independent replications (random shifts). The lattice used for this (for all examples) was constructed by a randomized CBC search with $r = 50$ using geometric order-dependent weights with $\gamma = 0.5$. Next, we select the criteria among (9) or (13)–(16) and the types of weights that we want to consider. The weights are selected as functions of the estimated ANOVA variances, using the strategies described in Sects. 4 and 5. Occasionally, the ANOVA variance estimators are zero or take a smaller value than their precision. Then, we give these projections a weight equal to 1/100 of the smallest nonzero computed weight. For each selected criterion and type of weight, we construct lattices using random CBC searches with $r = \min(50, n-1)$, for 86 different prime values of n ranging from $2^5 - 1 = 31$ to $2^{22} - 3 = 4,194,301$. Then, for each retained lattice, we estimate the RQMC variance with 100 independent replications.

When constructing lattices with the $\mathcal{P}_{\gamma,2\alpha}$ criterion for use with the baker's transformation, we set $\alpha = 2$; otherwise, we set $\alpha = 1$. The weights for the criteria based on the spectral test are taken simply as $\gamma_{\mathfrak{u}} = \sigma_{\mathfrak{u}}^2$, where the latter is estimated, and we take $\mathcal{J} = \{\mathfrak{u} : \emptyset \neq \mathfrak{u} \subseteq \{1, \ldots, s\}\}$, unless indicated otherwise.

In most cases, the variance behaves approximately linearly in logarithmic scale for $n \geq 10^3$. Then we fit a linear model of the form

$$\ln \mathrm{Var}[\mu_{n,\mathrm{rqmc}}] = \ln a_0 - \nu \ln n + \varepsilon \tag{17}$$

for positive constants a_0 and ν, where ε represents a noise term. We do this by computing the values $\hat{a}_0$ and $\hat{\nu}$ of a_0 and ν that minimize the sum of squares of the values of ε for the 61 (out of 86) values of n that are greater than 10^3. Our estimated (or empirical) convergence rate is then $\mathcal{O}(n^{-\hat{\nu}})$. We report the precision on our estimates of $\hat{\nu}$ via 95% confidence intervals, assuming that ε is normally distributed with mean 0 and variance S_ε^2 (we have checked empirically that this is indeed a reasonable assumption).

In (17), the parameters a_0 and ν tell us how the log-variance decreases "on average" as a function of n, for a given lattice construction procedure and a given example. They are the primary quality indicators for the procedure. The parameter ε represents the departure of the log-variance from the linear model for the particular lattice selected at a given n, together with the estimation error in the RQMC log-variance because it is based on a finite number of replications. The latter error can be made arbitrarily small by making more independent replicates of the RQMC estimator. The departure of the true log-variance from the linear model typically has a larger contribution to ε in our examples. This departure depends on the lattice parameters that are retained by the selection algorithm for the given n; it is intrinsic to the lattice construction procedure and it generally depends on the criterion and type of weights. A small standard deviation S_ε means that the linear model is a better predictor of the performance for a given n, and that the returned lattices tend to be more robust and reliable in terms of RQMC variance. When the linear models for two or more criteria predict similar RQMC variances, we should prefer the one with the smallest S_ε.

We define the variance reduction factor (VRF) for a specific n-point randomly-shifted lattice rule as the variance σ^2 of the MC estimator divided by n times the variance of the RQMC estimator. We estimate σ^2 by the empirical variance S_n^2. In some cases, we replace the RQMC variance of the specific lattice at a given value of n by the variance $\hat{a}_0 n^{-\hat{\nu}}$ interpolated from our linear model in log scale, and we report the corresponding interpolated VRF, $\widehat{\mathrm{VRF}}(n) = n^{\hat{\nu}-1} S_n^2 / \hat{a}_0$, usually with $n = 2^{20}$. This interpolation is more stable than the actual variance at a given n.

Detailed results of our experiments are given in the online appendix. In the following sections, we only summarize these results.

9 An Indicator Function

In our first set of experiments, we consider a simple discontinuous integrand defined as the indicator that a sum of s independent random variables exceeds a given threshold. We assume that $Y_1, \ldots, Y_s$ are independent random variables, and that Y_j is exponential with rate λ_j, for each j. We estimate the probability $\mu = \mathbb{P}[Y_1 + \cdots + Y_s > x]$ by MC or RQMC, for some constant x. The *basic estimator* is $X = \mathbb{I}[Y_1 + \cdots + Y_s > x]$, where $\mathbb{I}$ denotes the indicator function. It corresponds to the discontinuous s-dimensional integrand

$$f(u_1, \ldots, u_s) = \mathbb{I}[F_1^{-1}(u_1) + \cdots + F_s^{-1}(u_s) > x],$$

where $F_j^{-1}(u_j) = -\ln(1 - u_j)/\lambda_j$ is the inverse cdf of Y_j evaluated at u_j.

We also consider the *conditional MC* (CMC) estimator

$$\begin{aligned} X_{\text{cmc}} &= \mathbb{P}[Y_1 + \cdots + Y_s > x \mid Y_1 + \cdots + Y_{s-1}] \\ &= \exp[-\lambda_s(x - Y_1 - \cdots - Y_{s-1})] \cdot \mathbb{I}[x - Y_1 - \cdots - Y_{s-1} \geq 0]. \end{aligned}$$

The associated integrand,

$$f(u_1, \ldots, u_{s-1}) = 1 - F_s(x - F_1^{-1}(u_1) - \cdots - F_{s-1}^{-1}(u_{s-1})),$$

has dimension $s - 1$ and is continuous, but has a discontinuity in its first-order derivatives, because the cdf of Y_s, $F_s(y) = [1 - \exp(-\lambda_s y)] \cdot \mathbb{I}[y > 0]$, has a discontinuous derivative at $y = 0$.

For the one-dimensional case, it is known (see [17]) that the basic RQMC estimator can take only two values and its variance converges as $\mathcal{O}(n^{-2})$ regardless of the choice of lattice. Using the dual-space exploration algorithm here does not work well because the Fourier coefficients do not decrease monotonously with $\|\mathbf{h}\|$.

We simulated these estimators for $s = 2, \ldots, 6$, for the following four cases: $\lambda_j = 1$, $\lambda_j = j^{-1}$, and $\lambda_j = j^{-2}$, with x chosen so that the probability μ to be estimated is close to 0.5, and $\lambda_j = j^{-1}$ with x chosen so that μ is near 0.1.

To select the lattice parameters, we tried the criterion (9) with $\alpha = 1$ for both the basic and CMC estimators, with $\alpha = 2$ for the CMC estimator with the baker transformation, and the criteria (13)–(16) with $\beta = 1$ and 2, with projection-dependent, product, order-dependent and geometric order-dependent weights in all cases. In general, the observed convergence rates (reported in the online appendix) do not vary too much when only λ_j or x changes. Although none of the integrands here meets the smoothness requirements that justify using the $\mathcal{P}_{\gamma,2\alpha}$ criterion, in the sense that we have no guaranteed convergence rate for the variance of the corresponding RQMC estimators, lattices constructed with that criterion and projection-dependent weights gave slightly higher values of $\widehat{\text{VRF}}(2^{20})$ and $\hat{\nu}$ together with smaller values of $\hat{S}_\varepsilon$ on average, compared to those obtained with criteria based on the spectral test. They give empirical convergence rates exponents of approximately $\hat{\nu} \approx (s+1)/s$ for the basic estimator. For the CMC estimator, the best convergence rates, of $\mathscr{O}(n^{-2})$ without the baker's transformation and of $\mathscr{O}(n^{-4})$ with the baker's transformation, are obtained at $s = 2$ and degrade as a function of s down to around $\mathscr{O}(n^{-1.5})$ and $\mathscr{O}(n^{-1.6})$, respectively, at $s = 6$. The improvement on the convergence rate due to the baker's transformation is clear at $s = 2$ or 3 but seems marginal for $s \geq 4$. The observed convergence rates for the CMC estimator for $s = 2$ were expected, because in that case the integrand is one-dimensional, is continuous in $(0, 1)$ but its periodic continuation is not, and these are the known convergence rates for such an integrand [12].

Regarding the choices of weights, our estimations of the ANOVA variances suggested that product and order-dependent weights were not justified, yet we found no clearly observable best choice of weights for the basic estimator. For the CMC estimator, however, projection-dependent weights, when used with the $\mathscr{P}_{\gamma,2\alpha}$ criterion, consistently offer good performance.

We also examined a simpler case with $s = 2$ where $Y_1 \sim U[0, m)$ for some $m \in [0, 1)$ and $Y_2 \sim U[0, 1)$. Our experiments with $m = 0.375$ and $x = 0.8$ revealed that some lattices with excellent values of standard figures of merit, such as $\mathscr{P}_{2\alpha}$ and those based on the spectral test, are not among the best in terms of variance reduction. These criteria, introduced earlier, are not really appropriate in this situation, because they do not take into account the alignment between the lattice points and the discontinuity, which turns out to be a key factor here. On the other hand, even for this specific artificial example, by examining all lattices for a given n, we found a clear positive correlation between the RQMC variance and $\mathscr{P}_{2\alpha}$. Here, the choice of weights is not an issue, because there is a single projection in more than one dimension.

10 Example: A Stochastic Activity Network

We consider the stochastic activity network example taken from [9] and represented in Fig. 2, where the lengths $V_1, \ldots, V_{13}$ of edges $1, \ldots, 13$ are independent random variables with distribution functions $F_1, \ldots, F_{13}$, respectively. We take the same cdf's F_j as in [2, Sect. 4.1]. For the activities $j = 1, 2, 4, 11, 12$, we have $V_j = \max(0, \tilde{V}_j)$ where $\tilde{V}_j$ is a normally distributed random variable with mean θ_j and variance $\theta_j^2/16$. The other V_j's are exponential with mean θ_j. The values of $\theta_1, \ldots, \theta_{13}$ are 13.0, 5.5, 7.0, 5.2, 16.5, 14.7, 10.3, 6.0, 4.0, 20.0, 3.2, 3.2, 16.5, respectively. See [1, 2, 15] for a complete description of the problem. We are interested in estimating the probability that the longest path from source (node 1) to sink (node 9) has a length T larger than some constant x with the estimator $X = \mathbb{I}[T > x]$. We also consider the CMC estimator obtained by simulating the V_j's only for the edges that *are not* in the cut set $\mathscr{L} = \{5, 6, 7, 9, 10\}$, and taking the probability that $T > x$ conditional on those V_j's, as in [1]. This CMC estimator can be written as

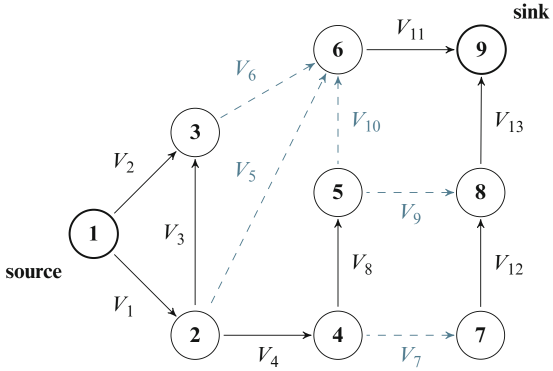


Fig. 2 Graph of the stochastic activity network with 9 nodes and 13 links. The dashed links are not simulated in the CMC variant.

$$X_{\mathrm{CMC}} = \mathbb{P}\left[T > x \mid \{V_j : j \notin \mathscr{L}\}\right] = 1 - \prod_{j\in\mathscr{L}} \mathbb{P}[V_j \le x - P_j] \tag{18}$$

where P_j is the length of the longest path that goes through edge j when we put $V_j = 0$ (i.e., we exclude edge j). The main motivation for considering this estimator is that it is continuous as a function of the V_j's that are generated (and therefore as a function of the underlying uniform random numbers), in contrast to the original estimator X, and it is also easy to compute. This example generalizes the problems and the CMC estimators considered in the previous section. This integrand has dimension $s = 13$ with the basic estimator X, and dimension $s = 8$ with the CMC estimator X_{CMC}. We also estimated $\mathbb{E}[T]$ by simulation; the corresponding integrand has dimension $s = 13$.

For all types of estimators, we have estimated the ANOVA variances and observed that they vary a lot across projections of a given order, so we do not expect order-dependent or geometric weights to work well. In our experiments, we found that the $\mathscr{P}_{\gamma,2\alpha}$ criterion (with $\alpha = 1$ for the standard estimator and $\alpha = 2$ for the CMC estimator) performed well in all cases, with relatively high values of $\widehat{\mathrm{VRF}}(2^{20})$ and $\hat{\nu}$, together with low values of $\hat{S}_\varepsilon$, with slightly better performance for projection-dependent and product weights. We also found that using the inappropriate order-dependent or geometric weights does not guarantee poor performance—in some cases the VRF's were even slightly higher than with the more appropriate projection-dependent and product weights—but it makes it more unpredictable, with VRFs as low as half of the best ones in some cases. The criteria based on the spectral test did not perform as well as $\mathscr{P}_{\gamma,2\alpha}$, at least for projection-dependent and product weights. The standard and CMC estimators had similar qualitative behavior, but the observed VRFs were much larger for the CMC estimator. For example, the best VRF for $x = 60$ interpolated at $n = 2^{20}$ is 27 with an empirical convergence rate of 1.20 for the standard estimator, obtained with the $\mathscr{P}_{\gamma,2\alpha}$ criterion with projection-dependent weights and $\alpha = 1$. For the same case but with CMC estimator and $\alpha = 2$, we observed a fitted VRF of 4.4×10^3 with an empirical convergence rate of 1.51. The baker's transformation offered very little improvement on the CMC estimator. All these results are in the online appendix.

11 Example: Asian Call Option

We consider an Asian call option based on the price $S(t)$ of single asset at times $t_0 = 0, t_1, \ldots, t_s$, with payoff:

$$Y = \mathrm{e}^{-\tilde{r}t_s} \max \left[0,\ \frac{1}{s} \sum_{j=1}^{s} S(t_j) - K \right],$$

where $\tilde{r}$ is the risk-free interest rate and K is the strike price. The asset price is a geometric Brownian motion:

$$S(t) = S(0) \exp[(\tilde{r} - \sigma^2/2)t + \sigma B(t)],$$

where $\{B(t) : t \geq 0\}$ is a standard Brownian motion, and σ it the volatility. We also consider a down-and-in variant of the Asian option with payoff

$$Y' = Y \cdot \mathbb{I}\left[\min_{j=1,\ldots,s} S(t_j) \leq K' \right],$$

where K' is a barrier. We estimate $\mathbb{E}[Y]$ and $\mathbb{E}[Y']$ with MC and RQMC. For our experiments, we set $S(0) = 100$, $K = 100$, $K' = 80$, $\tilde{r} = 0.05$, $\sigma = 0.5$, $t_j = j/s$ for $j = 0, \ldots, s$, and $s = 6$. To simulate $S(t_1), \ldots, S(t_s)$, Y and Y', we sample a standard normal vector $\mathbf{Z} = (Z_1, \ldots, Z_s)$ with $Z_j = \Phi^{-1}(U_j)$, where Φ is the standard normal distribution function. Then we generate $\mathbf{B} = (B(t_1), \ldots, B(t_s)) = \mathbf{AZ}$, where $\mathbf{C} = \mathbf{AA}^{\mathrm{t}}$ is the covariance matrix of $\mathbf{B}$ with elements $c_{j,k} = \sigma^2 \min(t_j, t_k)$ We consider two standard choices for the decomposition $\mathbf{AA}^{\mathrm{t}}$. The first is the Cholesky factorization where $\mathbf{A}$ is a lower triangular matrix. The second, based on principal component analysis (PCA), is $\mathbf{A} = \mathbf{PD}^{1/2}$, where $\mathbf{P}$ is the matrix of right eigenvectors of $\mathbf{C}$ and $\mathbf{D}$ is a diagonal matrix that contains the eigenvalues of $\mathbf{C}$ sorted by increasing order so that the components of $\mathbf{B}$ depend more on the first components of $\mathbf{Z}$ than on the others.

For the Asian option with PCA, our estimations of the ANOVA variances showed that projection $\{1\}$ itself accounts for nearly 99% of the total variance for the Asian option, whereas with Cholesky all projections of order 1 together account for only 73% of the total variance. For the down-and-in option, the largest part of the variance is contributed by projections of order 2 and more, and PCA barely improves the situation with respect to Cholesky by raising from 9% to 14% the percentage of variance contributed by projections of order 1. Note that there is only one projection

Table 1 Fitted variance reduction factors at $n = 2^{20}$ and empirical convergence rates for the Asian and down-and-in options. The baker's transformation was applied for the Asian option, but not for the down-and-in option. When CBC is followed by a value of r, it refers to random CBC, otherwise it refers to exhaustive CBC, and similarly for Korobov. Order-dependent of order 2 is abbreviated as *order 2*.

Criterion	Construction	r	Weight type	$\widehat{\text{VRF}}(2^{20})$	$\hat{\nu}$	$\hat{S}_\varepsilon$
Asian option (PCA), $s = 6$						
$\mathscr{P}_{\gamma,4}$	CBC	50	Proj.-dep.	3.1×10^5	1.846 ± 0.004	0.325
			Product	3.1×10^5	1.840 ± 0.005	0.335
			Order-dep.	1.6×10^5	1.707 ± 0.008	0.632
			Geometric	2.4×10^5	1.784 ± 0.005	0.399
			Order 2	1.4×10^5	1.710 ± 0.010	0.852
		–	Proj.-dep.	3.5×10^5	1.870 ± 0.020	0.317
	Korobov	50	Proj.-dep.	2.6×10^5	1.825 ± 0.005	0.354
		–	Proj.-dep.	3.0×10^5	1.850 ± 0.010	0.333
$\mathscr{M}_{\gamma,2}$	CBC	50	Proj.-dep.	1.7×10^5	1.751 ± 0.007	0.545
$\mathscr{M}'_{\gamma,2}$	CBC	50	Proj.-dep.	2.2×10^5	1.807 ± 0.007	0.492
Down-and-in option (PCA), $s = 6$						
$\mathscr{P}_{\gamma,4}$	CBC	50	Geometric	7.8	1.180 ± 0.003	0.238
			Product	7.5	1.212 ± 0.004	0.332
			Proj.-dep.	7.5	1.169 ± 0.004	0.267
			Order-dep.	7.1	1.149 ± 0.005	0.372
			Order 2	4.0	1.160 ± 0.010	0.793
		–	Proj.-dep.	9.0	1.193 ± 0.009	0.227
	Korobov	50	Proj.-dep.	7.1	1.195 ± 0.005	0.341
		–	Proj.-dep.	7.6	1.181 ± 0.008	0.217
$\mathscr{M}_{\gamma,2}$	CBC	50	Proj.-dep.	6.0	1.160 ± 0.004	0.313
$\mathscr{M}'_{\gamma,2}$	CBC	50	Proj.-dep.	6.2	1.183 ± 0.007	0.500

of order 6 and it accounts for 9.4% and 13% of the total variance for Cholesky and PCA, respectively.

The Asian option payoff function is continuous with respect with to the uniforms, but the down-and-in variant is not, so we use the baker's transformation for the former but not for the latter. For the PCA case, we show in Table 1 the fitted VRF's and empirical convergence rates for various types of weights with the $\mathscr{P}_{\gamma,2\alpha}$ criterion using random CBC construction, and for projection-dependent weights for the $\mathscr{M}_{\gamma,2}$ and $\mathscr{M}'_{\gamma,2}$ criteria. The error on $\ln \hat{a}_0$ (not shown in the table) is in general of the order of one tenth of the value of $\hat{S}_\varepsilon$ or less. Besides constant order-truncated weights at order 2, which yield poor performance as confirmed in Fig. 3, the other types of weights all seem to offer comparable performance. With PCA, compared to Cholesky, the VRF's are much higher, the convergence with n is faster, and there is less noise in the observed variances (see the appendix).

We compared the relative performance of the criteria $\mathscr{M}_{\gamma,\beta}$, $\mathscr{M}'_{\gamma,\beta}$, $\widetilde{\mathscr{M}}_{\gamma,\beta}$ and $\widetilde{\mathscr{M}}'_{\gamma,\beta}$, for $\beta = 1$ and 2, and for projection-dependent, product, order-dependent and geometric order-dependent weights. With Cholesky factorization, $\widetilde{\mathscr{M}}_{\gamma,\beta}$ and

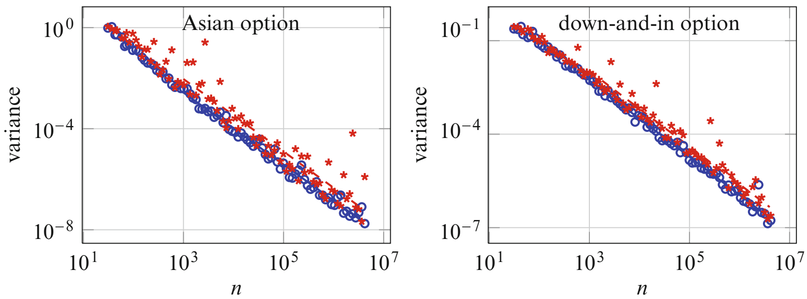


Fig. 3 Estimated and fitted variance of the RQMC estimator, using lattices constructed with the $\mathscr{P}_{\gamma,2\alpha}$ criterion, for the Asian option with $\alpha = 2$ and the baker's transformation (*left*) and for the down-and-in option with $\alpha = 1$ without the baker's transformation (*right*), using Cholesky factorization, with projection-dependent weights (○) and with constant order-dependent weights truncated at order 2 (∗).

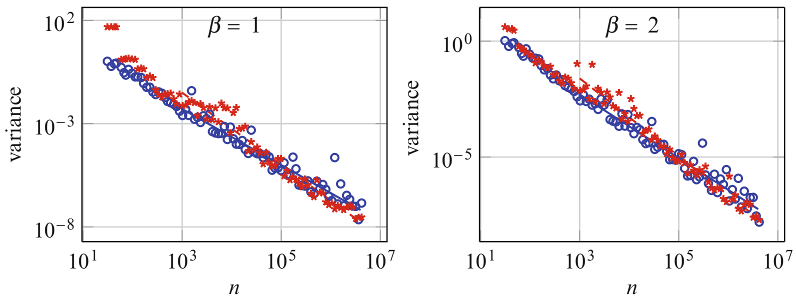


Fig. 4 Estimated and fitted variance of the RQMC estimator for the Asian option (Cholesky) with the baker transformation, using lattices constructed with the $\mathcal{M}'_{\gamma,\beta}$ (○) and $\widetilde{\mathcal{M}}'_{\gamma,\beta}$ (∗) criteria, with $\beta = 1$ (*left*) and $\beta = 2$ (*right*) and with product weights.

$\widetilde{\mathcal{M}}'_{\gamma,\beta}$, based on the worst projection, generally yield faster convergence and larger VRF than their counterparts $\mathcal{M}_{\gamma,\beta}$ and $\mathcal{M}'_{\gamma,\beta}$ based on a weighted sum over all projections. Besides this, it is hard to discriminate between criteria and weight types. We illustrate part of these observations in Fig. 4, where we compare (14)–(16) for $\beta = 1$ and 2 and product weights. The observed variances are more noisy on average when using (16), but the convergence seems faster. When using PCA, on the other hand, we did not observe any significant difference in the results across different criteria. The easy explanation is that for integrands where only a small part of the variance lies in projections of order 2 or more, all criteria and weight types under consideration here are practically equivalent in terms of the variance of the RQMC estimator.

In Table 1, we also give some results for exhaustive CBC and Korobov constructions for projection-dependent weights, for the Asian option. Random Korobov means that we tried r random values of a. The exhaustive CBC construction generally provides a slightly better variance reduction than random CBC, and the

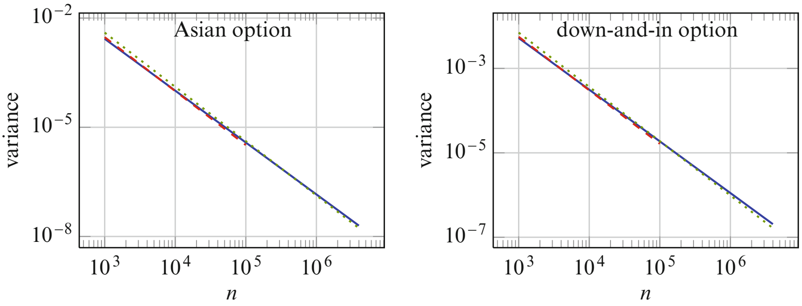


Fig. 5 Estimated and fitted variance of the RQMC estimator, using lattices constructed with the $\mathscr{P}_{\gamma,2\alpha}$ criterion and projection-dependent weights, for the Asian option with $\alpha = 2$ and the baker's transformation (*left*) and for the down-and-in option with $\alpha = 1$ without the baker's transformation (*right*), using Cholesky factorization, with random CBC construction with $r = 50$ (——), exhaustive CBC construction (— — —) or exhaustive Korobov construction (.......) .

Korobov construction is slightly worse than CBC, but the difference is thin, as can be seen in Fig. 5. Note that because the cost of exhaustive CBC increases with n (there are $(s-1)(n-1)$ vectors to examine) we have results only for $n \leq 10^5$ in this case.

We also constructed lattices using values of n that are powers of 2. In some cases, they clearly produced larger RQMC variances than lattices with prime n, as illustrated in Fig. 6. But in most cases, the variances for n prime or a power-of-two are comparable. For instance, this occurs for the example of Fig. 6, but with product weights instead of projection-dependent weights. Note that in order to have each a_j relatively prime with n for $j = 2, \ldots, s$ when n is a power of 2, a_j has to be an odd number, which means that for each component of the generating vector $\mathbf{a}$ except the first which is fixed at $a_1 = 1$, there is only half the number of possible values to consider. In other words, the space of lattice parameters is 2^{s-1} times smaller for values of n that are powers of 2 than for prime values of n. This could be part of the explanation.

We also did a few experiments with the $\widetilde{\mathscr{M}}_{\gamma,1}$ criterion as in (15), with $\mathscr{J} = \mathscr{J}(32, 24, 16, 12)$, as proposed in [15]. As shown in Fig. 7, this criterion does not perform well. It does not appear appropriate for the problems at hand, because too many coordinates (up to 32) are considered by the criterion whereas projections of order 5 and 6 are ignored.

Finally, to show a situation where projection-dependent weights perform clearly better than other types of weights, we give some results for an (artificial) example where we have two independent Asian options, each with $s = 3$ and the same parameters, and the payoff is the sum of the payoffs of the two options. Of course, we could estimate the expected payoff of each of the two options by RQMC separately and add up, but here, for the purpose of the illustration, we simulate the first option using the first three coordinates of the six-dimensional point set and the second option, using the last three coordinates. Then, the ANOVA variances are

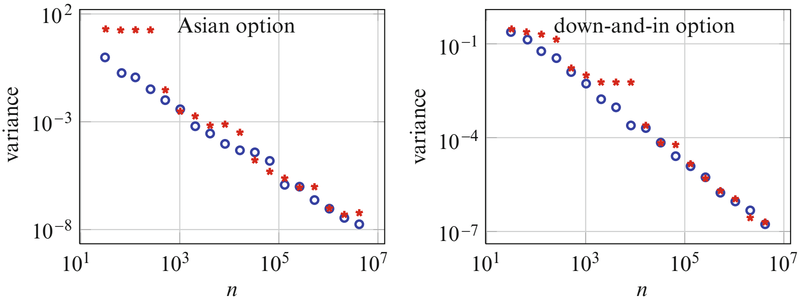


Fig. 6 Estimated and fitted variance of the RQMC estimator for lattices constructed with the $\mathcal{P}_{\gamma,2\alpha}$ criterion with projection-dependent weights, for the Asian option with the baker's transformation and $\alpha = 2$ (*left*) and the down-and-in option without the baker's transformation with $\alpha = 1$ (*right*), for with prime values of n ($\circ$) and for values of n that are powers of 2 ($*$).

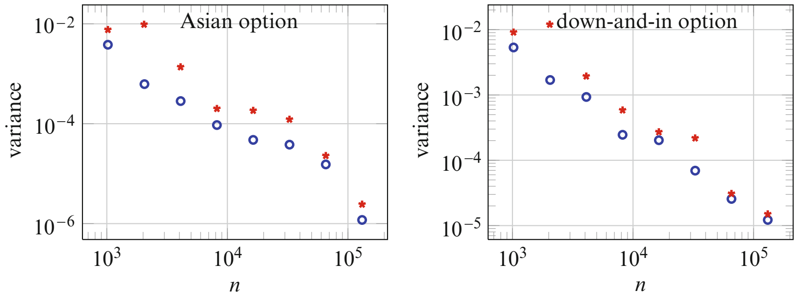


Fig. 7 Fitted variance of the RQMC estimator for the Asian option with the baker's transformation (*left*) and the down-and-in option without the baker's transformation (*right*) with Cholesky factorization, for lattices constructed using the $\mathcal{P}_{\gamma,2\alpha}$ criterion with random CBC with $r = 50$ ($\circ$) and the $\widetilde{\mathcal{M}}_{\gamma,1}$ criterion as in (15) with $\mathcal{J} = \mathcal{J}(32, 24, 16, 12)$ ($*$) and Korobov construction.

non-zero only for projections $\mathfrak{u}$ such that $\emptyset \neq \mathfrak{u} \subseteq \{1, 2, 3\}$ or $\emptyset \neq \mathfrak{u} \subseteq \{4, 5, 6\}$. There are thus only 14 out of 63 total projections that are relevant to the problem. This way, order-dependent weights are unlikely to perform well, because they give significant weight to the 9 irrelevant projections of order 2 and to the 18 irrelevant projections of order 3, rather than concentrate the weights over the important projections. We expect product weights to do even worse, because they waste their weights to these and on the 22 other irrelevant projections of order 4–6. This is confirmed in Fig. 8 and Table 2. Interestingly, the lattices obtained using $\alpha = 1$ appear more robust than those with $\alpha = 2$, even if the baker's transformation was used in both cases.

In summary, other choices of weights frequently perform almost as well as (more general) projection-dependent weights even when they are not really justified, which is good news, but there are situations where the projection-dependent weights really perform much better.

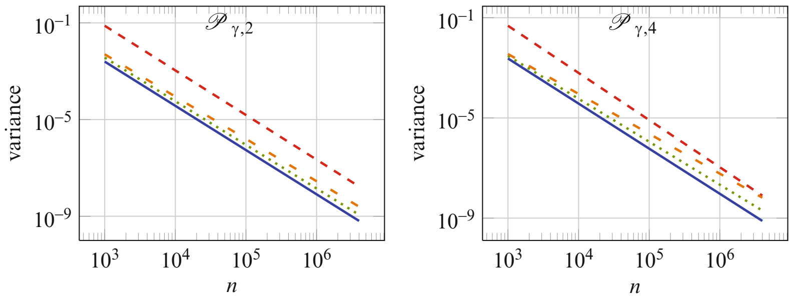


Fig. 8 Fitted variance of the RQMC estimator for the sum of 2 Asian payoffs, with Cholesky factorization, using the baker's transformation and criterion $\mathcal{P}_{\gamma,2\alpha}$ with $\alpha = 1$ (*left*) and $\alpha = 2$ (*right*), using projection-dependent weights (——), product weights (— — —), order-dependent weights (.......), and geometric weights (_ _).

Table 2 Estimated $\widehat{\mathrm{VRF}}$, $\hat{\nu}$ and $\hat{S}_\varepsilon$ for the RQMC estimator of the sum of 2 Asian options for the criterion $\mathcal{P}_{\gamma,2\alpha}$ with $\alpha = 1$ and 2 with the baker's transformation in both cases.

Weight type	$\mathcal{P}_{\gamma,2}$			$\mathcal{P}_{\gamma,4}$		
	$\widehat{\mathrm{VRF}}(2^{20})$	$\hat{\nu}$	$\hat{S}_\varepsilon$	$\widehat{\mathrm{VRF}}(2^{20})$	$\hat{\nu}$	$\hat{S}_\varepsilon$
Proj.-dep.	1.9×10^5	1.829 ± 0.005	0.351	1.7×10^5	1.800 ± 0.004	0.328
Product	7.2×10^3	1.85 ± 0.03	1.88	1.5×10^4	1.88 ± 0.02	1.35
Order-dep.	1.1×10^5	1.80 ± 0.01	0.669	7.2×10^4	1.72 ± 0.01	0.738
Geometric	5.6×10^4	1.75 ± 0.01	1.10	2.6×10^4	1.59 ± 0.01	1.00

12 Conclusion

The optimal lattice, which minimizes the variance when estimating an integral by a randomly shifted lattice rule, depends on the integrand f, and optimizing this lattice is harder in general than computing the integral itself. The idea of constructing efficient adaptive algorithms by estimating the Fourier coefficients or the variance components, for general applications, is attractive at first sight, but estimating those quantities with reasonable accuracy usually too costly. Fortunately, crude estimates of the variance components are generally sufficient to identify the subsets of coordinates on which to put more weight when constructing the lattice, and doing this with a weighted $\mathcal{P}_{\gamma,2\alpha}$ figure of merit with projection-dependent weights is a robust approach that gives very good results in most examples that we have tried. In fact, lattices constructed based on a weighted $\mathcal{P}_{\gamma,2\alpha}$ with reasonable choices of weights, such as order-dependent weights that decrease geometrically (but not too fast) with the cardinality of coordinate subsets, perform well enough in most cases. Such lattices could be provided in general-purpose RQMC software. On the other hand, lattices constructed with lousy choices of weights, that give too little weight to some important projections (for example, giving weight only to the projections of order 2), or too much weight to several irrelevant projections, often perform poorly. We also saw counterexamples (indicator functions in two dimensions) where a lattice having the best $\mathscr{P}_{\gamma,2\alpha}$ performs very poorly, not because of a poor choice of weights, but because $\mathscr{P}_{2\alpha}$ is not always a relevant measure in these examples. Thus, all the practical methods that we can propose to define a figure of merit for general applications are heuristic and none is foolproof. However, these counterexamples were constructed on purpose and such cases are rarely encountered in applications.

The theoretical asymptotic convergence rate of $\mathscr{O}(n^{-2\alpha+\delta})$ for $\mathscr{P}_{\gamma,2\alpha}$ and for the RQMC variance for certain classes of smooth functions is rarely observed in the practical range of values of n, say up to a few millions. The rates we have observed empirically, with the best lattices we found, are typically somewhere between $\mathscr{O}(n^{-2})$ and $\mathscr{O}(n^{-1})$. Interestingly, this applies not only to smooth functions f, but also to non-smooth integrands, and even to discontinuous and unbounded ones.

An ongoing project related to this study is to build integrated software tools that can construct lattices based on a variety of parameterized figures of merit, with flexibility for the choices of weights (or parameters), and feed them to simulation software for arbitrary RQMC applications. This will include lattices extensible in both the dimension s and the number of points n. Hopefully, this will put these RQMC methods closer to the hands of practitioners and promote their utilization in a large variety of applications.

The online appendix to this paper can be found at http://www.iro.umontreal.ca/~lecuyer/myftp/papers/mcqmc-plenary-app.pdf.

Acknowledgements This research has been supported by NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to the first author. Computations were performed using the facilities of the Réseau québécois de calcul haute performance (RQCHP).

References

1. Avramidis, A.N., Wilson, J.R.: Integrated variance reduction strategies for simulation. Operations Research **44**, 327–346 (1996)
2. Avramidis, A.N., Wilson, J.R.: Correlation-induction techniques for estimating quantiles in simulation experiments. Operations Research **46**(4), 574–591 (1998)
3. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups, 3rd edn. Grundlehren der Mathematischen Wissenschaften 290. Springer-Verlag, New York (1999)
4. Cools, R., Nuyens, D.: A Belgian view on lattice rules. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 3–21. Springer-Verlag, Berlin (2008)
5. Cranley, R., Patterson, T.N.L.: Randomization of number theoretic methods for multiple integration. SIAM Journal on Numerical Analysis **13**(6), 904–914 (1976)
6. Dick, J., Sloan, I.H., Wang, X., Wozniakowski, H.: Liberating the weights. Journal of Complexity **20**(5), 593–623 (2004)
7. Dick, J., Sloan, I.H., Wang, X., Wozniakowski, H.: Good lattice rules in weighted Korobov spaces with general weights. Numerische Mathematik **103**, 63–97 (2006)
8. Efron, B., Stein, C.: The jackknife estimator of variance. Annals of Statistics **9**, 586–596 (1981)
9. Elmaghraby, S.: Activity Networks. Wiley, New York (1977)

10. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Mathematics of Computation **67**(221), 299–322 (1998)
11. Hickernell, F.J.: Lattice rules: How well do they measure up? In: P. Hellekalek, G. Larcher (eds.) Random and Quasi-Random Point Sets, *Lecture Notes in Statistics*, vol. 138, pp. 109–166. Springer-Verlag, New York (1998)
12. Hickernell, F.J.: Obtaining $O(N^{-2+\varepsilon})$ 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)
13. L'Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. Operations Research **47**(1), 159–164 (1999)
14. L'Ecuyer, P.: Quasi-Monte Carlo methods with applications in finance. Finance and Stochastics **13**(3), 307–349 (2009)
15. L'Ecuyer, P., Lemieux, C.: Variance reduction via lattice rules. Management Science **46**(9), 1214–1235 (2000)
16. L'Ecuyer, P., Lemieux, C.: Recent advances in randomized quasi-Monte Carlo methods. In: M. Dror, P. L'Ecuyer, F. Szidarovszky (eds.) Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications, pp. 419–474. Kluwer Academic, Boston (2002)
17. L'Ecuyer, P., Munger, D., Tuffin, B.: On the distribution of integration error by randomly-shifted lattice rules. Electronic Journal of Statistics **4**, 950–993 (2010)
18. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer-Verlag, New York, NY (2009)
19. 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)
20. Owen, A.B.: Latin supercube sampling for very high-dimensional simulations. ACM Transactions on Modeling and Computer Simulation **8**(1), 71–102 (1998)
21. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Clarendon Press, Oxford (1994)
22. 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)
23. Sloan, I.H., Rezstov, A.: Component-by-component construction of good lattice rules. Mathematics of Computation **71**, 262–273 (2002)
24. Sloan, I.H., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals. Journal of Complexity **14**, 1–33 (1998)
25. Sobol', I.M., Myshetskaya, E.E.: Monte Carlo estimators for small sensitivity indices. Monte Carlo Methods and Applications **13**(5–6), 455–465 (2007)
26. Wang, X.: Constructing robust good lattice rules for computational finance. SIAM Journal on Scientific Computing **29**(2), 598–621 (2007)

A Study of the Efficiency of Exact Methods for Diffusion Simulation

Stefano Peluchetti, and Gareth O. Roberts

Abstract In this paper we investigate the efficiency of some simulation schemes for the numerical solution of uni- and multi-dimensional stochastic differential equation (SDE) with particular interest in a recently developed technique for diffusion simulation [5] which avoids the need for any time-discretisation approximation (the so-called *exact algorithm* for diffusion simulation). The schemes considered are: the Exact Algorithm, the Euler, the Predictor-Corrector and the Ozaki-Shoji schemes. The analysis is carried out via a simulation study using some test SDEs. We also consider efficiency issues arising by the extension of EA to the multi-dimensional setting.

1 Introduction

A general methodology for the simulation of uni- and multi-dimensional diffusions was recently introduced in a series of papers [5–7]. The unique feature of these methods is that they permit the simulation of analytically intractable diffusions without recourse to time-discretisation. The methods can therefore claim to be *exact* (at least up to the precision limits of the computer used to perform the simulation) and we shall call the general method the *exact algorithm* (EA).

Numerical schemes for the simulation of diffusion processes have been around for some time, the first contribution probably being that of [13]. Theoretical work has largely focused on strong and weak approximation results, see for example [2,9,

S. Peluchetti
HSBC, 8 Canada Square, London, E14 5HQ, UK
e-mail: phd.st.p@gmail.com

G.O. Roberts
Department of Statistics, University of Warwick, Coventry, CV4 7AL, UK
e-mail: gareth.o.roberts@warwick.ac.uk

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_7, © Springer-Verlag Berlin Heidelberg 2012

12] and the references therein. However before the work of [7], exact simulation was confined to a very small class of diffusion processes whose stochastic differential equations yielded explicit solutions (see for example [4]).

It is natural to expect there to be a price to pay for the exactness, and this is the question we address here. Thus the purpose of this paper is to consider the efficiency of EA, and to compare it with other off-the-shelf methods for diffusion simulation. In our experiments we find that in most cases, EA methods are at least as computationally efficient as the Euler, Predictor Corrector or Ozaki-Shoji schemes. Thus, surprisingly, there appears to be no price to pay for exactness, at least in general.

Some preliminary results on the efficiency of EA, in a Monte Carlo scenario, can be found in [8]. However this paper gives a substantially more extensive investigation of EA. We initially consider a class of test models that synthesise a range of one-dimensional diffusive dynamics that are encountered in a range of applications. We thus simulate them using three well known discretisation schemes and EA and we compare the results obtained. We also give a detailed simulation study of the efficiency of EA in multi-dimensional settings.

1.1 Broad Conclusions, Comparisons, Restrictions and the Value of Exactness

Of course any comparison between methods depends on the diffusion functional expectation we wish to estimate. Here we focus on finite-dimensional distributions and sample path maxima. The comparison also depends on the level of accuracy required and/or the available computing resource. Therefore any comparison between methods has to lead to cautious conclusions. We have however tried to consider a range of examples which focus on known efficiency properties (both weak and strong) of EA methods.

It is well-known that the computational complexity of approximating a diffusion functional moment to an accuracy (root mean square error) of ϵ is typically ϵ^{-3} for routine application of (say) the Euler-Maruyama scheme on a diffusion with suitable smooth (e.g. Lipschitz) coefficients. Multi-scale methods can improve this complexity to $(\log \epsilon)^2 \epsilon^{-2}$ [1]. However it is intrinsically impossible to improve this complexity to ϵ^{-2} using discretisation methods.

On the other hand, calculations for EA methods are much more straightforward since there is no requirement to trade off approximation bias with computational time. Given computing resource T, the weak error of estimating any functional with finite variance is just $O(T^{-1/2})$ by the central limit theorem Thus the computational complexity of approximating to accuracy ϵ is automatically ϵ^{-2}. This means that EA methods will always outperform any discretisation methods for all sufficiently highly prescribed precision.

In this paper, the comparisons we make do not involve very high accuracy requirements. We consider instead the non-asymptotic regime where moderate accuracy is required or limited computing resource is available.

It is not always possible to apply EA methods. Whilst the literature strong and weak approximation results for discretisation schemes often uses Lipschitz (or somewhat weaker) conditions on diffusion and drift functions, current EA methods require both to be C^1 functions. Furthermore, for multi-dimensional diffusions, a major restriction is that the diffusions must be transformable to unit-diffusion coefficient diffusions, and then must have drifts given as potential gradients. Whilst the differentiability conditions can (and are being) weakened, the unit-diffusion, gradient drift condition is fundamentally essential for EA methods.

Even where EA methods turn out to be inefficient (rare in our comparison) one important role for EA methods is to be able to quantify approximation errors for faster discretisation methods.

We do not claim that the numerical comparison we carry out is exhaustive in any sense. Clearly that would not be possible. In particular, we have chosen only three from the vast array of possible discretisation methods; we have only considered a relatively small collection of diffusions to simulate; and our comparison criteria are necessarily somewhat arbitrary. However, we do expect that the broad conclusions borne from this study will apply much more generally, and this is supported by our numerical experience in other examples of the use of EA and discretisation methods.

1.2 The Structure of the Paper

This paper is organised as follows. In Sect. 2, EA and the three discretisation schemes are briefly introduced. Section 3 consists of the simulation study where the efficiency of the four schemes is studied. The main difficulty is comparing a scheme that returns the exact result with schemes that return approximated results. Consequently it is necessary to introduce a comparison criterion that measures a "distance" between the true and the approximated result. We are interested in both the sensitivity of the schemes to the parameters of the test SDEs and the ratio of efficiency between EA and the other schemes. In Sect. 4 the efficiency of the multi-dimensional extension of EA is investigated, without any comparison with the other discretisation schemes. Section 5 concludes the paper.

2 The Simulation Schemes

2.1 The Exact Algorithm

We begin considering a generic one-dimensional and time homogeneous Stochastic Differential Equation (SDE)

$$\begin{aligned} dY_t &= b\left(Y_t\right) dt + \sigma\left(Y_t\right) dB_t \qquad\qquad 0 \le t \le T \\ Y_0 &= y \end{aligned} \tag{1}$$

where B is the scalar Brownian Motion (BM) and y is the initial condition. The drift coefficient b and the diffusion coefficient σ are assumed to satisfy the proper conditions for the existence and uniqueness of a strong solution of (1). Let Y be the diffusion process strong solution of (1).

Under the additional requirement that σ is continuously differentiable and strictly positive let

$$\eta(u) := \int^{u} \sigma^{-1}(z)\,dz \tag{2}$$

be the anti-derivative of σ^{-1}. It follows that $X_t := \eta(Y_t)$ satisfies the unit diffusion coefficient SDE

$$\begin{aligned} dX_t &= \alpha(X_t)\,dt + dB_t \qquad\qquad 0 \le t \le T \\ X_0 &= x := \eta(y) \end{aligned} \tag{3}$$

with drift coefficient

$$\alpha(u) := \frac{b\left\{\eta^{-1}(u)\right\}}{\sigma\left\{\eta^{-1}(u)\right\}} - \frac{\sigma'\left\{\eta^{-1}(u)\right\}}{2} \tag{4}$$

SDE (3) is assumed to admit a unique strong solution and we denote by $\mathbb{X}$ the state space of X. The map (2), also known as the Lamperti transform, allows us to consider the simpler problem of simulating from (3) for a vast class of one-dimensional SDEs.

In what follows the laws of stochastic processes are defined on the measurable space of continuous functions $C([0,T],\mathbb{R})$ with its cylinder sigma algebra $\mathscr{C}([0,T],\mathbb{R})$, or on the obvious restrictions of this space. Let $\mathbb{Q}_T^x$ and $\mathbb{W}_T^x$ denote the law of the diffusion X and the law of a BM respectively on $[0,T]$ both started at x. From now on the following hypotheses are assumed to hold

- $(C1)$ $\forall x \in \mathbb{X}$ $\mathbb{Q}_T^x \ll \mathbb{W}_T^x$ and the Radon-Nikodym derivative is given by Girsanov's formula

$$\frac{d\mathbb{Q}_T^x}{d\mathbb{W}_T^x}(\omega) = \exp\left\{\int_0^T \alpha(\omega_s)\,dX_s - \frac{1}{2}\int_0^T \alpha^2(\omega_s)\,ds\right\} \tag{5}$$

  where $\omega \in C([0,T],\mathbb{X})$
- $(C2)$ $\alpha \in C^1(\mathbb{X},\mathbb{R})$;
- $(C3)$ $\alpha^2 + \alpha'$ is bounded below on $\mathbb{X}$.

An application of Ito's formula to the function $A(u) = \int_0^u \alpha(z)\,dz$ results in a more tractable form of (5)

$$\frac{d\mathbb{Q}_T^x}{d\mathbb{W}_T^x}(\omega) = \exp\{A(\omega_T) - A(x)\}\exp\left\{-\int_0^T \frac{\alpha^2 + \alpha'}{2}(\omega_s)\,ds\right\} \tag{6}$$

Under the integrability assumption

- $(C4)\ \forall x \in \mathbb{X}\ \eta_{x,T} := \mathbb{E}_{\mathbb{W}_T^x}\left[e^{A(\omega_T)}\right] < \infty$

it is possible to get rid of the (possibly unbounded) term $A(\omega_T)$ of (6) introducing a new process Z with law $\mathbb{Z}_T^x$ by the Radon-Nikodym derivative

$$\frac{d\mathbb{Z}_T^x}{d\mathbb{W}_T^x}(\omega) = e^{A(\omega_T)}/\eta_{x,T} \tag{7}$$

$$\eta_{x,T} = \mathbb{E}_{\mathbb{W}_T^x}\left[e^{A(\omega_T)}\right] \tag{8}$$

We refer to Z as the Biased Brownian Motion (BBM). This process can be alternatively defined as a BM with initial value x conditioned on having its terminal value Z_T distributed according to the density

$$h_{x,T}(u) := \eta_{x,T} \times \exp\left\{A(u) - \frac{(u-x)^2}{2T}\right\} \tag{9}$$

It follows that

$$\frac{d\mathbb{Q}_T^x}{d\mathbb{Z}_T^x}(\omega) = \eta_{x,T}\exp\{-A(x)\}\exp\left\{-\int_0^T \frac{\alpha^2+\alpha'}{2}(\omega_s)\,ds\right\} \tag{10}$$

$$\propto \exp\left\{-\int_0^T \phi(\omega_s)\,ds\right\} \leq 1 \tag{11}$$

where $\phi(u) := \left(\alpha^2(u) + \alpha'(u)\right)/2 - l$ and $l := \inf_{r\in\mathbb{X}}\left(\alpha^2(r) + \alpha'(r)\right)/2 < \infty$. Equation 11 suggests the use of a rejection sampling algorithm to generate realisations from $\mathbb{Q}_T^x$. However it is not possible to generate a sample from Z, being Z an infinite-dimensional variate, and moreover it is not possible to compute analytically the value of the integral in (11).

Let $\mathbb{L}$ denote the law of a unit rate Poisson Point Process (PPP) on $[0,T]\times[0,\infty)$, and let $\Phi = \{\chi, \psi\}$ be distributed according to $\mathbb{L}$. We define the event Γ as

$$\Gamma := \bigcap_{j\geq 1} \phi\left(Z_{\chi_j}\right) \leq \psi_j \tag{12}$$

that is the event that all the Poisson points fall into the epigraph of $s \mapsto \phi(Z_s)$. The following theorem is proven in [6]

Theorem 1 (Wiener-Poisson factorisation). *If* $(Z,\Phi) \sim \mathbb{Z}_T^x \otimes \mathbb{L} \mid \Gamma$ *then* $Z \sim \mathbb{Q}_T^x$

At this stage the result is a purely theoretical, as it is not possible to simulate from the law $\mathbb{L}$. However, in the specific case of ϕ bounded above by $m < \infty$ it suffices to consider Φ as a PPP on $[0,T] \times [0,m]$. The reason is that for the determination of the event Γ, only the points of Φ below m matter. The algorithm resulting from this restrictive boundedness condition on ϕ is called EA1.

The hypothesis of bounded ϕ can be weakened or even removed, successively generalised and leading to EA2 [5] and to EA3 [6] respectively. Both extensions involves the simulation of some functional of Z or of an event depending on Z which restrict the range of Z, and by continuity the range of $\phi(Z)$.

EA1 and EA3 are used in the simulation study. Details of EA3 are described in the appendix. We also give there some important implementational details which are relevant to both EA1 and EA3.

2.2 The Discretisation Schemes

We now briefly introduce the three discretisation schemes (DS) whose efficiency, with that of EA, is investigated in the simulation study. All the DSs are assumed to have an equi-spaced discretisation interval of length $\Delta = T/n$, where n is the number of steps and Y^{Δ} denotes a corresponding generic DS. In the following $i = 1, \cdots, n$ and $Y_0 = x$ implicitly.

The Euler scheme is the simplest DS that can be used to approximate the solution of (1). It can be defined by the recursion

$$W_{\Delta}^{i} \overset{iid}{\sim} \mathscr{N}(0,\Delta) \tag{13}$$

$$Y_{i\Delta} = Y_{(i-1)\Delta} + b\left(Y_{(i-1)\Delta}\right)\Delta + \sigma\left(Y_{(i-1)\Delta}\right)W_{\Delta}^{i} \tag{14}$$

The Predictor-Corrector scheme is defined by

$$W_{\Delta}^{i} \overset{iid}{\sim} \mathscr{N}(0,\Delta) \tag{15}$$

$$\overline{Y}_{i\Delta} = Y_{(i-1)\Delta} + b\left(Y_{(i-1)\Delta}\right)\Delta + \sigma\left(Y_{(i-1)\Delta}\right)W_{\Delta}^{i} \tag{16}$$

$$Y_{i\Delta} = Y_{(i-1)\Delta} + \frac{1}{2}\left\{b\left(Y_{(i-1)\Delta}\right) + b\left(\overline{Y}_{i\Delta}\right)\right\}\Delta + \sigma\left(Y_{(i-1)\Delta}\right)W_{\Delta}^{i} \tag{17}$$

The idea behind this DS is to make a Euler prediction $\bar{Y}_{i\Delta}$ by using (16) and adjust $\bar{Y}_{i\Delta}$ by computing an average of the drift's value over the time step $((i-1)\Delta, i\Delta]$ using the trapezoid quadrature formula. This approach results in the correction (17). It is fundamental to_use the same W_{Δ}^{i} in (16) and (17). For more details about the Euler and the Predictor-Corrector schemes see [12].

Finally we introduce the Ozaki-Shoji scheme. This DS uses a completely different approach that is only applicable to diffusion process with constant diffusion coefficient and, without loss of generality, to (3). This DS belongs to the family of "linearisation schemes" which approximates the drift α of (3) by some sort of linear approximation. The specific version here presented it the one of [15]. The idea behind this DS is to approximate the behaviour of $\alpha(X_t)$ in a neighbourhood of X_t using Ito's Lemma

$$d\alpha(X_t) = \alpha'(X_t)\,dX_t + \frac{1}{2}\alpha''(X_t)\,dt \tag{18}$$

$$\alpha(X_{t+h}) \approx \alpha(X_t) + \alpha'(X_t)(X_{t+h} - X_t) + \frac{1}{2}\alpha''(X_t)\,h \tag{19}$$

The law of the Ozaki-Shoji scheme on the time interval $(0, \Delta]$ is given by the solution of the linear SDE

$$dX_t = \left\{\alpha(x) + \alpha'(x)(X_t - x) + \frac{1}{2}\alpha''(x)\,t\right\} dt + dB_t \tag{20}$$

i.e. an Ornstein-Uhlenbeck process. By the time-homogeneity this DS under time-discretisation Δ is termed X^{Δ} and defined by the iterative formulae

$$\tilde{W}^i_{\Delta} \overset{iid}{\sim} \mathcal{N}\left(0, \frac{\exp\left\{2\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)\Delta\right\} - 1}{2\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)}\right) \tag{21}$$

$$X^{\Delta}_{i\Delta} = X^{\Delta}_{(i-1)\Delta} + \frac{\alpha\left(X^{\Delta}_{(i-1)\Delta}\right)}{\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)}\left(\exp\left\{\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)\Delta\right\} - 1\right) \tag{22}$$

$$+ \frac{\alpha''\left(X^{\Delta}_{(i-1)\Delta}\right)}{2\left(\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)\right)^2}\left\{\exp\left\{\alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)\Delta\right\} - 1 - \alpha'\left(X^{\Delta}_{(i-1)\Delta}\right)\Delta\right\} + \tilde{W}^i_{\Delta} \tag{23}$$

3 Some Uni-dimensional Simulation Studies

A standard way to compare DSs is related to the concepts of weak and strong convergence.

X^{Δ} is said to be a strong approximation of (1) if $\exists \Delta^*, k, \mathscr{S} > 0 : \forall \Delta \leq \Delta^*$

$$\mathbb{E}\left|X_T - X^{\Delta}_T\right| \leq k\Delta^{\mathscr{S}} \tag{24}$$

where $\mathscr{S}$ is the rate of convergence. This strong convergence criterion basically states the L_1 convergence of the last simulated point X^{Δ}_T to X_T. As such, the rate $\mathscr{S}$ is an indicator of how well X^{Δ} approximates the paths of X (for a fixed ω).

However is important to remember that the leading order constant k depends on (1). Of course other more stringent path dependent comparisons could be considered. However our aim here is to show that even for this criterion, DS methods are often computationally less efficient than EA alternatives.

X^{Δ} is said to be a weak approximation of (1) if $\exists \Delta^{*}, k, \mathscr{W} > 0 : \forall \Delta \leq \Delta^{*}$, $g \in \mathscr{G}$

$$\left|\mathbb{E}\left[g\left(X_T\right)\right] - \mathbb{E}\left[g\left(X_T^{\Delta}\right)\right]\right| \leq k\Delta^{\mathscr{W}} \tag{25}$$

where $\mathscr{W}$ is the rate of weak convergence and $\mathscr{G}$ is a class of test functions. Here the rate $\mathscr{W}$ is an indicator of how accurately the distribution of X^{Δ} approximates the distribution of X. Hence this convergence criterion is more obviously relevant if we are interested in Monte Carlo simulations based on X^{Δ}. As in (24) the constant k of (25) depends on the SDE (1), limiting the practical appeal of these criteria. Our empirical results shows that DSs with the same $\mathscr{W}$ can perform very differently.

The framework of the simulation study is very simple: we consider a unit diffusion coefficient SDE X (3) and a functional F, possibly path-dependent, of interest. In this framework we compare the efficiency of EA and the three DSs previously introduced.

As EA does not clearly involves any discretisation error, its efficiency is inversely proportional to the average computational cost required to sample a single realisation of the functional $F(X)$.

For a given X^{Δ}, the smallest computational cost, i.e. the biggest Δ, required for $F\left(X^{\Delta}\right)$ to be an accurate approximation of $F(X)$ is then computed. More precisely, we are interested in how similar the distribution of $F\left(X^{\Delta}\right)$ is to the distribution of $F(X)$. Our test of choice is the two-sided two-sample Kolmogorov-Smirnov (KS) test. EA is used to sample $F(X)$ exactly. Let $\alpha \in (0,1)$ be a fixed threshold and Δ^{*} be the biggest value of Δ such that the p-value of the KS test of $\left\{F(X), F\left(X^{\Delta}\right)\right\}$ is higher then the threshold α. The efficiency of X^{Δ} is then defined as inversely proportional to the computational cost required for the simulation of a single realisation of the functional $F\left(X^{\Delta^{*}}\right)$.

To compute the KS test of $\left\{F(X), F\left(X^{\Delta}\right)\right\}$ we choose to sample $N \in \mathbb{N}$ skeletons from X using EA and N discretisation using X^{Δ}. For each one of these samples the value of the functional F is computed resulting in $2N$ samples: N exact and N approximated observations. Finally the p-value of the KS statistic calculated over these $2N$ samples. Moreover to decrease the variance of the KS test (that in this framework is just stochastic noise) we average its value over $M \in \mathbb{N}$ repetitions. All these simulations needs to be repeated until we find the right Δ^{*} for each of the three DSs considered in the comparison, i.e. the smallest Δ so that we accept the null hypothesis according to the KS test. Finally we repeat all these steps for a reasonable number of combinations of the parameters of the SDE, to obtain computational cost surfaces (as a function of the parameters) for EA and the DSs.

In our simulation study the following arbitrary values are considered: $\alpha = 0.05, N = 10^5, M = 10^3$. The choice of the KS test is arbitrary too, but there are a number of reasons why we opted for the this test. First of all, it has an intuitive meaning. More importantly, it is possible to obtain the limiting distribution of the KS statistic under the null hypothesis. Lastly we want to be cautious about our conclusions. The use of a more powerful goodness of fit test would pose questions about the robustness of our results to the choice of the test statistic considered. This would be especially true for tests that give more importance to the tails of the distribution, as preliminary examination of the histograms of the densities involved reveals that the biggest differences are usually in the tails.

The aim of this simulation study is to obtain useful indication about the efficiency of EA and the three DSs. The choice of the diffusion models that we take into account reflects this objective, they are "toy examples".

3.1 The Case of EA1

The class of parametric diffusion models that can be considered is limited by the assumptions of EA1. We focus on the following three models:

- The PSINE SDE

$$dX_t = \theta \sin(\gamma X_t)\, dt + dB_t \qquad \theta > 0, \gamma > 0 \tag{26}$$

- The NSINE SDE

$$dX_t = \theta \sin(\gamma X_t)\, dt + dB_t \qquad \theta < 0, \gamma > 0 \tag{27}$$

- The PTANH SDE

$$dX_t = \theta \tanh(\gamma X_t)\, dt + dB_t \qquad \theta > 0, \gamma > 0 \tag{28}$$

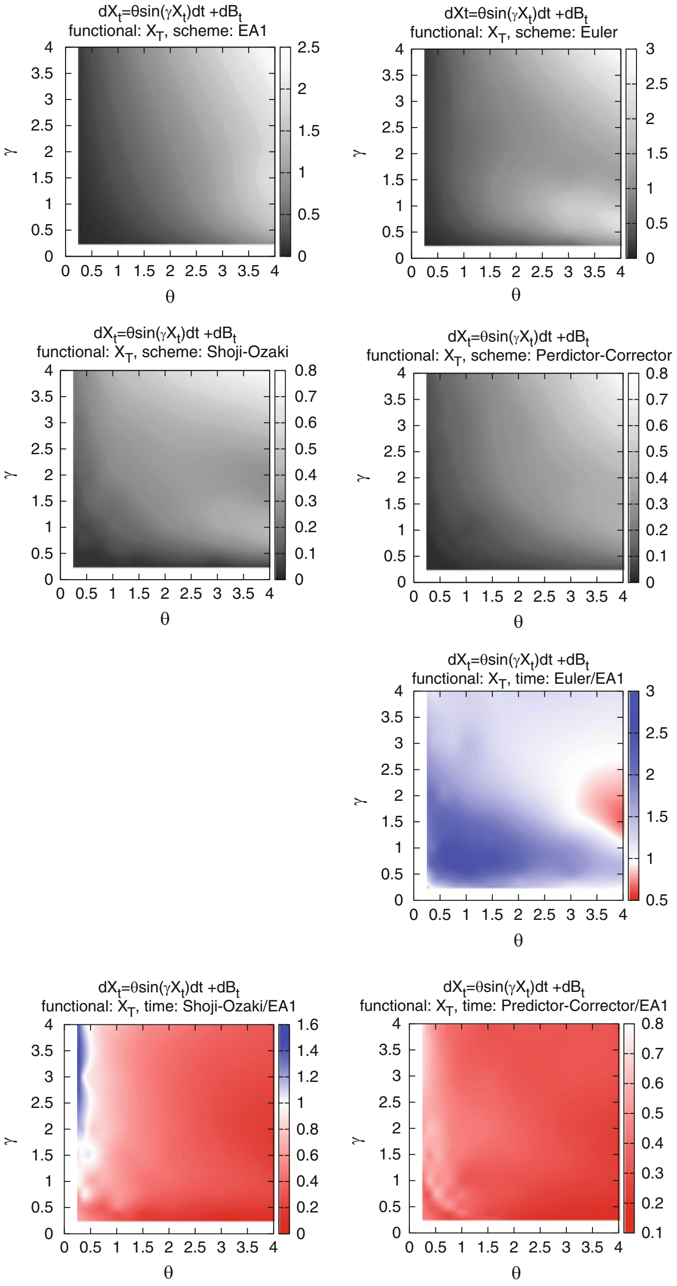
- The NTANH SDE

$$dX_t = \theta \tanh(\gamma X_t)\, dt + dB_t \qquad \theta < 0, \gamma > 0 \tag{29}$$

We take into account these models because they summarise a good range of diffusion dynamics. In every model the starting point x and the terminal time T are fixed to 0 and 1 respectively.

The functionals considered are the last point $L(X) = X_T$ and the maximum of the path $M(X) = \sup_{0 \leq s \leq T} X_s$. For $M(X)$ we simulate the maximum of a BB between each discretized value even when dealing with DSs.

In Figs. 1, 2, 3, 4, 5 and 6 the four plots on the top of each figure represents on the Z-axis the computational time required by EA and by the three DSs to complete the simulation (with the required level of accuracy) as function of the values of the SDE's parameters.



$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, scheme: EA1

$dXt=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, scheme: Euler

$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, scheme: Shoji-Ozaki

$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, scheme: Perdictor-Corrector

$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, time: Euler/EA1

$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, time: Shoji-Ozaki/EA1

$dX_t=\theta\sin(\gamma X_t)dt+dB_t$
functional: X_T, time: Predictor-Corrector/EA1

Fig. 1 Model: PSINE, functional: $L\,(X)$.

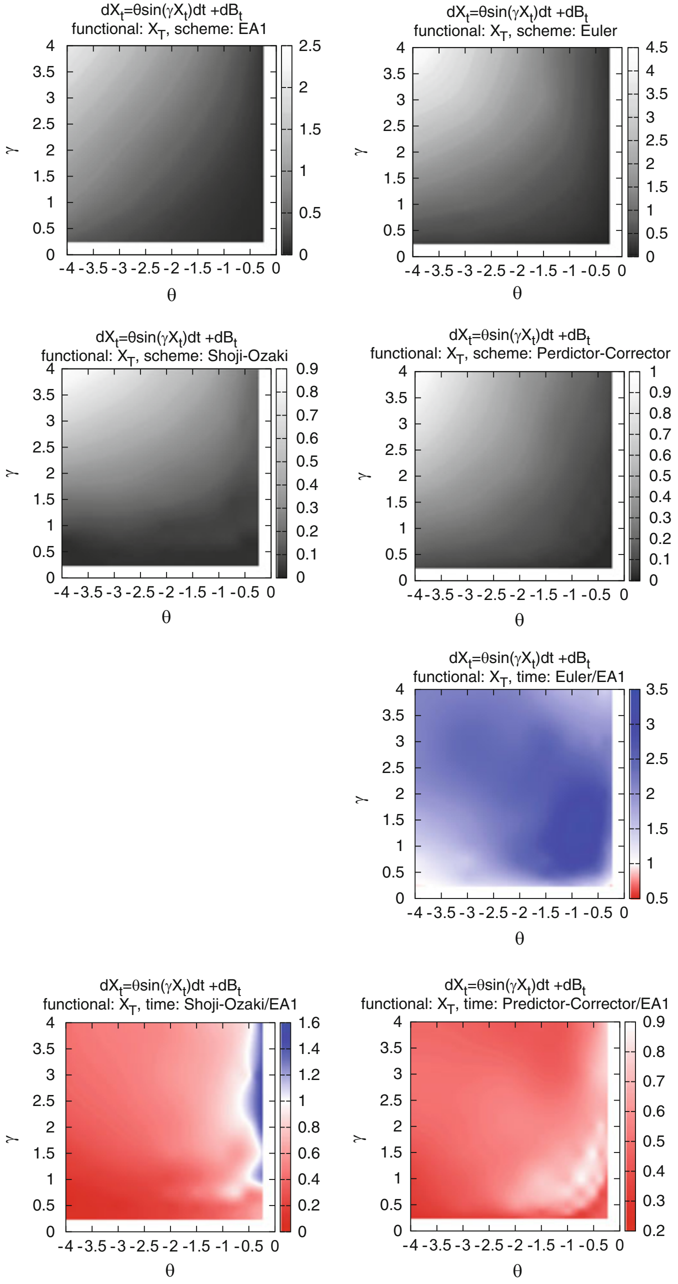


Fig. 2 Model: NSINE, functional: $L(X)$.

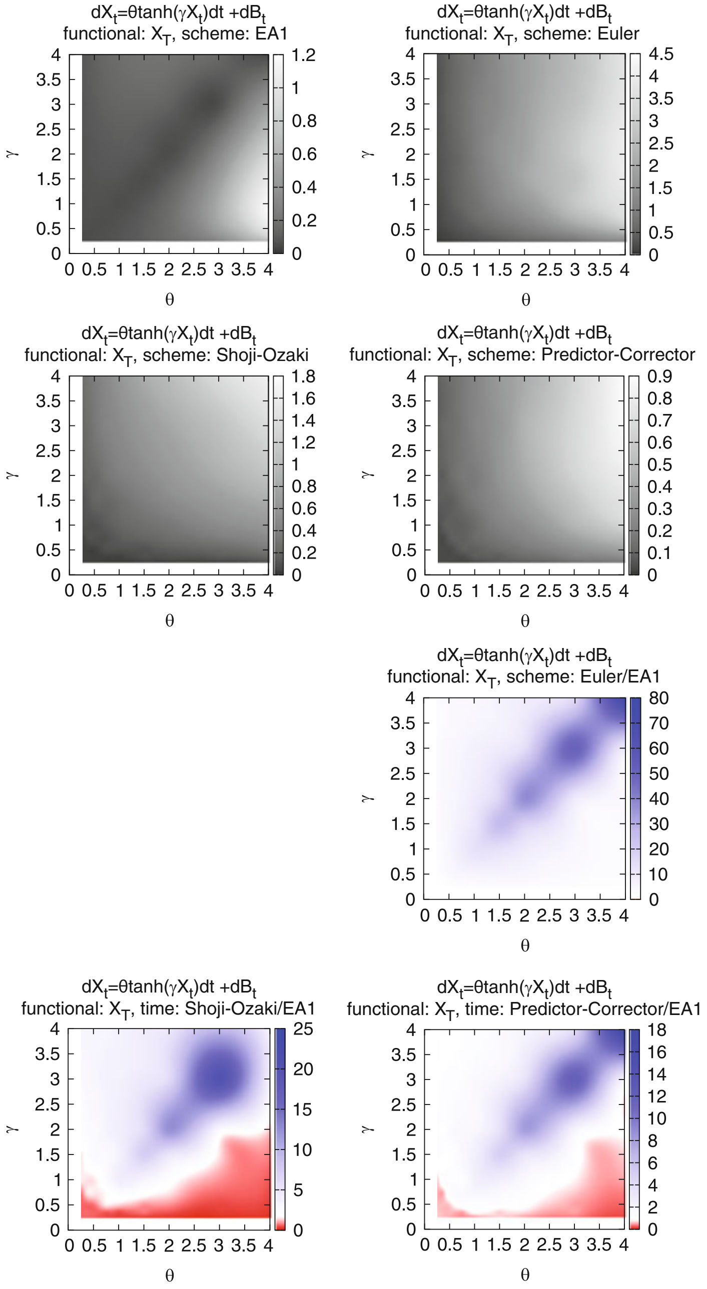


Fig. 3 Model: PTANH, functional $L(X)$. Ozaki-Shoji scheme does not converge if $-\theta = \gamma = 4$.

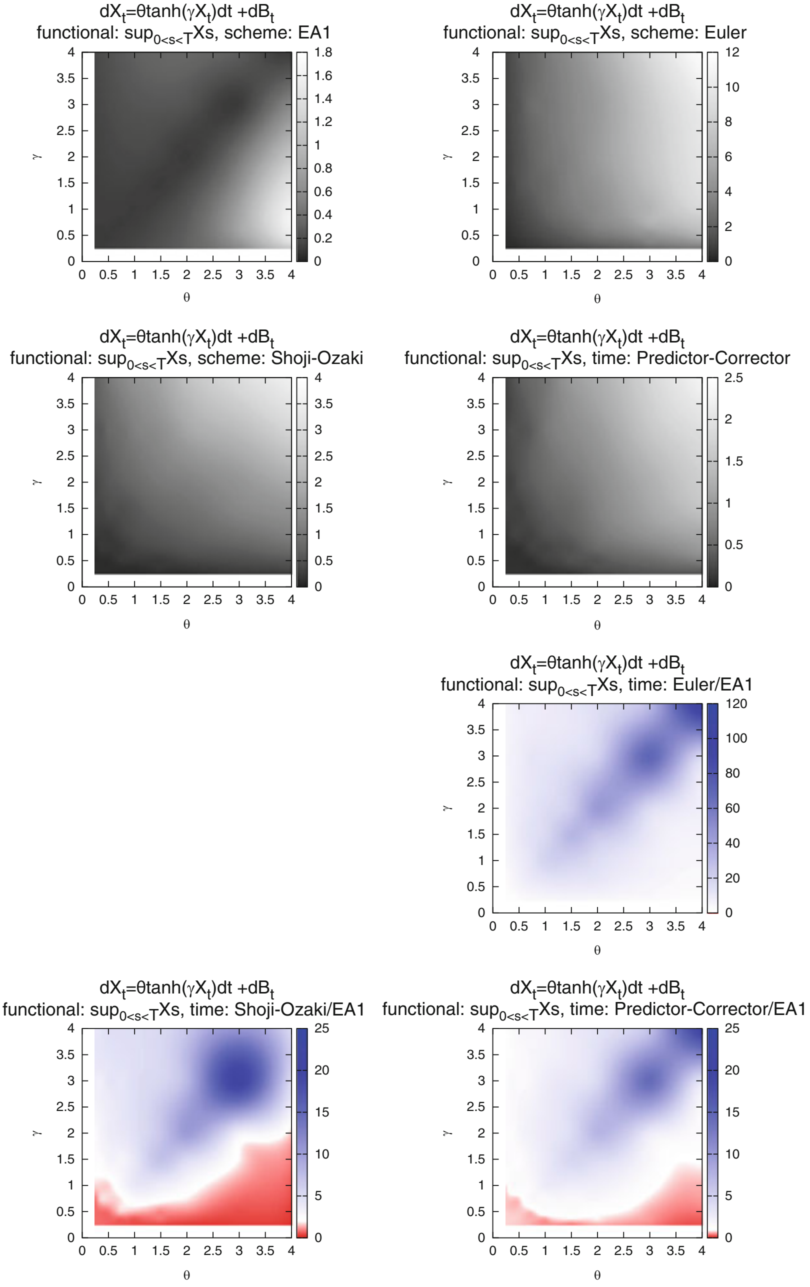
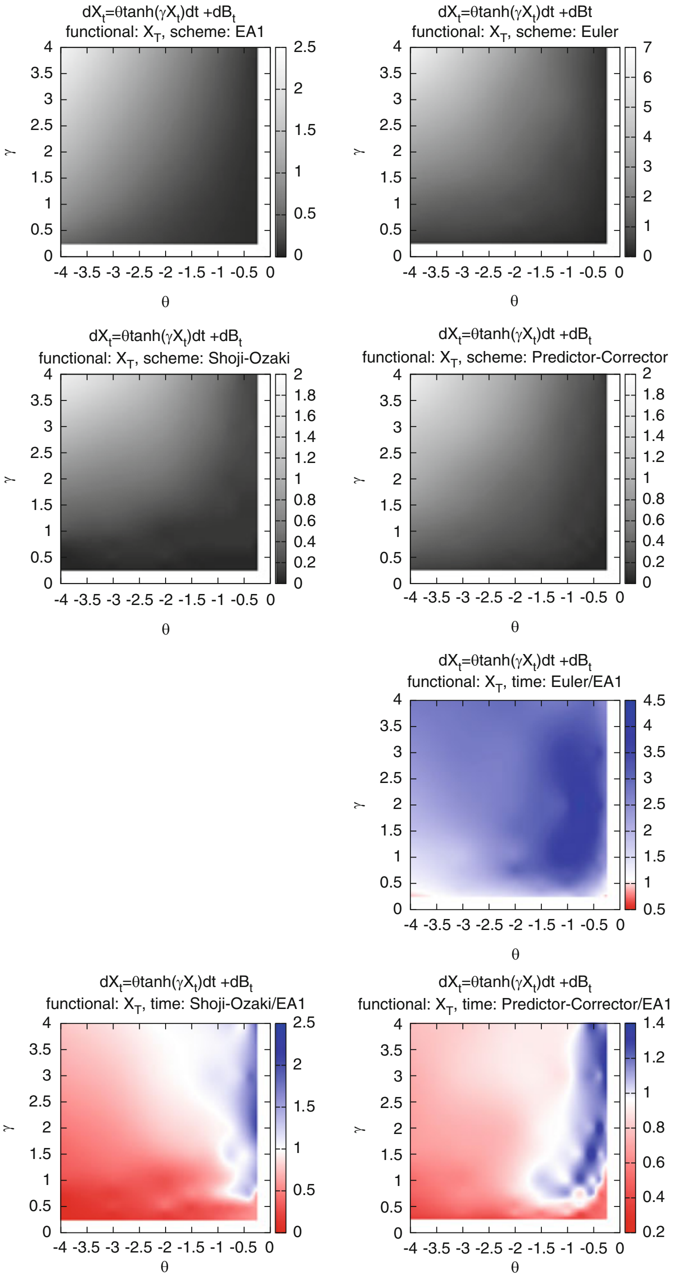


Fig. 4 Model: PTANH, functional: $M(X)$. Ozaki-Shoji scheme does not converge if $\theta = \gamma = 4$.



dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, scheme: EA1

dX_t=θtanh(γX_t)dt +dBt
functional: X_T, scheme: Euler

dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, scheme: Shoji-Ozaki

dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, scheme: Predictor-Corrector

dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, time: Euler/EA1

dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, time: Shoji-Ozaki/EA1

dX_t=θtanh(γX_t)dt +dB_t
functional: X_T, time: Predictor-Corrector/EA1

Fig. 5 Model: NTANH, functional: $L(X)$.

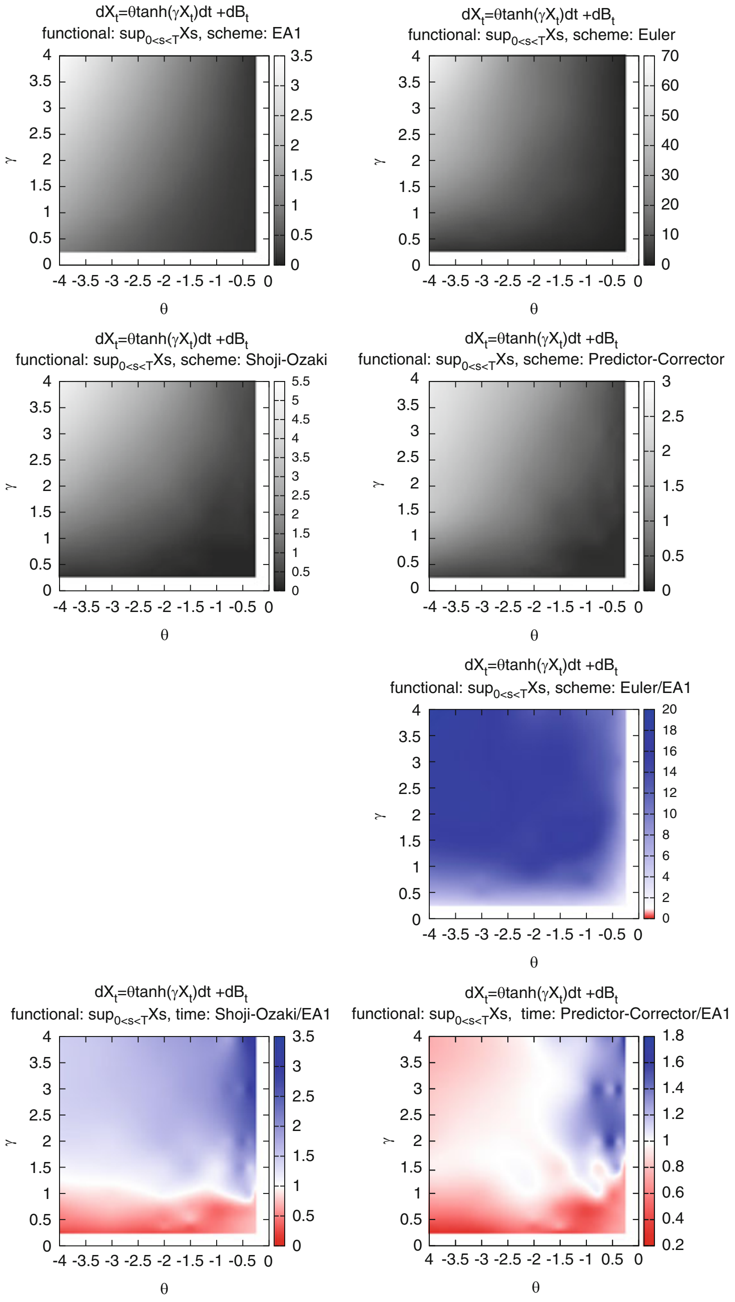


Fig. 6 Model: NTANH, functional: $M\,(X)$.

In the remaining three plots of each figure, the ratio of the computational time of a DS over the computational time of EA is represented on the Z-axis, again as a function of the SDE's parameters. Whenever possible, the white colour represents a unitary ratio, the red colour a ratio lower than 1 and the blue colour a ratio higher than 1. We remark that these ratios are the results of our arbitrary choices. For example comparing a higher number of observations would increase the power of the test and this would result in a lower efficiency of the DSs.

Moreover the shape of these surfaces is of interest on its own, as it says how the DSs behave with respect to parametric classes of drift and diffusion coefficients. From this point of view EA is a valuable validation tool.

The two main goals of this simulation are: commenting the efficiency of EA with respect to other DSs and study the behaviour of EA and of the DSs with respect to qualitative characteristics of the diffusion model X. Regarding the first of these points, we note that:

1. EA1 has a computational cost that is comparable to that of good DSs such as the Predictor-Corrector scheme. This means that there is generally not a huge difference between simulating from the approximated or the exact law of the process.
2. EA1 is favoured when we consider the functional $M(X)$. One possible explanation for this is that while simulating $L(X)$ the discretisation errors of every step are likely to cancel, but when simulating $M(X)$ the errors are likely to accumulate. Moreover, we are using two levels of approximations: we approximate the discretized path and also the maximum of the path conditionally on the discretisation.
3. While all DSs share a very good performance when γ is very low, independently of the value of θ, this is not the case with EA1. While the computational cost in EA1 remains very contained it increases with $|\theta|$ more rapidly. Conversely, EA1 has better efficiency than DSs when $|\theta|$ is low.
4. There are situations where EA1 performs much better, for instance for the PTANH model. This happens because if $\alpha^2 = \alpha'$ in (3) it follows that EA always accept the proposed skeleton. In this case we actually know the transition density of X. This is the case when $\gamma = \theta$ in the PTANH. When we move away from the diagonal the range of $\alpha^2 + \alpha'$ increases and so does the rejection rate.

Concerning the second of these points, we note that:

1. Euler scheme is clearly the least efficient DS. In some situation it can be 20 times more inefficient than the other two DSs. Moreover the implementation difficulty off all these DSs is comparable.
2. Predictor-Corrector and Ozaki-Shoji scheme shares more or less the same efficiency, even if in the same situations the former can be two times more efficient than the latter. Furthermore, the Ozaki-Shoji scheme exhibits numerical instabilities every time $\alpha'\left(X_{(i-1)\Delta}\right) \approx 0$. Hence it is necessary to introduce an

extra check for the algorithm that would slow down the simulation even more. All this suggests that the Predictor-Corrector scheme should be the first choice in most situations.

3. As already stated, the weak convergence criterion is not very useful from a practitioner point of view. In fact both the Euler DS and the Predictor-Corrector DS share the same unit-order of weak convergence.
4. It is very difficult from this limited study to infer any link between the efficiency of the DSs and the qualitative behaviour of the target diffusion model X. We just remark that the computational time surface has more or less the same shape in all the DSs. The difference is in the multiplicative factor.

3.2 *The Case of EA3*

We consider the following diffusion models

- The LANG SDE

$$dX_t = -k \operatorname{sign}(X_t) |X_t|^{\beta} \, dt + dB_t \qquad k > 0, \beta \in \mathbb{N} \tag{30}$$

- The XXCUBE SDE

$$dX_t = \left\{-\alpha X_t^3 + \beta X_t\right\} dt + dB_t \qquad \alpha > 0, \beta > 0 \tag{31}$$

In the case of EA3, we can no longer easily and exactly simulate from the law of $M(X)$, hence the comparison is only limited to the $L(X)$ functional. As the results of Sect. 3.1 suggests that Shoji-Ozaki scheme does not offer any clear advantage against Predictor-Corrector scheme, while showing numerical instabilities, we decide to include the Euler DS and the Predictor-Corrector DS in the comparison only.

Regarding the efficiency of EA3 with respect to Predictor-Corrector scheme (Figs. 7 and 8), we notice that the former is always less efficient then the the latter. The most obvious reason is that EA3 is much more complicated from an algorithmic point of view than EA1, and this results in a higher computational time. However, everything is relative to the choice of the specific comparison criterion considered. As a rule of thumb we can say that EA3 is a factor of 10 slower than EA1.

Given these results, there is no obvious link between qualitative behaviour of the diffusion model X and the expected efficiency of the DSs. The relative efficiency of Euler with respect to Predictor-Corrector is confirmed. But for the first time we observe a difference in the shape of the computational time surfaces of the Euler and the Predictor-Corrector schemes. This is the case of the LANG model. More investigation is needed to find the reasons of this result.

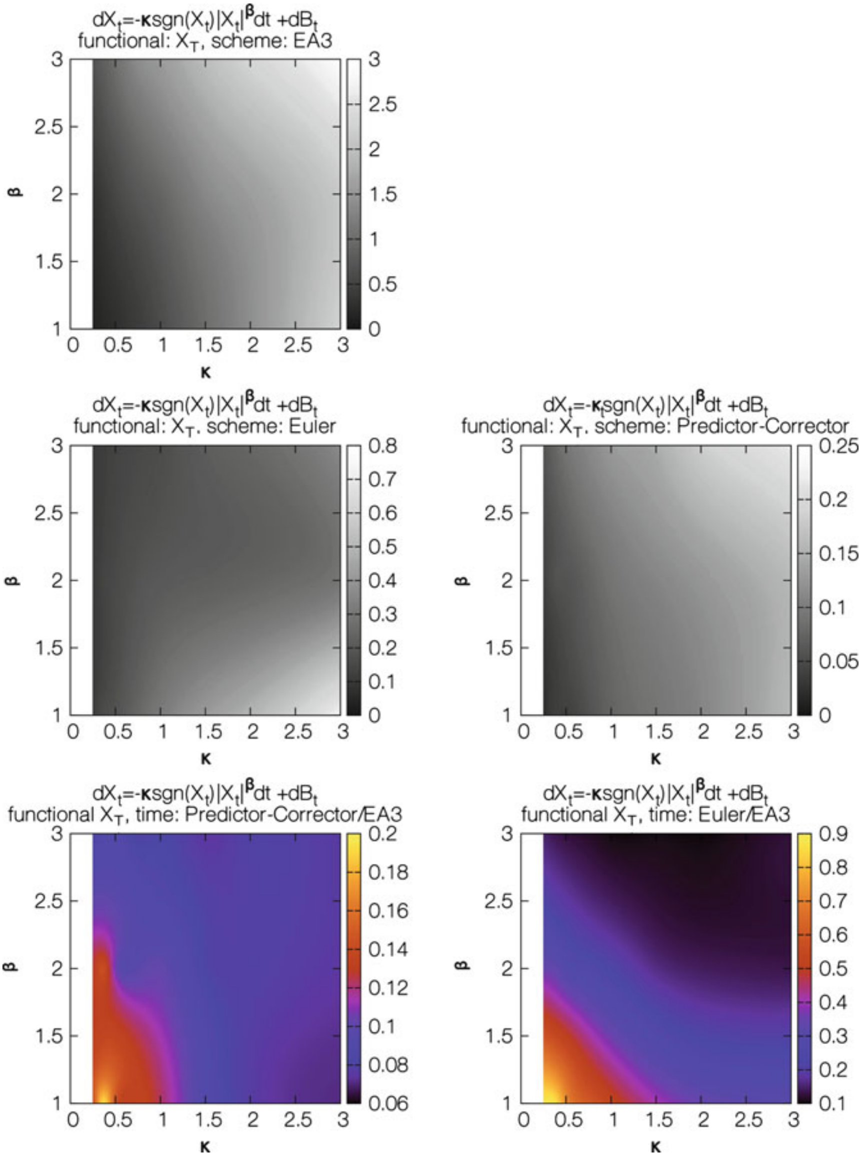


Fig. 7 Model: LANG, functional: $L(X)$.

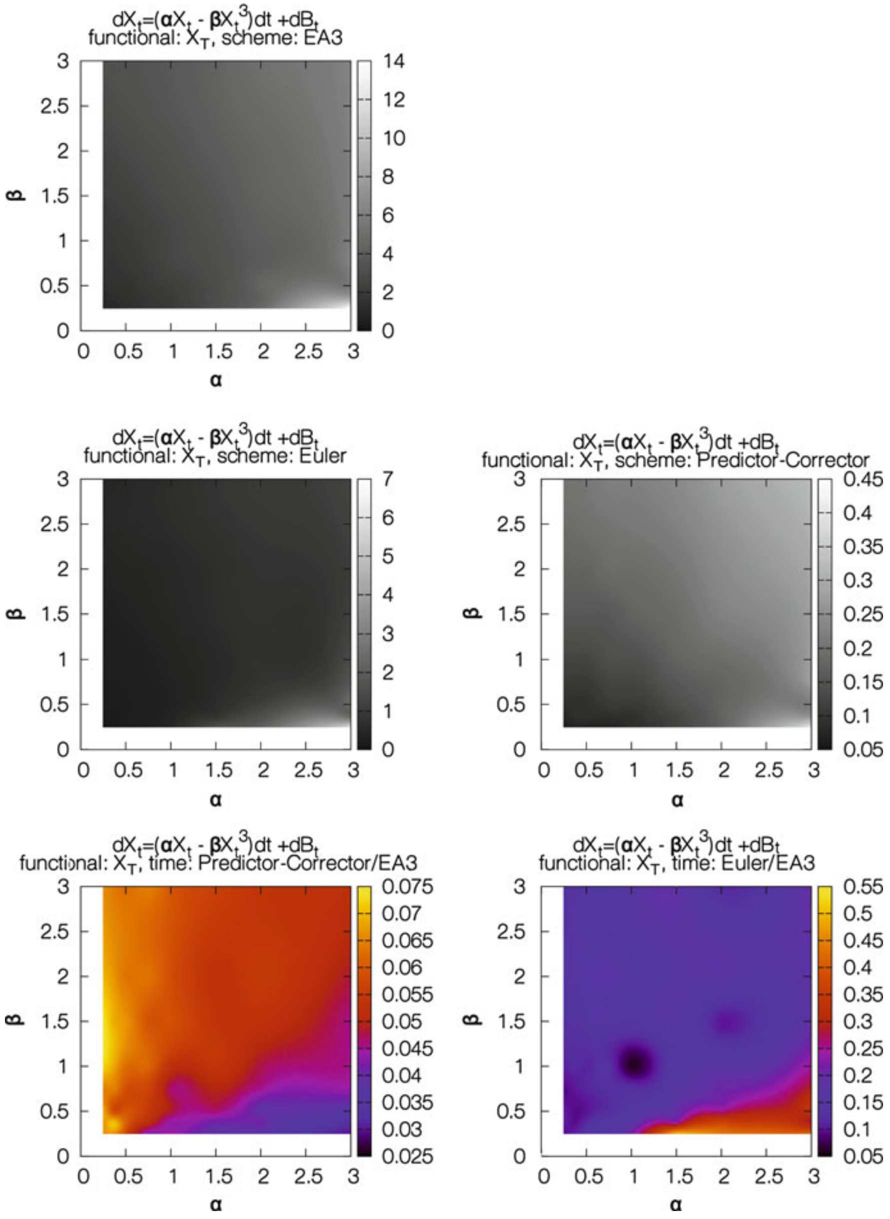


Fig. 8 Model: XXCUBE, functional: $L\,(X)$.

4 The Multi-dimensional Setting

We now concentrate on the unit-diffusion d-dimensional SDE

$$\begin{aligned} d\mathbf{X}_t &= \alpha\left(\mathbf{X}_t\right)dt + d\mathbf{B}_t \qquad\qquad t \in [0, T] \\ \mathbf{X}_0 &= \mathbf{x} \end{aligned} \tag{32}$$

where $\mathbf{B}_t$ is the d-dimensional BM. The drift coefficient α is assumed to satisfy proper conditions that guarantee the existence of a unique non-explosive strong solution of (32). In this section $\mathbb{Q}_T^{\mathbf{x}}$ and $\mathbb{W}_T^{\mathbf{x}}$ represent the law of the diffusion process $\mathbf{X}$ solution of (32) and the d-dimensional Wiener measure for the initial condition $\mathbf{B}_0 = \mathbf{x}$ respectively. Let $\mathbb{X}$ be the state space of $\mathbf{X}$.

It is possible to find equivalent conditions to $(C1)$–$(C4)$ for the d-dimensional framework and we refer to [6] for a formal development of EA in this setting. The main theoretical limitations of EA in the d-dimensional setting are:

1. The necessary and sufficient condition for the existence of a transformation from a generic d-dimensional SDE to the unit diffusion coefficient SDE (32) is quite demanding (see [3]);
2. We require the existence of a potential function $A : \mathbb{R}^d \to \mathbb{R}$ such that $\alpha\left(\mathbf{u}\right) = \nabla A\left(\mathbf{u}\right)$.

EA then generalises to this setting in a simple way. We define the d-dimensional BBM $\mathbf{Z}$ as a d-dimensional BM with initial value $\mathbf{x}$ conditioned on having its final value $\mathbf{Z}_T$ distributed according to $h_{\mathbf{x},t}\left(\mathbf{u}\right)$ where

$$h_{\mathbf{x},t}\left(\mathbf{u}\right) \propto \exp\left\{A\left(\mathbf{u}\right) - \frac{\parallel \mathbf{u} - \mathbf{x} \parallel^2}{2T}\right\} \tag{33}$$

and denote with $\mathbb{Z}_T^{\mathbf{x}}$ its law. Let $\phi : \mathbb{X} \to \mathbb{R}$, assumed to be bounded below, be defined as $\phi\left(\mathbf{u}\right) := \left(\parallel \alpha\left(\mathbf{u}\right) \parallel^2 + \mathrm{div}\alpha\left(\mathbf{u}\right)\right)/2 - l$ and $l := \inf_{r \in \mathbb{X}} \phi\left(\mathbf{r}\right) < \infty$. As before $\mathbb{L}$ denote the law of a unit rate Poisson Point Process (PPP) on $[0, T] \times [0, \infty)$, and let $\Phi = \{\chi, \psi\}$ be distributed according to $\mathbb{L}$. We define the event Γ as

$$\Gamma := \bigcap_{j \geq 1} \phi\left(\mathbf{Z}_{\chi_j}\right) \leq \psi_j \tag{34}$$

The following extension of Theorem 1 holds

Theorem 2 (Multivariate Wiener-Poisson factorisation). *If* $\left(\mathbf{Z}, \Phi\right) \sim \mathbb{Z}_T^{\mathbf{x}} \otimes \mathbb{L} \mid \Gamma$ *then* $\mathbf{Z} \sim \mathbb{Q}_T^{\mathbf{x}}$

Proof. see [6]

Using Theorem 2, the extension of EA1 to the d-dimensional setting is immediate. The only difficulty is finding the global maximum of ϕ over the domain $\mathbb{X}$. The extension of EA3 to the d-dimensional setting is similarly immediate, with the

Table 1 The multi-dimensional EA1.

Dimension	1	2	4	8	16
EA1 comp.cost	0.48	0.92	1.85	5.56	27.51

Table 2 The multi-dimensional EA3.

Dimension	1	2	3	4	5	6	7	8	9	10	11
LPS acceptance	83.9	71.3	61.3	52.3	44.4	37.4	32.5	27.2	23.3	19.3	17.2
EA3 comp. cost	0.39	0.31	0.40	0.46	0.75	1.21	2.15	5.87	15.9	45.9	129.9

added difficulty that we now have to compute the maximum of ϕ over a bounded d-dimensional hyper-rectangle in $\mathbb{X}$.

As in the case of the one-dimensional EA the simulation of $\mathbf{Z}$ requires to sample from $\{h_{\mathbf{x},T}(\mathbf{u})\}_{\mathbf{x}\in\mathbb{X}}$. Unfortunately the high dimensionality of the problem makes any adaptive approach, such as the ones in [14], infeasible. However, if we can find a d-dimensional matrix K, a vector $\mathbf{v}$ and a constant k such that

1. $\forall \mathbf{u} \in \mathbb{X}\; A(\mathbf{u}) \leq (\mathbf{u}-\mathbf{v})' K (\mathbf{u}-\mathbf{v}) + k$
2. $\int_{\mathbb{X}} \exp\left\{(\mathbf{u}-\mathbf{v})' K (\mathbf{u}-\mathbf{v}) - \frac{\|\mathbf{u}-\mathbf{x}\|^2}{2T}\right\} < \infty$

it is possible to implement a simple accept-reject sampler using a multivariate Gaussian variate as proposal (the LPS from now on). In most diffusion models of interest it is possible to find such $K, \mathbf{v}, k$ that satisfies these conditions (at least for T small enough) indeed.

To see how the computation cost of EA scales as d increases we considered two test d-dimensional SDEs defined by their potential function A:

- The d-dimensional SINE, $A(\mathbf{u}) = -\cos\left(\sum_{i=1}^{d} u_i\right)$
- The d-dimensional LANG, $A(\mathbf{u}) = -\sum_{i=1}^{n} u_i^4$

The initial value $\mathbf{x}$ is the origin of $\mathbb{R}^d$ and $T = 1$. Theoretical consideration suggests that partitioning $[0, T]$ in sub-intervals of length T/d (and applying EA sequentially) would keep the acceptance rate of EA stable as d changes. Our simulation study suggests that this intuition is correct and we adopt this strategy.

In Table 1 we report the computational cost (in seconds) required to sample 1,000 observations from the d-dimensional SINE SDE using EA1. We see that, apart from variations due to the implementation, the computational cost increases linearly with d. Due to the bounded nature of this example the acceptance rate of the LPS is stable.

In Table 2 we report the computational cost of the d-dimensional EA3 required to sample 100 observations from the d-dimensional LANG SDE. While the acceptance rate of the LPS decreases with d (as expected) this is not the reason of the explosive behaviour of the d-dimensional EA3's computational cost. The problem is the computation of the maximum of over a bounded d-dimensional hyper-rectangle that requires at least 2^d computations.

5 Conclusions

In this paper we have performed a simulation study of EA's efficiency. We have investigated the computational time required by EA1 and EA3 in different scenarios, both in the one and d-dimensional setting. In the one-dimensional case the results of this simulation are compared with the computational time required by three other numerical schemes too. The results are encouraging: EA1 is proved to be very competitive with respect to the other DSs as the computational time required for an accurate approximation using traditional DSs is comparable to that necessary for an exact simulation of the SDE. Thus our opinion is that the exact nature of EA1 makes it the preferred discretisation scheme. Additionally, knowing the true distribution of the path of the process conditioned on the returned skeleton makes the exact simulation of some path-dependent functionals possible.

In the case of EA3, the added complexity of the algorithm has inevitable consequences for computing cost. The choice of suggested discretisation schemes thus depends on the particular application. When a very precise simulation is needed, EA3 still presents a reasonable efficiency, being roughly a factor of 10 slower then EA1.

In the d-dimensional case EA1 scales quadratically with the dimension d, while in most cases EA3 scales exponentially. However, in practice EA methods remain feasible in reasonable dimensional models.

Moreover, the exact nature of EA is of great importance when efficiency is not the first concern. Thanks to EA we have been able to analyse the efficiency of other discretisation schemes with a high degree of accuracy. This was achieved by considering diffusion models for which the exact solution is not available in a closed form.

One aim of this work was to obtain heuristics for the quality of approximations for DSs as a function of characteristic of the diffusion and drift coefficients themselves. In our study however, we drew no clear conclusions on this issue. However it is clear that EA methods will have a role to play in addressing these questions in future research.

Appendix

We briefly consider the algorithm EA3. Full details can be found in [6].

The probability that the BB Z stays in an arbitrary interval can be expressed as an infinite series only. As a consequence the direct simulation of the minimum and the maximum of Z is not feasible. However, we can rearrange the terms of this series so that the sequence of the partial sums s_n satisfies the relations:

$$s_{n-1} \leq l \Rightarrow s_n \geq l \tag{35}$$

$$s_{n-1} \geq l \Rightarrow s_n \leq l \tag{36}$$

where l is the limit value of the series. As explained in [6] we can consider an increasing collection of nested intervals $\{I_n\}_{n\geq 1}$ which contains the starting and ending values of Z. Due to the behaviour of the partial sums s_n we can simulate the value n so that both the maximum and the minimum of Z are included in a specific I_n and at least one of them is included in $I_n \cap I_{n-1}^C$. Conditional on this event R_n the range of Z is bounded.

It remains to implement an algorithm to sample from $Z \mid R_n$, as we have to compute the value of this process at the time instances given by the PPP Φ. It is not sensible to use Z as a trivial RS proposal, the reason being that the number of proposed paths before the first acceptance has infinite expectation. A better RS algorithm proposes from a mixture of two probability measures with equal weight. One of them is the law of Z conditioned on achieving its minimum in $I_n \cap I_{n-1}^C$. The other one is the law of Z conditioned on achieving its maximum in $I_n \cap I_{n-1}^C$. Crucially, it is possible to sample the constrained minimum (or maximum) m of Z and the time τ at which Z hits this minimum (or maximum). Moreover $Z \mid m, \tau$ gets factorised in the product measure of two 3-dimensional Bessel bridges, whose simulation is trivial. As the Radon-Nikodym derivative of this proposal with respect to $Z \mid R_n$ is available in closed form we are done.

Implementational Issues for EA

The following material applies both to EA1 and EA3 implementation. From a practical point of view, every version of EA requires simulation from the density (9). This is not a trivial problem as the functional form of (9) depends on the drift coefficient α in (3). Moreover, theoretical results (see [5]) suggest that the acceptance rate of EA typically decreases exponentially with T. It turns out that it is usually more efficient to partition the time interval $[0, T]$ into smaller sub-intervals of length t and apply EA sequentially. This in turn implies that we have to sample from a parametric family of densities $\{h_{x,t}(u)\}_{x\in\mathbb{X}}$, as the starting value x is different on every sub-interval.

Furthermore the time spent in the simulation from $\{h_{x,t}(u)\}_{x\in\mathbb{X}}$ is not negligible in EA. In the particular case of EA1 roughly half of the time is spent in simulating from $\{h_{x,t}(u)\}_{x\in\mathbb{X}}$. Thus an efficient sampler results in a significantly lower computational cost for the EA. We briefly introduce two adaptive accept-reject samplers that we have developed to sample efficiently from $\{h_{x,t}(u)\}_{x\in\mathbb{X}}$ and we refer to [14] for a more detailed exposition.

We begin considering the case of a single $h_{x,t}$ for a fixed $x \in \mathbb{X}$ (t is always fixed). The first sampler, ARS1 from now on, requires the following semi sub-linear condition to hold

- $(E1)\ \exists n^+, N^+, m^-, M^-, \in \mathbb{R},\ c \in \mathbb{X}:$

$$\alpha(u) \leq n^+ + N^+ u \qquad c \leq u \tag{37}$$

$$m^- + M^- u \leq \alpha(u) \qquad u < c \tag{38}$$

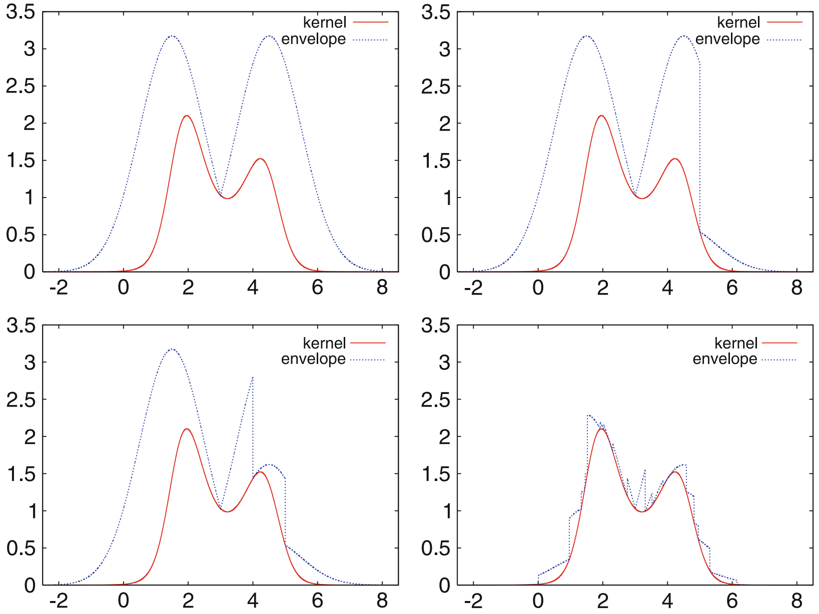


Fig. 9 The test kernel $h_{x,t}$ and the proposal q constructed from condition $(E1)$ for a test function $h_{x,t}$. Starting from quadrant IV going clockwise we have the envelope constructed from 1, 2, 3 points and the envelope that satisfies an acceptance rate of 95%.

The monotonicity of the integral and of the exponential function thus implies the following bounds on $h_{x,t}$

$$h_{x,t}(u) \le q_+^{u_0}(u) := e^{-\frac{(u-x)^2}{2t}+A(u_0)+n^+(u-u_0)+\frac{N^+}{2}\left(u^2-u_0^2\right)} \qquad c \le u_0 < u \tag{39}$$

$$h_{x,t}(u) \le q_-^{u_0}(u) := e^{-\frac{(u-x)^2}{2t}+A(u_0)+m^-(u-u_0)+\frac{M^-}{2}\left(u^2-u_0^2\right)} \qquad u < u_0 < c \tag{40}$$

To construct the envelope, we start by considering the point $u_0 = c$ (c is required to be a point of the envelope in this algorithm). Then, the initial envelope is given by

$$q(u) = q_-^c(u)\, 1_{[u<c]} + q_+^c(u)\, 1_{[c \le u]} \tag{41}$$

We have successfully bounded $h_{x,t}$ from above with a piece-wise function formed by the kernels of a Gaussian density times finite constants. Using the bounds (39) and (40) it is possible to refine $q(u)$ by adding more points to it too. We illustrate the results of this procedure in Fig. 9. If α is sub-linear, a different construction of q results in a tighter envelope for the same number of points.

Considerable attention has been put in the implementation of an efficient algorithm to sample from ARS1:

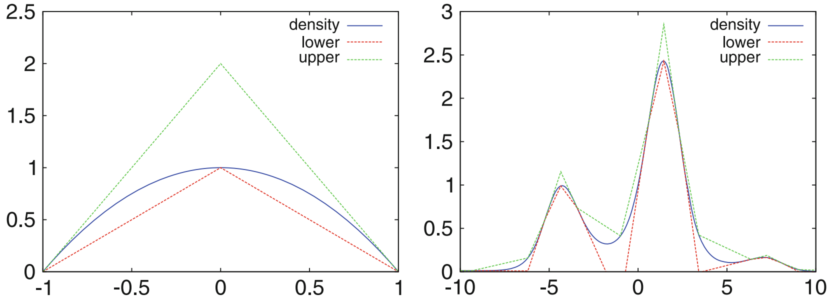


Fig. 10 The initial construction of the ARS2 on a single interval (*left*) and on the test density (*right*).

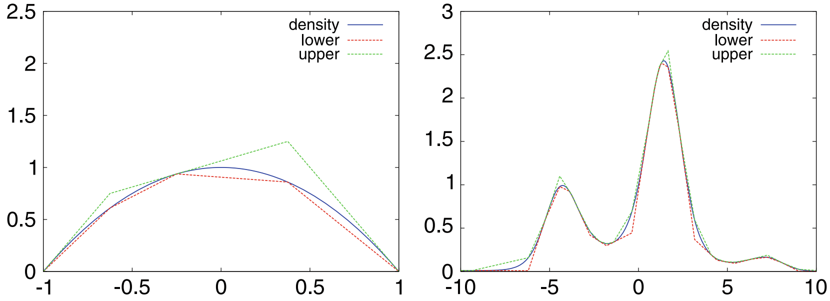


Fig. 11 The refined construction of the ARS2 on a single interval (*left*) and on the test density (*right*).

1. A binary search is performed (instead of a sequential one) to sample the interval of the piece-wise proposal q;
2. The same uniform variate used to sample the interval is used to sample from the proper truncated Gaussian distribution by the cdf inversion method;
3. All the values relevant to the algorithm are cached for re-use.

The second sampler, ARS2 from now on, has much weaker requirements than ARS1 and is of independent interest. We basically require the function $h_{x,t}$ to be piece-wise twice differentiable and to exhibit an exponential decay in the tails. This sampler is a generalisation of the adaptive accept-reject sampler introduced in [10, 11]. We partition the state space $\mathbb{X}$ into intervals where $h_{x,t}$ is convex/concave and use the geometric interpretation of convexity to construct linear bounds above and below $h_{x,t}$. We illustrate the results of this procedure in Figs. 10 and 11.

Similarly to the case of ARS1, considerable attention has been put in the implementation of an efficient algorithm to sample from ARS2. A brief simulation study in [14] reveals that the efficiency of ARS2 is comparable to that of the Gnu Scientific Library's ad-hoc samplers. ARS1, while somewhat less efficient, is a more robust sampler as it targets a very specific family of densities.

We now consider the more general problem of sampling from $\{h_{x,t}(u)\}_{x\in\mathbb{X}}$. Our idea is to slice the subset $\mathbb{D} \subseteq \mathbb{X}$ where the diffusion X is most likely to stay, to be found by a preliminary simulation, into a finite number of equi-spaced intervals. For each interval, we construct an envelope that uniformly bounds all the $h_{x,t}$ whose x is a point of this interval. To find this uniform bound we notice that for $l < r \in \mathbb{X}$

$$\sup_{l\leq x\leq r} h_{x,t} = \sup_{l\leq x\leq r} e^{A(u)-\frac{(u-x)^2}{2t}} \left\{1_{[u<l]} + 1_{[l\leq u\leq r]} + 1_{[r<u]}\right\} \tag{42}$$

$$\leq e^{A(u)-\frac{(u-l)^2}{2t}} 1_{[u<l]} + e^{A(u)} 1_{[l\leq u\leq r]} + e^{A(u)-\frac{(u-r)^2}{2t}} 1_{[r<u]} \tag{43}$$

$$\leq e^{A(u)-\frac{(u-l)^2}{2t}} 1_{[u<l]} + e^{Amax} 1_{[l\leq u\leq r]} + e^{A(u)-\frac{(u-r)^2}{2t}} 1_{[r<u]} \tag{44}$$

where $Amax = \sup_{l\leq u\leq r} A(u) < \infty$ as A is a continuous function on a bounded interval, hence A is bounded. The first and the last term of (44) can be easily bounded by envelopes resulting from ARS1 or ARS2. Regarding the central term of (44) we propose the trivial accept-reject sampling algorithm whose acceptance rate is high if the length of the intervals is reasonably short. We thus pre-compute and cache all these uniform envelopes, one for each intervals in which we split $\mathbb{D}$. During the simulation according to EA, if $x \in \mathbb{D}$ we select the right envelope, otherwise (an event whose probability can be arbitrarily small increasing $\mathbb{D}$) we create an envelope accordingly. As the intervals are equi-spaced there is virtually no efficiency penalty in searching for the right envelope.

References

1. Giles, Mike B.(2008) Multilevel Monte Carlo Path Simulation. In: Operations Research, vol 56, No. 3, pp 607–617.
2. Higham D. J., Mao X., Stuart A. M. (2002) Strong convergence of Euler-type methods for nonlinear stochastic differential equations. In: Journal of Numerical Analysis, vol 40, No. 3, pp 1041–1063.
3. Ait-Sahalia, Y. (2008) Closed-Form Likelihood Expansions for Multivariate Diffusions. In: Annals of Statisticss, vol 36, No. 3, pp 906–937, NBER.
4. Albanese C., Kuznetsov A. (2005) Transformations of Markov processes and classification scheme for solvable driftless diffusions. www3.imperial.ac.uk/mathfin/people/calban/papersmathfi).
5. Beskos A., Papaspiliopoulos O., Roberts G.O. (2006) Retrospective exact simulation of diffusion sample paths with applications. In: Bernoulli, vol 12, pp 1077–1098.
6. Beskos A., Papaspiliopoulos O., Roberts G.O. (2008) A new factorisation of diffusion measure and finite sample path construction. In: Methodology and Computing in Applied Probability, vol 10, No. 1, pp 85–104.
7. Beskos A., Roberts G.O. (2005) Exact simulation of diffusions In: Ann. Appl. Probab, vol 15, pp 2422–2444.
8. Bruno Casella (2005) Exact MC simulation for diffusion and jump-diffusion processes with financial applications. IMQ - Bocconi University.

9. Milstein G. N., Tretyakov M. V. (2004) Stochastic numerics for Mathematical Physics, Springer-Verlag New York.
10. Gilks WR (1992) Derivative-free adaptive rejection sampling for Gibbs sampling. In: Bayesian Statistics, vol 4, No. 2, pp 641–649.
11. Gilks WR, Wild P. (1992) Adaptive Rejection Sampling for Gibbs Sampling. In: Applied Statistics, vol 41, No. 2, pp 337–348, JSTOR.
12. Kloeden P.E., Platen E. (1992) Numerical Solution of Stochastic Differential Equations, Springer.
13. Maruyama G. (1955) Continuous Markov processes and stochastic equations. In: Rend. Circ. Mat. Palermo, vol 4, pp 48–90.
14. Stefano Peluchetti (2007) An analysis of the efficiency of the Exact Algorithm, IMQ - Università Commerciale Luigi Bocconi.
15. Shoji I., Ozaki T. (1998) Estimation for nonlinear stochastic differential equations by a local linearization method. In: Stochastic Analysis and Applications. vol 16, No. 4, pp 733–752, Taylor & Francis.

Polynomial Lattice Point Sets

Friedrich Pillichshammer

Abstract Polynomial lattice point sets are special types of (t, m, s)-nets as introduced by H. Niederreiter in the 1980s. Quasi-Monte Carlo rules using them as underlying nodes are called polynomial lattice rules. In their overall structure polynomial lattice rules are very similar to usual lattice rules due to E. Hlawka and N. M. Korobov. The main difference is that here one uses polynomial arithmetic over a finite field instead of the usual integer arithmetic. In this overview paper we give a comprehensive review of the research on polynomial lattice rules during the last decade. We touch on topics like extensible polynomial lattice rules, higher order polynomial lattice rules and the weighted discrepancy of polynomial lattice point sets. Furthermore we compare polynomial lattice rules with lattice rules and show what results for polynomial lattice rules also have an analog for usual lattice rules and vice versa.

1 Introduction

Assume we are interested in the approximation of multivariate integrals of the form $I_s(f) = \int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ using a quasi-Monte Carlo (QMC) rule of the form $Q_{N,s}(f) = (1/N)\sum_{n=0}^{N-1} f(\boldsymbol{x}_n)$ where $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}$ are fixed sample nodes from the unit cube $[0,1)^s$. On first sight this approach looks quite simple but the crux of this method is the choice of underlying nodes to obtain good approximations for large classes of functions.

Generally speaking, point sets with good uniform distribution properties yield a small absolute integration error. This is, for example, reflected in the Koksma-Hlawka inequality which states that

F. Pillichshammer
Institut für Finanzmathematik, Universität Linz, Altenbergerstraße 69, A-4040 Linz, Austria
e-mail: friedrich.pillichshammer@jku.at

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_8, © Springer-Verlag Berlin Heidelberg 2012

$$|I_s(f) - Q_{N,s}(f)| \le V(f) D_N^*(\mathscr{P})$$

where $V(f)$ is the variation of f in the sense of Hardy and Krause and where D_N^* denotes the star discrepancy of the point set $\mathscr{P} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\}$. The star discrepancy can be defined as follows: given a point set $\mathscr{P} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1}\}$ of N elements in $[0,1)^s$ the *discrepancy function* of $\mathscr{P}$ is defined by

$$\Delta_{\mathscr{P}}(\boldsymbol{z}) := \frac{\#\{0 \le n < N \ : \ \boldsymbol{x}_n \in [\boldsymbol{0}, \boldsymbol{z})\}}{N} - \lambda_s([\boldsymbol{0}, \boldsymbol{z})) \quad \text{for} \quad \boldsymbol{z} \in (0,1]^s,$$

where λ_s is the s-dimensional Lebesgue measure. The *star discrepancy* of $\mathscr{P}$ is then the L^∞-norm of $\Delta_{\mathscr{P}}$, i.e.,

$$D_N^*(\mathscr{P}) = \sup_{\boldsymbol{z} \in (0,1]^s} |\Delta_{\mathscr{P}}(\boldsymbol{z})|.$$

This is a quantitative measure for the deviation of $\mathscr{P}$ from uniform distribution modulo one. For more information on the Koksma-Hlawka inequality and the star discrepancy we refer to the books [22, 26, 44, 58].

For any point set $\mathscr{P}$ consisting of N points in $[0,1)^s$ it is known that

$$D_N^*(\mathscr{P}) \ge c_s (\log N)^{\kappa_s}/N$$

with a positive c_s independent of $\mathscr{P}$ and where $\kappa_2 = 1$ (see [5, 71]) and $\kappa_s \ge (s-1)/2$ for $s \ge 3$ which follows from a result of Roth [68]. (For $s \ge 3$ the lower bound on κ_s has recently been improved to $\kappa_s \ge (s-1)/2 + \delta_s$ for some unknown $0 < \delta_s < 1/2$; see [6].)

On the other hand, a point set $\mathscr{P}$ whose star discrepancy satisfies an upper bound of the form $D_N^*(\mathscr{P}) \le C_s (\log N)^{\alpha_s}/N$ with a positive C_s independent of $\mathscr{P}$ and where $\alpha_s \ge 0$, is informally called a *low discrepancy point set*. There are several methods to construct low discrepancy point sets:

- Hammersley point sets which are based on the infinite van der Corput sequence (see, e.g., [22, 58]);
- Lattice point sets (or, more general, integration lattices) which were introduced independently by Korobov [38] and Hlawka [36] and which are well explained in the books of Niederreiter [58] and of Sloan and Joe [72];
- (t,m,s)-nets in base b which were introduced by Niederreiter [56, 58] and which are the main topic of the recent book [22]. Very special examples of such nets go back to constructions of Sobol' [77] and Faure [27].

In this article we are concerned with a sub-class of (t,m,s)-nets which has a close relation to lattice point sets. Before we give its definition we recall the definition of (t,m,s)-nets in base b according to Niederreiter [56].

Definition 1. Let b, s, m, t be integers such that $s \ge 1$, $b \ge 2$ and $0 \le t \le m$. A point set $\mathscr{P}$ consisting of b^m points in $[0,1)^s$ is called (t,m,s)-*net in base* b if every so-called b-adic elementary interval of the form

$$\prod_{i=1}^{s}\left[\frac{a_i}{b^{d_i}},\frac{a_i+1}{b^{d_i}}\right)\subseteq[0,1)^s,\quad \text{where } a_i,d_i\in\mathbb{N}_0 \text{ for } 1\le i\le s,$$

of volume b^{t-m} contains exactly b^t points of $\mathscr{P}$.

Some remarks on the definition of (t,m,s)-nets in base b are in order (for more information see [22, 58]).

Remark 1. 1. Definition 1 states that for every b-adic elementary interval J volume b^{t-m} we have $\#\{\boldsymbol{x}\in\mathscr{P}\ :\ \boldsymbol{x}\in J\}-b^m\lambda_s(J)=0$.

2. The uniform distribution quality depends on the so-called *quality parameter* $t\in\{0,\dots,m\}$. A small t implies good uniform distribution. This is also reflected in Niederreiter's bound on the star discrepancy of a (t,m,s)-net $\mathscr{P}$ in base b which states that

$$D_N^*(\mathscr{P})=O_{s,b}(b^t(\log N)^{s-1}/N) \tag{1}$$

   where $N=b^m$; see [22,56,58], and where $O_{s,b}$ indicates that the implied constant depends on s and b.
3. The optimal value $t=0$ is only possible if the parameters b and s satisfy $s\le b+1$. On the other hand, any point set consisting of b^m elements in $[0,1)^s$ is an (m,m,s)-net in base b since this choice of parameters makes Definition 1 trivial (and also the discrepancy bound (1)).

As already mentioned we are concerned with a sub-class of (t,m,s)-nets. Introduced by Niederreiter [57, 58], today this sub-class is known as polynomial lattice point sets. This name has its origin in a close relation to ordinary lattice point sets. In fact, the research on polynomial lattice point sets and on ordinary lattice point sets often follows two parallel tracks and bears a lot of similarities. It is the aim of this overview to review the, in the author's opinion, most important results on polynomial lattice point sets during the last decade and to point out which of these results have counterparts for lattice point sets.

In the following two sections the basic definitions of (polynomial) lattice point sets and their duals are provided. In Sects. 4–9 we present the results on polynomial lattice point sets and point out their analogs for lattice point sets. The paper closes with a short summary and further remarks in Sect. 10.

Notation: Throughout the paper we assume that b is a prime number. By $\mathbb{Z}_b$ we denote the finite field with b elements and by $\mathbb{Z}_b[x]$ the set of polynomials over $\mathbb{Z}_b$. Define $G_{b,m}:=\{h\in\mathbb{Z}_b[x]\ :\ \deg(h)<m\}$ and $G_{b,m}^*=G_{b,m}\setminus\{0\}$. We have $|G_{b,m}|=b^m$.

The field of formal Laurent series over $\mathbb{Z}_b$ is denoted by $\mathbb{Z}_b((x^{-1}))$. Elements of $\mathbb{Z}_b((x^{-1}))$ are of the form

$$L = \sum_{\ell=w}^{\infty} t_\ell x^{-\ell}, \quad \text{where } w \in \mathbb{Z} \text{ and all } t_\ell \in \mathbb{Z}_b.$$

For $n \in \mathbb{N}$ let $\nu_n : \mathbb{Z}_b((x^{-1})) \to [0, 1)$ be defined by $\nu_n(L) = \sum_{\ell=\max(1,w)}^{n} t_\ell b^{-\ell}$.

For $x \in \mathbb{R}$ let $\{x\}$ denote the fractional part of x, and by $\{\boldsymbol{x}\}$ for $\boldsymbol{x} \in \mathbb{R}^s$ we mean that the fractional part is applied component-wise.

In many results which we are going to present in the following sections there appear constants c which are assumed to be different from case to case. Optionally these constants may depend on the dimension s, on b or on other quantities which are then indicated as sub-scripts. In most cases these constants could be given explicitly.

2 Polynomial Lattice Point Sets

On account of their close relation to polynomial lattice point sets we first recall the possibly more familiar concept of lattice point sets:

Definition 2. For an integer $N \geq 2$ and for $\boldsymbol{g} \in \mathbb{Z}^s$ the point set $\mathscr{P}(\boldsymbol{g}, N)$ consisting of the N elements

$$\boldsymbol{x}_n = \left\{ \frac{n}{N} \boldsymbol{g} \right\} \quad \text{for all } 0 \leq n < N$$

is called a *lattice point set (LPS)*. A QMC rule using $\mathscr{P}(\boldsymbol{g}, N)$ as underlying node set is called a *lattice rule*.

Polynomial lattice point sets are in their overall structure very similar to LPSs. The main difference is that LPSs are based on number theoretic concepts whereas polynomial lattice point sets are based on algebraic methods (polynomial arithmetic over a finite field). For simplicity we only discuss polynomial lattice point sets in prime base b. For the more general case of prime-power bases we refer to [22, 58].

Definition 3. For $s, m \in \mathbb{N}$, $p \in \mathbb{Z}_b[x]$, with $\deg(p) = m$, and $\boldsymbol{q} \in \mathbb{Z}_b[x]^s$ the point set $\mathscr{P}(\boldsymbol{q}, p)$ consisting of the b^m elements

$$\boldsymbol{x}_h = \nu_m \left(\frac{h(x)}{p(x)} \boldsymbol{q}(x) \right) \quad \text{for all } h \in G_{b,m}$$

is called a *polynomial lattice point set (PLPS)*. A QMC rule using $\mathscr{P}(\boldsymbol{q}, p)$ as underlying node set is called a *polynomial lattice rule*.

Note that we obtain an LPS when we choose $b = N, m = 1$ and $p(x) = x$. The structural similarity between Definitions 2 and 3 is evident. Hence let us compare the two concepts by means of some pictures.

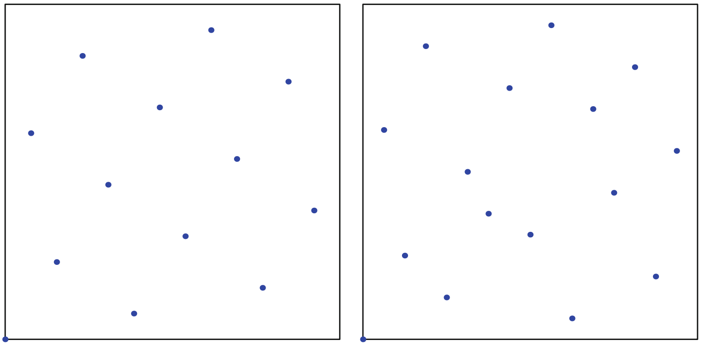


Fig. 1 *left*: $\mathscr{P}(\boldsymbol{g}, N)$ with $N = 13$ and $\boldsymbol{g} = (1, 8)$; *right*: $\mathscr{P}(\boldsymbol{q}, p)$ with $p(x) = x^4 + x^2 + 1$ and $\boldsymbol{q} = (1, x^3)$ over $\mathbb{Z}_2$.

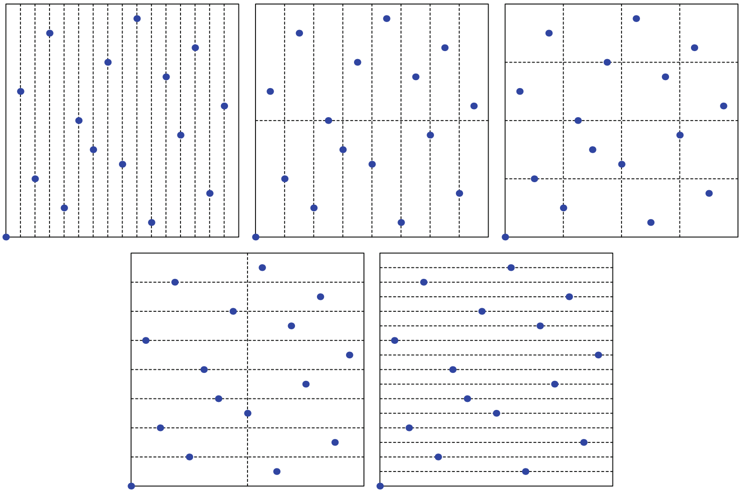


Fig. 2 $\mathscr{P}(\boldsymbol{q}, p)$ from Fig. 1 as $(0, 4, 2)$-net in base 2; every 2-adic elementary interval of area 2^{-4} contains exactly one point.

The LPS $\mathscr{P}(\boldsymbol{g}, N)$ shown in the left part of Fig. 1 shows a very regular lattice structure. Such a geometric structure cannot be observed for the PLPS $\mathscr{P}(\boldsymbol{q}, p)$ shown in the right part of Fig. 1. However, also this point set has some inherent structure, namely the (t, m, s)-net structure. In fact, for this example every 2-adic elementary interval of area 2^{-4} contains exactly one element of the point set $\mathscr{P}(\boldsymbol{q}, p)$ and hence we have a $(0, 4, 2)$-net in base 2; cf. Fig. 2.

A further example of an LPS and a PLPS is shown in Fig. 3.

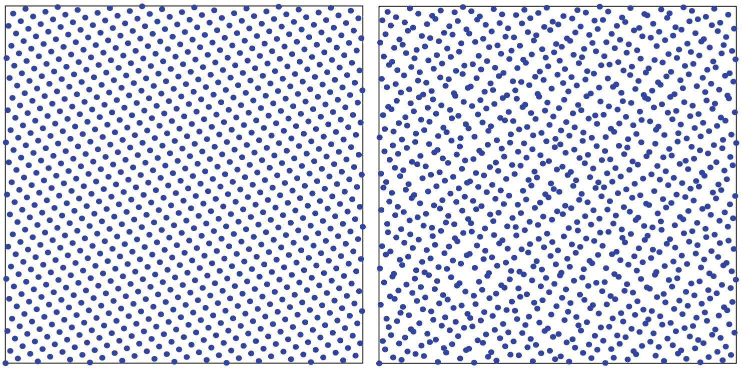


Fig. 3 *left*: $\mathscr{P}(\boldsymbol{g}, N)$ with $N = 987$ and $\boldsymbol{g} = (1{,}610)$; *right*: $\mathscr{P}(\boldsymbol{q}, p)$ with $p(x) = x^{10} + x^8 + x^4 + x^2 + 1$ and $\boldsymbol{q} = (1, x^9 + x^5 + x)$ over $\mathbb{Z}_2$.

3 The Dual Net

For LPSs one has the notion of a dual lattice which plays a crucial role in the quality analysis of such point sets.

Definition 4. The *dual lattice* of the LPS $\mathscr{P}(\boldsymbol{g}, N)$ from Definition 2 is defined as

$$\mathscr{L}_{\boldsymbol{g},N} = \{\boldsymbol{h} \in \mathbb{Z}^s \ : \ \boldsymbol{h} \cdot \boldsymbol{g} \equiv 0 \pmod{N}\}.$$

An important property of LPSs is that

$$\sum_{\boldsymbol{x} \in \mathscr{P}(\boldsymbol{g},N)} \mathrm{e}_{\boldsymbol{k}}(\boldsymbol{x}) = \begin{cases} N & \text{if } \boldsymbol{k} \in \mathscr{L}_{\boldsymbol{g},N}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathrm{e}_{\boldsymbol{k}}(x) = \exp(2\pi \mathtt{i} \boldsymbol{k} \cdot \boldsymbol{x})$. This relation is the reason why for the analysis of the integration error of lattice rules it is most convenient to consider one-periodic functions; see [58, 72].

The corresponding definition for PLPSs leads to the notion of a dual net.

Definition 5. The *dual net* of the PLPS $\mathscr{P}(\boldsymbol{q}, p)$ from Definition 3 is defined as

$$\mathscr{D}_{\boldsymbol{q},p} = \{\boldsymbol{k} \in G_{b,m}^s \ : \ \boldsymbol{k} \cdot \boldsymbol{q} \equiv 0 \pmod{p}\}.$$

An important property of PLPSs is that (see [22, Lemmas 4.75 and 10.6])

$$\sum_{\boldsymbol{x} \in \mathscr{P}(\boldsymbol{q},p)} {}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) = \begin{cases} b^m & \text{if } \boldsymbol{k} \in \mathscr{D}_{\boldsymbol{q},p}, \\ 0 & \text{otherwise,} \end{cases}$$

where ${}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})$ is the $\boldsymbol{k}$th b-adic Walsh function defined by ${}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) := \prod_{i=1}^{s} {}_b\mathrm{wal}_{k_i}(x_i)$ for $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ and $\boldsymbol{x} = (x_1, \ldots, x_s) \in [0,1)^s$. The one-dimensional kth b-adic Walsh function is defined by ${}_b\mathrm{wal}_k(x) := \exp(2\pi \mathtt{i}(\xi_1\kappa_0 + \cdots + \xi_{a+1}\kappa_a)/b)$ for $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_a b^a$ with $\kappa_i \in \{0, \ldots, b-1\}$ and $x = \xi_1 b^{-1} + \xi_2 b^{-2} + \cdots$ with infinitely many digits $\xi_i \neq b-1$. Many properties of Walsh functions are summarized in [22, Appendix A].

The above relation is the reason why for the analysis of the integration error of polynomial lattice rules it is most convenient to consider Walsh series. We will come back to this issue in Sect. 6.

4 Quality Measures and Existence Results

Based on the dual net one can introduce two quality measures for PLPSs (see [58, Chap. 4] or [22, Chap. 10]): for $p \in \mathbb{Z}_b[x]$ and $\boldsymbol{q} \in \mathbb{Z}_b[x]^s$ define

$$\rho(\boldsymbol{q}, p) = s - 1 + \min_{\boldsymbol{h} \in \mathscr{D}_{\boldsymbol{q},p} \setminus \{\boldsymbol{0}\}} \sum_{i=1}^{s} \deg(h_i)$$

and

$$R_b(\boldsymbol{q}, p) = \sum_{\boldsymbol{h} \in \mathscr{D}_{\boldsymbol{q},p} \setminus \{\boldsymbol{0}\}} \prod_{i=1}^{s} r_b(h_i),$$

where $r_b(0) = 1$ and $r_b(h) = b^{-r-1} \sin^{-2}(\pi \kappa_r / b)$ for $h \in G_{b,m}$ of the form $h = \kappa_0 + \kappa_1 b + \cdots + \kappa_r x^r$, $\kappa_r \neq 0$.

We remark here that analogous quality measures also exist for LPSs; see [58, Chap. 5]. Based on these quality measures Niederreiter [58] proved the following results:

Theorem 1. *The PLPS $\mathscr{P}(\boldsymbol{q}, p)$ is a (t, m, s)-net in base b with $m = \deg(p)$, $t = m - \rho(\boldsymbol{q}, p)$ and*

$$D^*_{b^m}(\mathscr{P}(\boldsymbol{q}, p)) \leq \frac{s}{b^m} + R_b(\boldsymbol{q}, p).$$

For example for $p = x^4 + x^2 + 1$ and $\boldsymbol{q} = (1, x^3)$ over $\mathbb{Z}_2$ the “minimal” element of $\mathscr{D}_{\boldsymbol{q},p}$ is $(h_1, h_2) = (x^2 + 1, x)$ and hence $\rho(\boldsymbol{q}, p) = 4$ in this case. Theorem 1 then shows that $\mathscr{P}(\boldsymbol{q}, p)$ is a $(0, 4, 2)$-net in base 2; cf. Fig. 2. Theorem 1 also gives a bound on the star discrepancy of PLPSs which is easier to handle than $D^*_{b^m}$ itself (note that the exact computation of the star discrepancy of a given point set is an NP-hard problem, see [28]). For an analogous discrepancy bound for LPSs we refer to [58, Chap. 5] or [22, Proposition 3.49]. Based on Theorem 1 one can use averaging arguments to obtain the following existence results:

Theorem 2. *Let $p \in \mathbb{Z}_b[x]$ with* $\deg(p) = m$.

1. *If p is irreducible, then there exists $\boldsymbol{q} \in G_{b,m}^s$ such that*

$$t \leq (s-1)\log_b m - (s-2) - \log_b \frac{(s-1)!}{(b-1)^{s-1}}.$$

   Hence $D_{b^m}^*(\mathscr{P}(\boldsymbol{q},p)) = O_{s,b}\left(m^{2s-2}b^{-m}\right)$.
2. *For $0 \leq \varepsilon < 1$ there are more than $\varepsilon|G_{b,m}^s|$ vectors $\boldsymbol{q} \in G_{b,m}^s$ with*

$$D_{b^m}^*(\mathscr{P}(\boldsymbol{q},p)) \leq \frac{s}{b^m} + R_b(\boldsymbol{q},p) = O_{s,b,\varepsilon}\left(\frac{m^s}{b^m}\right).$$

Part 1 of Theorem 2 for $b = 2$ has been shown by Larcher et al. [51]; see also [70] or [22, Chap. 10] for general b. Part 2 has been shown by Niederreiter [58, Chap. 4] and also by Dick et al. [14] and [18]. For an analogous discrepancy bound for LPSs we refer to [58, Chap. 5] or [22, Theorem 3.51].

The bound on R_b in Theorem 2 is best possible in the order of magnitude in m. This was shown recently by Kritzer and the author in [42]. A corresponding result for LPSs has been shown by Larcher [49].

Theorem 3. *There exists $c_{s,b} > 0$ such that for any $p \in \mathbb{Z}_b[x]$ with $\deg(p) = m$ and any $\boldsymbol{q} \in G_{b,m}^s$, $q_i \neq 0$, $1 \leq i \leq s$, we have*

$$R_b(\boldsymbol{q},p) \geq c_{s,b} b^{\deg(\delta_s)} \frac{(m-\deg(\delta_s))^s}{b^m} \quad \textit{where } \delta_s := \gcd(q_1,\ldots,q_s,p).$$

On the other hand, the bound on $D_{b^m}^*$ in Theorem 2 is *not* best possible in the order of magnitude in m. For example, in dimension $s = 2$ the so-called Fibonacci PLPS has a star discrepancy of order $O_b(mb^{-m})$; see [58, Chap. 4] or [22, Chap. 10]. For arbitrary dimension s it was shown by Larcher [50] that for any $m \geq 2$ there exists $\boldsymbol{q} \in G_{b,m}^s$ with

$$D_{b^m}^*(\mathscr{P}(\boldsymbol{q},x^m)) = O_{s,b}\left(m^{s-1}(\log m)b^{-m}\right);$$

see also [43] for an extension of this result to more general polynomials p. A counterpart of Larcher's result for LPSs is known for dimension $s = 2$ only; see [48, Corollary 3].

5 CBC Construction of Polynomial Lattice Point Sets

According to Theorem 2, for any given irreducible polynomial $p \in \mathbb{Z}_b[x]$ there exist a sufficiently large number of "good" vectors $\boldsymbol{q}$ of polynomials which yield PLPSs with reasonably low star discrepancy. Now one aims at finding such vectors by computer search. Unfortunately a full search is not possible (except maybe for small values of m, s) since one has to check b^{ms} vectors of polynomials.

At this point one gets a cue from the analogy between PLPSs and LPSs where the component-by-component (CBC) construction approach works very well. This approach was introduced by Korobov [39] for LPSs and later it was re-invented by Sloan and Reztsov [73]. The same idea applies to PLPSs. Here we use the more general weighted star discrepancy as introduced by Sloan and Woźniakowski [74] as underlying quality criterion: let $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots)$ be a sequence of weights in $\mathbb{R}^+$. Let $\mathscr{I}_s = \{1, \ldots, s\}$ and for $\mathfrak{u} \subseteq \mathscr{I}_s$ let $\gamma_{\mathfrak{u}} = \prod_{i \in \mathfrak{u}} \gamma_i$. For an s-dimensional vector $\boldsymbol{z} = (z_1, \ldots, z_s)$ and for $\mathfrak{u} \subset \mathscr{I}_s$ the s-dimensional vector whose ith component is z_i if $i \in \mathfrak{u}$ and 1 if $i \notin \mathfrak{u}$ is denoted by $(\boldsymbol{z}_{\mathfrak{u}}, 1)$. The *weighted star discrepancy* of an N-element point set $\mathscr{P}$ in $[0,1)^s$ is given by

$$D^*_{N,\boldsymbol{\gamma}}(\mathscr{P}) = \sup_{\boldsymbol{z} \in (0,1]^s} \max_{\emptyset \neq \mathfrak{u} \subseteq \mathscr{I}_s} \gamma_{\mathfrak{u}} |\Delta_{\mathscr{P}}((\boldsymbol{z}_{\mathfrak{u}}, 1))|.$$

The weights $\boldsymbol{\gamma}$ are additional parameters which model the importance of the different coordinate projections. For the weights $\boldsymbol{\gamma} = \mathbf{1} := (1, 1, \ldots)$ one has $D^*_{N,\boldsymbol{\gamma}}(\mathscr{P}) = D^*_N(\mathscr{P})$ for any point set $\mathscr{P}$. In the weighted setting the CBC construction has the advantage that the quadrature points $\mathscr{P}$ can be optimized with respect to $\boldsymbol{\gamma}$.

The weighted Koksma-Hlawka inequality then states that

$$|I_s(f) - Q_{N,s}(f)| \leq D^*_{N,\boldsymbol{\gamma}}(\mathscr{P}) \|f\|_{s,\boldsymbol{\gamma}}$$

with a certain norm $\|\cdot\|_{s,\boldsymbol{\gamma}}$; see [37, 74] or [22, Chap. 2] for details.

Let $p \in \mathbb{Z}_b[x]$ with $\deg(p) = m$ and let $\boldsymbol{q} \in G^s_{b,m}$. Then it can be shown (see [22, Corollary 10.16]) that

$$D^*_{b^m,\boldsymbol{\gamma}}(\mathscr{P}(\boldsymbol{q}, p)) \leq \sum_{\emptyset \neq \mathfrak{u} \subseteq \mathscr{I}_s} \gamma_{\mathfrak{u}} \left(1 - \left(1 - \frac{1}{b^m}\right)^{|\mathfrak{u}|}\right) + R_{b,\boldsymbol{\gamma}}(\boldsymbol{q}, p),$$

where

$$R_{b,\boldsymbol{\gamma}}(\boldsymbol{q}, p) = \sum_{\boldsymbol{h} \in \mathscr{D}_{\boldsymbol{q},p} \setminus \{\mathbf{0}\}} \prod_{i=1}^{s} r_b(h_i, \gamma_i)$$

and where for $h \in G_{b,m}$ we put $r_b(0, \gamma) = 1 + \gamma$ and $r_b(h, \gamma) = \gamma r_b(h)$ if $h \neq 0$, where $r_b(h)$ is as in Sect. 4. An analogous bound for the weighted star discrepancy of LPSs can be found in [37].

Now we deal with the quantity $R_{b,\boldsymbol{\gamma}}(\boldsymbol{q}, p)$ which can be computed in $O(b^m s)$ operations (see [22, Proposition 10.20]).

Theorem 4. *Let p be irreducible. If $\boldsymbol{q} \in G^s_{b,m}$ is constructed with Algorithm 2, then*

$$R_{b,\boldsymbol{\gamma}}(\boldsymbol{q}, p) \leq \frac{1}{b^m - 1} \prod_{i=1}^{s} \left(1 + \gamma_i \left(1 + m \frac{b^2 - 1}{3b}\right)\right),$$

Algorithm 2 CBC-algorithm for PLPSs

```
Require: $b$ a prime, $s, m \in \mathbb{N}$, $p \in \mathbb{Z}_b[x]$, with $\deg(p) = m$, and weights $\boldsymbol{\gamma} = (\gamma_i)_{i \geq 1}$.
Choose $q_1 = 1$.
for $d = 2$ to $s$ do
    find $q_d \in G_{b,m}^*$ which minimises the quantity $R_{b,\boldsymbol{\gamma}}((q_1, \ldots, q_{d-1}, z), p)$ as a function of $z$.
end for
return $\boldsymbol{q} = (q_1, \ldots, q_s)$.
```

A proof can be found in [18]. A similar result for not necessarily irreducible p has been shown in [14] and a corresponding result for LPSs is [37, Theorem 3].

Using an argument from [19, Sect. 7] one can deduce the following result from Theorem 4; see also [22, Corollary 10.30].

Corollary 1. *Let p be irreducible. If $\sum_{i=0}^{\infty} \gamma_i < \infty$, then for any $\delta > 0$ there exists $c_{\boldsymbol{\gamma},\delta} > 0$, such that for $\boldsymbol{q} \in G_{b,m}^s$ constructed with Algorithm 2 we have*

$$D_{b^m,\boldsymbol{\gamma}}^*(\mathscr{P}(\boldsymbol{q}, p)) \leq c_{\boldsymbol{\gamma},\delta} b^{-m(1-\delta)}.$$

Let $N \in \mathbb{N}$ with 2-adic expansion $N = 2^{m_1} + \cdots + 2^{m_k}$, where $0 \leq m_1 < m_2 < \ldots < m_k$. For $1 \leq j \leq k$ choose $p^{(j)} \in \mathbb{Z}_2[x]$ irreducible with $\deg(p^{(j)}) = m_j$ and construct $\mathscr{P}(\boldsymbol{q}^{(j)}, p^{(j)})$ with Algorithm 2. Then set $\mathscr{P}_N = \mathscr{P}(\boldsymbol{q}^{(1)}, p^{(1)}) \cup \ldots \cup \mathscr{P}(\boldsymbol{q}^{(k)}, p^{(k)})$. In [35] the following is shown:

Corollary 2. *If $\sum_{i=0}^{\infty} \gamma_i < \infty$, then for any $\delta > 0$ there exists $C_{\boldsymbol{\gamma},\delta} > 0$, such that*

$$D_{N,\boldsymbol{\gamma}}^*(\mathscr{P}_N) \leq C_{\boldsymbol{\gamma},\delta} N^{-1+\delta} \textit{ for any } N \in \mathbb{N}.$$

The weighted star discrepancy is strongly polynomial tractable with ε-exponent equal to one.

The cost of the CBC-algorithm is of $O(b^{2m}s^2)$ operations. This is comparable with the CBC construction cost of LPSs; cf. [37, Sect. 3]. However, in this form the CBC-algorithm can only be used for not too large cardinality b^m. A breakthrough for this problem was obtained by Nuyens and Cools [64, 65] when they introduced—first for LPSs and then for PLPSs—the fast CBC construction with a significant reduction of cost to $O(smb^m)$ operations using $O(b^m)$ memory space. Only through this reduction of the construction cost does the CBC-algorithm become applicable for the generation of PLPSs (and of LPSs) with reasonably large cardinality. See also [22, Sect. 10.3].

6 Integration of Walsh Series

As already mentioned in Sect. 3 it is most convenient for the error analysis of polynomial lattice rules to consider Walsh series. Let $\alpha > 1$ and let $\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}$ be the weighted Hilbert function space with reproducing kernel given by

$$K_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \rho_\alpha(\boldsymbol{k},\boldsymbol{\gamma})\, {}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})\,\overline{{}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{y})},$$

where for $\boldsymbol{k} = (k_1,\dots,k_s) \in \mathbb{N}_0^s$ we put $\rho_\alpha(\boldsymbol{k},\boldsymbol{\gamma}) = \prod_{j=1}^s \rho_\alpha(k_j,\gamma_j)$ with $\rho_\alpha(0,\gamma) = 1$ and $\rho_\alpha(k,\gamma) = \gamma b^{-\alpha v}$ if $b^v \le k < b^{v+1}$ for $v \in \mathbb{N}_0$. The norm in this function space is given by

$$\|f\|_{\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}} = \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \rho_\alpha(\boldsymbol{k},\boldsymbol{\gamma})^{-1}|\widehat{f}_{\mathrm{wal}}(\boldsymbol{k})|^2$$

where $\widehat{f}_{\mathrm{wal}}(\boldsymbol{k}) = \int_{[0,1]^s} f(\boldsymbol{x})\overline{{}_b\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}$. For more information on $\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}$ we refer to [20]. The counterpart to the function space $\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}$ for the analysis of LPSs is the so-called Korobov space ([25, 63, 75] or [62, Appendix A.1]) whose reproducing kernel looks similar to $K_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}$ but with the main difference that the Walsh function system is replaced by the trigonometric function system and Walsh coefficients are replaced by Fourier coefficients.

The worst-case integration error of a QMC rule is defined as the worst performance of the QMC algorithm over the unit ball of the function space under consideration, i.e., in our case $e(\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}},\mathscr{P}) := \sup_{\|f\|_{\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}}\le 1} |I_s(f) - Q_{b^m,s}(f)|$. For PLPSs it can be shown that

$$e^2(\boldsymbol{q},p) := e^2(\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}},\mathscr{P}(\boldsymbol{q},p)) = \sum_{\substack{\boldsymbol{k}\in\mathbb{N}_0^s\setminus\{\boldsymbol{0}\}\\ \mathrm{tru}_m(\boldsymbol{k})(x)\in\mathscr{D}_{\boldsymbol{q},p}}} \rho_\alpha(\boldsymbol{k},\boldsymbol{\gamma})$$

where $\mathrm{tru}_m(\boldsymbol{k}) :\equiv \boldsymbol{k} \pmod{b^m}$ (component-wise) and where

$$k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{m-1}b^{m-1} \in \mathbb{N}_0$$

is identified with

$$k(x) = \kappa_0 + \kappa_1 x + \cdots + \kappa_{m-1}x^{m-1} \in \mathbb{Z}_b[x].$$

For the worst-case integration error of a polynomial lattice rule for integration in $\mathscr{H}_{\mathrm{wal},s,\alpha,\boldsymbol{\gamma}}$ we have the following result which was first proved in [16] for irreducible p and later generalized in [41] to not necessarily irreducible p. The corresponding result for LPSs was shown by Korobov [39] for $\boldsymbol{\gamma} = \boldsymbol{1}$ and by Kuo [45] for general weights (see also [10]).

Theorem 5. *For any $p \in \mathbb{Z}_b[x]$ with $\deg(p) = m$ one can construct $\boldsymbol{q} \in G_{b,m}^s$ using a CBC algorithm such that (with $N = b^m$)*

$$e(\boldsymbol{q},p) \le c_{s,\alpha,\boldsymbol{\gamma},\delta} N^{-\alpha/2+\delta} \qquad \textit{for all } 0 < \delta \le \tfrac{\alpha-1}{2}.$$

If $\sum_{i=1}^{\infty} \gamma_i^{1/(\alpha-2\delta)} < \infty$, then $c_{s,\alpha,\boldsymbol{\gamma},\delta} \leq c_{\infty,\alpha,\boldsymbol{\gamma},\delta} < \infty$, i.e., the above bound can be made independent of the dimension s.

7 Extensible Polynomial Lattice Point Sets

A disadvantage of the CBC algorithm as used so far is that the generated vectors $\boldsymbol{q}$ depend on p and hence on $N = b^{\deg(p)}$. If one changes p, then one has to construct a new vector $\boldsymbol{q} \in \mathbb{Z}_b[x]^s$. The same problem appears for the CBC construction of LPSs. For this reason several authors have independently from each other introduced the concept of extensible LPSs, see [32–34, 40, 55]. Niederreiter [59] was the first who considered extensible PLPSs. A special case will be explained below.

For $p \in \mathbb{Z}_b[x]$ with $m = \deg(p) \geq 1$, let Y_p be the set of all p-adic polynomials $\sum_{k=0}^{\infty} a_k p^k$ with $\deg(a_k) < m$. Any $Q \in Y_p$ reduced modulo p^n gives a polynomial in $\mathbb{Z}_b[x]$ of degree less than nm, i.e., $Y_p/(p^n) = G_{b,nm}$. Let $\boldsymbol{Q} \in Y_p^s$ and for $n \in \mathbb{N}$ let $\boldsymbol{q}_n \equiv \boldsymbol{Q} \pmod{p^n}$. Then

$$\mathscr{P}(\boldsymbol{q}_1, p) \subseteq \mathscr{P}(\boldsymbol{q}_2, p^2) \subseteq \mathscr{P}(\boldsymbol{q}_3, p^3) \subseteq \ldots.$$

Definition 6. An *extensible PLPS* is defined as the formal union $\mathscr{P}(\boldsymbol{Q}, p) := \bigcup_{k \geq 1} \mathscr{P}(\boldsymbol{q}_k, p^k)$.

For $\mathscr{P}(\boldsymbol{q}_n, p^n)$ only the first n "digits" in the p-adic expansion of each component of $\boldsymbol{Q}$ are important. This observation is used in the following construction algorithm which uses ideas from Korobov [40] for LPSs.

Algorithm 3 Construction of extensible PLPSs

```
Require: $b$ a prime, $s, m \in \mathbb{N}$, $p \in \mathbb{Z}_b[x]$ monic and irreducible with $\deg(p) = m$, and weights $\boldsymbol{\gamma} = (\gamma_i)_{i \geq 1}$.
Find $\boldsymbol{q}_1 := \boldsymbol{q}$ by minimizing $e^2(\boldsymbol{q}, p)$ over all $\boldsymbol{q} \in G_{b,m}^s$.
for $n = 2, 3, \ldots$ do
    find $\boldsymbol{q}_n := \boldsymbol{q}_{n-1} + p^{n-1}\boldsymbol{q}$ by minimizing $e^2(\boldsymbol{q}_{n-1} + p^{n-1}\boldsymbol{q}, p^n)$ over all $\boldsymbol{q} \in G_{b,m}^s$.
    return $\boldsymbol{q}_n$.
end for
```

Theorem 6. *If $\boldsymbol{q}_n \in G_{b,m}^s$ is constructed according to Algorithm 3, then*

$$e^2(\boldsymbol{q}_n, p^n) \leq c_{s,b,\boldsymbol{\gamma},\alpha} b^{-nm}.$$

If $\sum_{i=1}^{\infty} \gamma_i < \infty$, then $c_{s,\alpha,\boldsymbol{\gamma},\delta} \leq c_{\infty,\alpha,\boldsymbol{\gamma},\delta} < \infty$, i.e., the above bound can be made independent of the dimension s.

A proof of this result and also a corresponding result for LPSs can be found in [61]; see also [22]. A disadvantage of the above error bound is that the worst-case error converges only with order $O(N^{-1/2})$ compared to $O(N^{-\alpha/2+\delta})$ from Theorem 5 for not necessarily extensible PLPSs.

There exists another algorithm—first introduced for LPSs in [23] and then for PLPSs in [11]—which is called CBC sieve algorithm (see [22, Sect. 10.4]) and which yields better error bounds, but with the disadvantage that the generated PLPSs (and LPSs respectively) are only finitely extensible. In this context one also speaks about *embedded PLPSs* (and *embedded LPSs* respectively). For embedded LPSs we also refer to [7]. A pure existence result for extensible PLPSs with small star discrepancy is due to Niederreiter [59]. For existence results for extensible LPSs we refer to Hickernell and Niederreiter [34].

8 Integration in Sobolev Spaces

For $x = x_1 b^{-1} + x_2 b^{-2} + \cdots$ and $\sigma = \sigma_1 b^{-1} + \sigma_2 b^{-2} + \cdots$ with $x_i, \sigma_i \in \{0, \ldots, b-1\}$ the digitally shifted point $y = x \oplus \sigma$ is given by $y = y_1 b^{-1} + y_2 b^{-2} + \cdots$, where $y_i = x_i + \sigma_i \pmod{b}$. For vectors $\boldsymbol{x}$ and $\boldsymbol{\sigma}$ we define the digitally shifted point $\boldsymbol{y} = \boldsymbol{x} \oplus \boldsymbol{\sigma}$ component-wise. This digital shift can be used to randomize a PLPS.

Definition 7. For $\boldsymbol{\sigma} \in [0,1)^s$ the point set $\mathscr{P}_{\boldsymbol{\sigma}}(\boldsymbol{q}, p) := \mathscr{P}(\boldsymbol{q}, p) \oplus \boldsymbol{\sigma}$ is called a *digitally shifted PLPS*.

In the context of LPSs one often uses a “geometric” shift instead of the digital shift to randomize the point set and speaks then about shifted LPSs.

Similar results to those from Sect. 6 hold for the mean square worst-case error of digitally shifted polynomial lattices for integration in the Sobolev space $\mathscr{H}^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}$ with reproducing kernel

$$K^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{s} \left(1 + \gamma_i B_1(x_i) B_1(y_i) + \frac{\gamma_i}{2} B_2(|x_i - y_i|)\right),$$

where B_i is the ith Bernoulli polynomial. The function space $\mathscr{H}^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}$ contains all functions $f : [0,1]^s \to \mathbb{R}$ whose mixed partial derivatives up to order one in each variable are square integrable. See [24, 76] and [62, Appendix A.2.3.] for more information on $\mathscr{H}^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}$.

The *mean square worst-case error* of digitally shifted PLPSs for integration in $\mathscr{H}^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}$ is defined by

$$\widehat{e}^2(\boldsymbol{q}, p) = \int_{[0,1]^s} e^2(\mathscr{H}^{(1)}_{\mathrm{sob},s,\boldsymbol{\gamma}}, \mathscr{P}_{\boldsymbol{\sigma}}(\boldsymbol{q}, p))\, \mathrm{d}\boldsymbol{\sigma}.$$

We have the following result the proof of which can be found in [22, Theorem 12.14]; see also [16]. The corresponding result for shifted LPSs was shown by Kuo [45] (and follows in the case $\boldsymbol{\gamma} = \mathbf{1}$ also from [39]).

Theorem 7. *For any $p \in \mathbb{Z}_b[x]$ with $\deg(p) = m$ we can construct $\boldsymbol{q} \in G_{b,m}^s$ using a CBC algorithm such that (with $N = b^m$)*

$$\widehat{e}(\boldsymbol{q}, p) \leq c_{s,b,\boldsymbol{\gamma},\varepsilon} N^{-1+\varepsilon} \quad \textit{for all } 0 < \varepsilon \leq 1/2.$$

If $\sum_{i=1}^s \gamma_i^{1/(2(1-\varepsilon))} < \infty$, then $c_{s,b,\boldsymbol{\gamma},\varepsilon} \leq c_{\infty,b,\boldsymbol{\gamma},\varepsilon} < \infty$, i.e., the above bound can be made independent of the dimension s.

Remark 2. Baldeaux and Dick [1] showed that in the *randomized setting* one can obtain an improved error bound by using Owen's scrambling (see [66] or [22, Chap. 13]). For scrambled PLPSs one has

$$\mathbb{E}\left[|I_s(f) - Q_{N,s}(f)|^2\right] \leq c_{s,b,\boldsymbol{\gamma},\varepsilon} N^{-3+\varepsilon} \quad \text{for } \varepsilon > 0$$

where $N = b^m$ and where the expectation is with respect to all random scramblings of a PLPS. Such a result is not known for LPSs.

Now we assume more smoothness for integrands. Consider the Sobolev space $\mathscr{H}_{\mathrm{sob},s,\boldsymbol{\gamma}}^{(2)}$ with reproducing kernel

$$K_{\mathrm{sob},s,\boldsymbol{\gamma}}^{(2)}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{s} \left(1 + \gamma_i B_1(x_i) B_1(y_i) + \frac{\gamma_i^2}{4} B_2(x_i) B_2(y_i) - \frac{\gamma_i^2}{24} B_4(|x_i - y_i|)\right),$$

where B_i is the ith Bernoulli polynomial. The function space $\mathscr{H}_{\mathrm{sob},s,\boldsymbol{\gamma}}^{(2)}$ contains all functions $f : [0,1]^s \to \mathbb{R}$ whose mixed partial derivatives up to order two in each variable are square integrable. See [22, Sect. 14.6] for more information.

Using an idea from Hickernell [31] we use the tent transformation $\phi(x) = 1 - |2x - 1|$. For vectors $\boldsymbol{x}$ we apply ϕ component-wise and for a point set $\mathscr{P}$, $\phi(\mathscr{P})$ means that the tent transformation is applied to every element of $\mathscr{P}$. We call $\phi(\mathscr{P})$ the *folded* point set $\mathscr{P}$. Define the mean square worst-case error of folded digitally shifted PLPSs by

$$\widehat{e}_{\phi}^2(\boldsymbol{q}, p) = \int_{[0,1]^s} e^2(\mathscr{H}_{\mathrm{sob},s,\boldsymbol{\gamma}}^{(2)}, \phi(\mathscr{P}_{\boldsymbol{\sigma}}(\boldsymbol{q}, p)))\, \mathrm{d}\boldsymbol{\sigma}.$$

The following result, proved in [9], shows that one can obtain an improved convergence rate for the mean square worst-case error of folded digitally shifted PLPSs for functions $f \in \mathscr{H}_{\mathrm{sob},s,\boldsymbol{\gamma}}^{(2)}$ as integrands. A corresponding result for LPSs has been shown by Hickernell [31].

Theorem 8. *For any $p \in \mathbb{Z}_2[x]$ with $\deg(p) = m$ we can construct $\boldsymbol{q} \in G_{2,m}^s$ using a CBC algorithm such that (with $N = 2^m$)*

$$\widehat{e}_\phi(\boldsymbol{q}, p) \leq c_{s,\boldsymbol{\gamma},\varepsilon} N^{-2+\varepsilon} \quad \textit{for all } 0 < \varepsilon \leq 3/2.$$

If $\sum_{i=1}^s \gamma_i^{1/(2(2-\varepsilon))} < \infty$, then $c_{s,\boldsymbol{\gamma},\varepsilon} \leq c_{\infty,\boldsymbol{\gamma},\varepsilon} < \infty$, i.e., the above bound can be made independent of the dimension s.

9 Higher Order Polynomial Lattice Rules

Now we go a step further and consider functions with arbitrary smoothness as integrands. For a more detailed definition of the function spaces under consideration we need some notation:

For $k = \kappa_1 b^{a_1-1} + \kappa_2 b^{a_2-1} + \cdots + \kappa_v b^{a_v-1}$, where $1 \leq a_v < \cdots < a_1$, $v \in \mathbb{N}$ and $\kappa_1, \ldots, \kappa_v \in \{1, \ldots, b-1\}$, and for $\alpha \geq 1$ define

$$\mu_\alpha(k) := a_1 + \cdots + a_{\min(v,\alpha)}.$$

Furthermore, for $\gamma > 0$ put $r_\alpha(0, \gamma) = 1$ and $r_\alpha(k, \gamma) = \gamma b^{-\mu_\alpha(k)}$ for $k \in \mathbb{N}$. For $\boldsymbol{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots)$, set $r_\alpha(\boldsymbol{k}, \boldsymbol{\gamma}) := \prod_{i=1}^s r_\alpha(k_i, \gamma_i)$.

Let $\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}} \subseteq L_2([0,1]^s)$ be the space consisting of all Walsh series $f = \sum_{\boldsymbol{k} \in \mathbb{N}_0^s} \widehat{f}_{\mathrm{wal}}(\boldsymbol{k})\, {}_b\mathrm{wal}_{\boldsymbol{k}}$ for which

$$\|f\|_{\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}}} := \sup_{\boldsymbol{k} \in \mathbb{N}_0^s} \frac{|\widehat{f}_{\mathrm{wal}}(\boldsymbol{k})|}{r_\alpha(\boldsymbol{k}, \boldsymbol{\gamma})} < \infty.$$

For $\alpha \geq 2$ the function space $\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}}$ contains all functions $f : [0,1]^s \to \mathbb{R}$ whose mixed partial derivatives up to order α in each variable are square integrable; see [12]. We call α the *smoothness parameter* of the function space.

Of course one would expect that the higher smoothness of integrands is reflected in the convergence rate of the integration error. Higher smoothness should lead to improved convergence rates. However, it turns out that this is *not* the case when the concept of (digitally shifted) PLPSs, as introduced in Definition 3, is used as underlying nodes. For this reason the following suitable generalization has been introduced in [21]; see also [22, Sect. 15.7].

Definition 8. For $s, m, n \in \mathbb{N}$, $m \leq n$, $p \in \mathbb{Z}_b[x]$, with $\deg(p) = n$, and $\boldsymbol{q} \in \mathbb{Z}_b[x]^s$ the point set $\mathscr{P}_{m,n}(\boldsymbol{q}, p)$ consisting of the b^m points

$$\boldsymbol{x}_h = v_n\left(\frac{h(x)}{p(x)}\boldsymbol{q}(x)\right) \quad \text{for all } h \in G_{b,m}$$

is called a *higher order polynomial lattice point set (HOPLPS)*. A QMC rule using $\mathscr{P}_{m,n}(\boldsymbol{q}, p)$ is called a *higher order polynomial lattice rule*.

Remark 3. For $m = n$ we have $\mathscr{P}_{m,m}(\boldsymbol{q}, p) = \mathscr{P}(\boldsymbol{q}, p)$.

Definition 9. The *dual net* of the HOPLPS $\mathscr{P}_{m,n}(\boldsymbol{q}, p)$ from Definition 8 is defined as

$$\mathscr{D}_{\boldsymbol{q},p} = \{\boldsymbol{k} \in G_{b,n}^s \,:\, \boldsymbol{k} \cdot \boldsymbol{q} \equiv u \pmod{p} \text{ with } \deg(u) < n - m\}.$$

Similar as in Sect. 4 one can introduce a generalization of the quality measure ρ for HOPLPSs which can then be related to the worst-case integration error of HOPLPSs. This was done in [15] (see also [22, Definition 15.27]). Instead of following this track here we study the worst-case error of HOPLPSs in $\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}}$ more directly.

For $\alpha \ge 2$ the worst-case error for integration in $\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}}$ using $\mathscr{P}_{m,n}(\boldsymbol{q}, p)$ is given by (see [2, Proposition 2.1])

$$e_\alpha^2(\boldsymbol{q}, p) := e_\alpha^2(\mathscr{W}_{\alpha,s,\boldsymbol{\gamma}}, \mathscr{P}_{m,n}(\boldsymbol{q}, p)) = \sum_{\substack{\boldsymbol{k} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\} \\ \mathrm{tru}_n(\boldsymbol{k})(x) \in \mathscr{D}_{\boldsymbol{q},p}}} r_\alpha(\boldsymbol{k}, \boldsymbol{\gamma}).$$

The following result has been shown in [2].

Theorem 9. *For any irreducible $p \in \mathbb{Z}_b[x]$ with $\deg(p) = n$ we can construct $\boldsymbol{q} \in G_{b,n}^s$ using a CBC algorithm such that*

$$e_\alpha(\boldsymbol{q}, p) \le c_{s,\alpha,\boldsymbol{\gamma},\tau} b^{-\min(\tau m, n)} \quad \textit{for all } 1 \le \tau < \alpha.$$

If $\sum_{i=1}^\infty \gamma_i^{1/\tau} < \infty$ then $c_{s,\alpha,\boldsymbol{\gamma},\tau} \le c_{\infty,\alpha,\boldsymbol{\gamma},\tau} < \infty$, i.e., the above bound can be made independent of the dimension s.

Remark 4. Choosing n large we obtain a convergence order of $N^{-\alpha+\varepsilon}$ for $\varepsilon > 0$ where $N = b^m$. By a result of Šarygin [69] this convergence rate is essentially best possible. For a fast version of the CBC algorithm mentioned in Theorem 9 we refer to [4].

The result from Theorem 9 holds for a fixed smoothness parameter $\alpha \ge 2$. However, in practical applications the smoothness parameter is in general not known a priori. Hence it is reasonable to ask for constructions of HOPLPSs which achieve almost optimal convergence rates for a range of smoothness parameters and which adjust themselves to the smoothness of a given integrand.

The basic idea in [2] can be roughly explained as follows. Assume that $p \in \mathbb{Z}_b[x]$ is given. If there exists a large enough amount of HOPLPSs $\mathscr{P}(\boldsymbol{q}, p)$ which perform well for the smoothness parameter α and if there exists a large enough amount of HOPLPSs $\mathscr{P}(\boldsymbol{q}, p)$ which perform well for the smoothness parameter α', then there must be a HOPLPS $\mathscr{P}(\boldsymbol{q}, p)$ which performs well for both smoothness parameters α and α'. The underlying mathematical argument is the following “sieve principle”: let X be some finite set and $A, B \subseteq X$. If $|A|, |B| > |X|/2$, then $|A \cap B| > 0$.

Algorithm 4 Sieve Algorithm for HOPLPSs

```
Require: $b$ a prime, $s, m, \beta \in \mathbb{N}$, $\beta \geq 2$, $p \in \mathbb{Z}_b[x]$ irreducible with $\deg(p) = m$, weights $\boldsymbol{\gamma} = (\gamma_i)_{i \geq 1}$,
Set $n = \beta m$.
Find $\lfloor (1-\beta^{-1}) b^{\beta m s} \rfloor + 1$ vectors $\boldsymbol{q}$ in $G^s_{b,\beta m}$ which satisfy
    $$e_2(\boldsymbol{q}, p) \leq c_{s,b,\boldsymbol{\gamma},m,\beta,2,\tau_2} b^{-\tau_2 m} \quad \text{for all } 1 \leq \tau_2 < 2,$$
  and label this set $\mathscr{T}_2$.
for $\alpha = 3, \ldots, \beta$ do
    find $\lfloor (1-(\alpha-1)\beta^{-1}) b^{\beta m s} \rfloor + 1$ vectors $\boldsymbol{q}$ in $\mathscr{T}_{\alpha-1}$ which satisfy
        $$e_\alpha(\boldsymbol{q}, p) \leq c_{s,b,\boldsymbol{\gamma},m,\beta,\alpha,\tau_\alpha} b^{-\tau_\alpha m} \quad \text{for all } 1 \leq \tau_\alpha < \alpha$$
    and label this set $\mathscr{T}_\alpha$.
end for
return Select $\boldsymbol{q}^*$ to be any vector from $\mathscr{T}_\beta$.
```

Algorithm 4 only presents the basic idea of a construction for HOPLPS which perform well for a range of smoothness parameters. In practice this algorithm would not be applicable since it is much too time consuming. However, in [2, Algorithm 2] it has been show how Algorithm 4 can be combined with the CBC approach. This leads then to the following result which is [2, Theorem 4.2]:

Theorem 10. *Let $s, m, \beta \in \mathbb{N}$, $\beta \geq 2$, then one can construct a vector $\boldsymbol{q} \in G^s_{b,\beta m}$ such that*

$$e_\alpha(\boldsymbol{q}, p) \leq c_{s,b,\alpha,\beta,\boldsymbol{\gamma},\tau_\alpha} b^{-\tau_\alpha m} \quad \textit{for all } 1 \leq \tau_\alpha < \alpha$$

and for all $2 \leq \alpha \leq \beta$.

If $\sum_{i=1}^{\infty} \gamma_i^{1/\tau_\alpha} < \infty$, then $c_{s,b,\alpha,\beta,\boldsymbol{\gamma},\tau_\alpha} \leq c_{\infty,b,\alpha,\beta,\boldsymbol{\gamma},\tau_\alpha} < \infty$, i.e., the above bound can be made independent of the dimension s.

There exists no counterpart of the results from this section for LPSs.

10 Summary and Further Comments

In this paper we have reviewed the main progress in the analysis of PLPSs over the last decade and we pointed out several connections to the theory of LPSs.

For both concepts we have comparable discrepancy bounds and tractability properties, and the worst-case error analysis in several reproducing kernel Hilbert spaces follows parallel tracks. PLPSs and LPSs can both be constructed with the (fast) CBC approach and both can be made extensible in the number of elements. The tent transformation together with a suitable randomization leads in both cases to improved error bounds for smoother integrands.

However, there are also some differences. For example, with a slight generalization of the concept of PLPSs one can achieve almost optimal convergence rates for smooth integrands (even with varying smoothness from a finite range) together with strong tractability, which means that the error bound is independent of the dimension. Such a result is not known for LPSs until now. (But it is known that with LPSs one can obtain almost optimal convergence rates together with strong tractability for smooth *periodic* functions from a Korobov space.)

A further difference is that for PLPSs it makes sense to apply Owen's scrambling scheme since this preserves the (t, m, s)-net structure of a point set but not the geometric lattice structure. This leads to an improved error bound in the randomized setting, a result which is not known for LPSs.

Also the consideration of the quality parameter t of LPSs makes in general little sense since these point sets are not constructed to have a good (t, m, s)-net structure. Nevertheless, the analog of the quality measure $\rho(\boldsymbol{q}, p) = m - t$ from Sect. 4 has some interpretation, namely it is the enhanced trigonometric degree of a lattice rule [8, 54]. A cubature rule of enhanced trigonometric degree δ is one that integrates all trigonometric polynomials of degree less then δ exactly. However, in this vein $\rho(\boldsymbol{q}, p) = m - t$ from Sect. 4 can also be interpreted as the enhanced Walsh degree of a polynomial lattice rule since any (t, m, s)-net in base b integrates all Walsh polynomials of degree $\leq m - t$ exactly (this follows from [30, Lemma 1]).

A further point which was not discussed so far but which is worth to be mentioned is that with LPSs one can even obtain exponential convergence for the worst-case error of infinitely times differentiable periodic functions; see [17]. This should also be possible with PLPSs.

LPSs and PLPSs can also be applied for the problem of function approximation. More information in this direction can be found in [46, 47] for LPSs and in [3, 13] for PLPSs.

We close this paper with an outlook to more general constructions: a more general form of LPSs as given in Definition 2 is the concept of *integration lattices* which are presented in [58, Sect. 5.3] and in [72]. An integration lattice is a discrete subset of $\mathbb{R}^s$ which is closed under addition and subtraction, and which contains $\mathbb{Z}^s$ as a subset. In the same vein Lemieux and L'Ecuyer [52, 53] introduced so-called *polynomial integration lattices* which generalize the concept of PLPSs from Definition 3. Results on the star discrepancy and the t-parameter of such point sets can be found in [29].

A very general construction of point sets in $[0, 1)^s$ for which PLPSs serve as special cases is the concept of cyclic nets due to Niederreiter [60] and, even more general, of hyperplane nets due to Pirsic et al. [67]. Cyclic and hyperplane nets are constructions of digital (t, m, s)-nets which are inspired by a close connection between coding theory and the theory of digital nets. In fact, the cyclic net construction is the analog to the construction of so-called cyclic codes which are well known in coding theory. For more information we refer to [22, Chap. 11] and the references therein.

Acknowledgements The author is partially supported by the Austrian Science Foundation (FWF), Project S9609, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory". The author also thanks Josef Dick and Peter Kritzer for many remarks and suggestions.

References

1. Baldeaux, J. and Dick, J.: A construction of polynomial lattice rules with small gain coefficients. Numer. Math. 119: 271–297, 2011.
2. Baldeaux, J., Dick, J., Greslehner, J. and Pillichshammer, F.: Construction algorithms for higher order polynomial lattice rules. J. Complexity 27: 281–299, 2011.
3. Baldeaux, J., Dick, J. and Kritzer, P.: On the approximation of smooth functions using generalized digital nets. J. Complexity 25: 544–567, 2009.
4. Baldeaux, J., Dick, J., Leobacher, G., Nuyens, D. and Pillichshammer, F.: Efficient calculation of the worst-case error and (fast) component-by-component construction of higher order polynomial lattice rules. Numer. Algorithms 59: 403–431, 2012.
5. Béjian, R.: Minoration de la discrépance d'une suite quelconque sur T. Acta Arith. 41: 185–202, 1982. (French)
6. Bilyk, D., Lacey, M.T., and Vagharshakyan, A.: On the small ball inequality in all dimensions. J. Funct. Anal. 254: 2470–2502, 2008.
7. Cools, R., Kuo, F.Y. and Nuyens, D.: Constructing embedded lattice rules for multivariable integration. SIAM J. Sci. Comput. 28: 2162–2188, 2006.
8. Cools, R. and Lyness, J.N.: Three- and four-dimensional K-optimal lattice rules of moderate trigonometric degree. Math. Comp. 70: 1549–1567, 2001.
9. Cristea, L.L., Dick, J., Leobacher, G. and Pillichshammer, F.: The tent transformation can improve the convergence rate of quasi-Monte Carlo algorithms using digital nets. Numer. Math. 105: 413–455, 2007.
10. Dick, J.: On the convergence rate of the component-by-component construction of good lattice rules. J. Complexity 20: 493–522, 2004.
11. Dick, J.: The construction of extensible polynomial lattice rules with small weighted star discrepancy. Math. Comp. 76: 2077–2085, 2007.
12. Dick, J.: Walsh spaces containing smooth functions and quasi-Monte Carlo rules of arbitrary high order. SIAM J. Numer. Anal. 46: 1519–1553, 2008.
13. Dick, J., Kritzer, P., and Kuo, F.Y.: Approximation of functions using digital nets. In: *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 275–297, Springer, Berlin, 2008.
14. 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.
15. Dick, J., Kritzer, P., Pillichshammer, F. and Schmid, W. Ch.: On the existence of higher order polynomial lattices based on a generalized figure of merit. J. Complexity 23: 581–593, 2007.
16. Dick, J., Kuo, F.Y., Pillichshammer, F. and Sloan, I.H.: Construction algorithms for polynomial lattice rules for multivariate integration. Math. Comp. 74: 1895–1921, 2005.
17. Dick, J., Larcher, G., Pillichshammer, F. and Woźniakowski, H.: Exponential convergence and tractability of multivariate integration for Korobov spaces. Math. Comp. 80: 905–930, 2011.
18. 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.
19. 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.
20. Dick, J. and Pillichshammer, F.: Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces. J. Complexity 21: 149–195, 2005.

21. Dick, J. and Pillichshammer, F.: Strong tractability of multivariate integration of arbitrary high order using digitally shifted polynomial lattices rules. J. Complexity 23: 436–453, 2007.
22. Dick, J. and Pillichshammer, F.: *Digital Nets and Sequences. Discrepancy Theory and Quasi-Monte Carlo Integration.* Cambridge University Press, Cambridge, 2010.
23. Dick, J., Pillichshammer, F. and Waterhouse, B.J.: The construction of good extensible rank-1 lattices. Math. Comp. 77: 2345–2373, 2008.
24. Dick, J., Sloan, I.H., Wang, X. and Woźniakowski, H.: Liberating the weights. J. Complexity 20: 593–623, 2004.
25. Dick, J., Sloan, I.H., Wang, X. and Woźniakowski, H.: Good lattice rules in weighted Korobov spaces with general weights. Numer. Math. 103: 63–97, 2006.
26. Drmota, M. and Tichy, R.F.: *Sequences, Discrepancies and Applications*. Lecture Notes in Mathematics 1651, Springer, Berlin, 1997.
27. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. 41:337–351, 1982. (French).
28. Gnewuch, M., Srivastav, A. and Winzen, C.: Finding optimal volume subintervals with k-points and calculating the star discrepancy are NP-hard problems. J. Complexity 25: 115–127, 2009.
29. Greslehner, J. and Pillichshammer, F.: Discrepancy of higher rank polynomial lattice point sets. Monte Carlo Methods Appl. 18: 79–108, 2012.
30. Hellekalek, P.: Digital (t, m, s)-nets and the spectral test. Acta Arith. 105: 197–204, 2002.
31. Hickernell, F.J.: Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In: *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 274–289, Springer, Berlin, 2002.
32. Hickernell, F.J. and Hong, H.S.: Computing multivariate normal probabilities using rank-1 lattice sequences. In: *Proceedings of the Workshop on Scientific Computing (Hong Kong)*, pages 209–215, Springer, Singapore, 1997.
33. Hickernell, F.J., Hong, H.S., L'Ecuyer, P. and Lemieux, C.: Extensible lattice sequences for quasi-Monte Carlo quadrature. SIAM J. Sci. Coput. 22: 1117-1138, 2000.
34. Hickernell, F.J. and Niederreiter, H.: The existence of good extensible rank-1 lattices. J. Complexity 19: 286–300, 2003.
35. Hinrichs, A., Pillichshammer, F. and Schmid, W. Ch.: Tractability properties of the weighted star discrepancy. J. Complexity 24: 134–143, 2008.
36. Hlawka, E.: Zur angenäherten Berechnung mehrfacher Integrale. Monatsh. Math. 66: 140–151, 1962. (German)
37. 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.
38. Korobov, N.M.: The approximate computation of multiple integrals. Dokl. Akad. Nauk SSSR 124: 1207–1210, 1959. (Russian)
39. Korobov, N.M.: *Number-theoretic methods in approximate analysis.* Gosudarstv. Izdat. Fiz.-Mat. Lit., Moscow, 1963. (Russian)
40. Korobov, N.M.: On the calculation of optimal coefficients. Soviet Math. Dokl. 26: 590–593, 1982.
41. Kritzer, P. and Pillichshammer, F.: Constructions of general polynomial lattices for multivariate integration. Bull. Austral. Math. Soc. 76: 93–110, 2007.
42. Kritzer, P. and Pillichshammer, F.: A lower bound on a quantity related to the quality of polynomial lattices. Funct. Approx. Comment. Math. 45: 125–137, 2011
43. Kritzer, P. and Pillichshammer, F.: Low discrepancy polynomial lattice point sets. Submitted, 2011.
44. Kuipers, L. and Niederreiter, H.: *Uniform Distribution of Sequences.* John Wiley, New York, 1974; reprint, Dover Publications, Mineola, NY, 2006.
45. Kuo, F. Y.: Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. J. Complexity 19: 301–320, 2003.
46. Kuo, F.Y., Sloan, I.H. and Woźniakowski, H.: Lattice rules for multivariate approximation in the worst case setting. In: *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 289–330, Springer, Berlin, 2006.

47. Kuo, F.Y., Sloan, I.H. and Woźniakowski, H.: Lattice rule algorithms for multivariate approximation in the average case setting. J. Complexity 24: 283–323, 2008.
48. Larcher, G.: On the distribution of sequences connected with good lattice points. Monatsh. Math. 101: 135–150, 1986.
49. Larcher, G.: A best lower bound for good lattice points. Monatsh. Math. 104: 45–51, 1987.
50. Larcher, G.: Nets obtained from rational functions over finite fields. Acta Arith. 63: 1–13, 1993.
51. Larcher, G., Lauss, A., Niederreiter, H. and Schmid, W. Ch.: Optimal polynomials for (t,m,s)-nets and numerical integration of multivariate Walsh series. SIAM J. Numer. Anal. 33: 2239–2253, 1996.
52. L'Ecuyer, P.: Polynomial integration lattices. In: *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 73–98, Springer, Berlin, 2004.
53. Lemieux, Ch. and L'Ecuyer P.: Randomized polynomial lattice rules for multivariate integration and simulation. SIAM J. Sci. Comput. 24: 1768–1789, 2003.
54. Lyness, J.: Notes on lattice rules. J. Complexity 19: 321–331, 2003.
55. Maize, E.H.: *Contributions to the Theory of Error Reduction of Quasi Monte Carlo Methods.* Thesis (Ph.D.)The Claremont Graduate University, 162 pp., 1981.
56. Niederreiter, H.: Point sets and sequences with small discrepancy. Monatsh. Math. 104: 273–337, 1987.
57. Niederreiter, H.: Low-discrepancy point sets obtained by digital constructions over finite fields. Czechoslovak Math. J. 42: 143–166, 1992.
58. Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods.* No. 63 in CBMS-NSF Series in Applied Mathematics. SIAM, Philadelphia, 1992.
59. Niederreiter, H.: The existence of good extensible polynomial lattice rules. Monatsh. Math. 139: 295–307, 2003.
60. Niederreiter, H.: Digital nets and coding theory. In: *Coding, Cryptography and Combinatorics*, pages 247–257, Birkhäuser, Basel, 2004.
61. Niederreiter, H. and Pillichshammer, F.: Construction algorithms for good extensible lattice rules. Constr. Approx. 30: 361–393, 2009.
62. Novak, E. and Woźniakowski, H.: *Tractability of Multivariate Problems. Volume I: Linear Information.* EMS Tracts in Mathematics, 6. European Mathematical Society (EMS), Zürich, 2008.
63. Novak, E. and Woźniakowski, H.: *Tractability of Multivariate Problems. Volume II: Standard Information for Functionals.* EMS Tracts in Mathematics, 12. European Mathematical Society (EMS), Zürich, 2010.
64. Nuyens, D. and 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.
65. Nuyens, D. and Cools, R.: Fast component-by-component construction, a reprise for different kernels. In: *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 373–387, Springer, Berlin, 2006.
66. Owen, A.B.: Randomly permuted (t,m,s)-nets and (t,s)-sequences. In: *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 299–317, Springer, New York, 1995.
67. 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.
68. Roth, K.F.: On irregularities of distribution. Mathematika 1: 73–79, 1954.
69. Šarygin, I.F.: Lower bounds for the error of quadrature formulas on classes of functions. Ž. Vyčisl. Mat. i Mat. Fiz. 3: 370–376, 1963. (Russian).
70. Schmid, W. Ch.: Improvements and extensions of the "Salzburg Tables" by using irreducible polynomials. In: *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 436–447, Springer, Berlin, 2000.
71. Schmidt, W.M.: Irregularities of distribution VII. Acta Arith. 21: 45–50, 1972.
72. Sloan, I.H. and Joe, S.: *Lattice Methods for Multiple Integration.* Clarendon Press, Oxford, 1994.

73. Sloan, I.H. and Reztsov, A.V.: Component-by-component construction of good lattice rules. Math. Comp. 71: 263–273, 2002.
74. Sloan, I.H. and Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? J. Complexity 14: 1–33, 1998.
75. Sloan, I.H. and Woźniakowski, H.: Tractability of multivariate integration for weighted Korobov classes. J. Complexity 17: 697–721, 2001.
76. Sloan, I.H. and Woźniakowski, H.: Tractability of integration in non-periodic and periodic weighted tensor product Hilbert spaces. J. Complexity 18: 479–499, 2002.
77. Sobol', I.M.: Distribution of points in a cube and approximate evaluation of integrals. Ž. Vyčisl. Mat. i Mat. Fiz. 7: 784–802, 1967. (Russian)

Liberating the Dimension for Function Approximation and Integration

G.W. Wasilkowski

Abstract We discuss recent results on the complexity and tractability of problems dealing with ∞-variate functions. Such problems, especially *path integrals*, arise in many areas including mathematical finance, quantum physics and chemistry, and stochastic differential equations. It is possible to replace the ∞-variate problem by one that has only d variables since the difference between the two problems diminishes with d approaching infinity. Therefore, one could use algorithms obtained in the Information-Based Complexity study, where problems with arbitrarily large but fixed d have been analyzed. However, to get the optimal results, the choice of a specific value of d should be a part of an efficient algorithm. This is why the approach discussed in the present paper is called *liberating the dimension.* Such a choice should depend on the cost of sampling d-variate functions and on the error demand ε. Actually, as recently observed for a specific class of problems, optimal algorithms are from a family of *changing dimension* algorithms which approximate ∞-variate functions by a combination of special functions, each depending on a different set of variables. Moreover, each such set contains no more than $d(\varepsilon) = \mathscr{O}(\ln(1/\varepsilon)/\ln(\ln(1/\varepsilon)))$ variables. This is why the new algorithms have the total cost polynomial in $1/\varepsilon$ even if the cost of sampling a d-variate function is exponential in d.

1 Introduction

We discuss some recent results on computational problems dealing with functions of infinitely many variables, which are called ∞-*variate functions*. Such problems arise in many areas including mathematical finance, quantum physics and chemistry,

G.W. Wasilkowski
Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA
e-mail: greg@cs.uky.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_9, © Springer-Verlag Berlin Heidelberg 2012

and in solving deterministic and stochastic differential equations. The main source of such problems is probably given by *path integrals*. A partial list of references include [3–7, 12–14, 17, 36]. Actually, all problems involving expectations of stochastic processes $\mathbf{X}(t)$ can be viewed as integration problems for ∞-variate functions, i.e., path integration. Indeed, consider the expectation $\mathbb{E}(V(\mathbf{X}(t)))$ for a function V and a Gaussian process $\mathbf{X}$, e.g., Brownian motion. Due to the Karhunen-Loéve expansion, $\mathbf{X}(t) = \sum_{j=1}^{\infty} x_j \cdot g_j(t)$ for i.i.d. $\mathcal{N}(0,1)$ random variables x_j and some functions g_j, the expectation is the integral of the ∞-variate function

$$f(x_1, x_2, \dots) := V\left(\sum_{j=1}^{\infty} x_j \cdot g_j(t)\right)$$

with respect to the probability density functions $\rho(x_i) = e^{-x_i^2/2}/\sqrt{2\pi}$.

One of the main tools used so far in practice is a variant of the Monte Carlo algorithm; however, it may be slow. Since typical ∞-variate functions could be approximated by functions with finite but sufficiently large number d of variables, the volume of results from *Information-Based Complexity* (*IBC* for short) could be applied, see, e.g., [21]. However, we believe that such an approach to ∞-variate problems would not yield the most efficient algorithms.

Indeed, the majority of IBC papers on the complexity of multivariate problems consider spaces of functions with d variables for finite yet arbitrarily large d, see again [21] and papers cited there. A typical question addressed in these papers is: How does the cost depend on the error demand ε and d? There are many positive results. However, since d may be arbitrarily large independently of ε, there are also many negative results.

We are convinced that when dealing with ∞-variate problems

the selection of d should be a part of efficient algorithms

and, in particular, should depend on the cost of sampling d-variate functions which is is denoted here by $\$(d)$. For instance, sampling a d-variate polynomial of degree 2 requires $\$(d) = \mathcal{O}(d^2)$ arithmetic operations, whereas sampling polynomials of degree 10 is more expensive, $\$(d) = \mathcal{O}(d^{10})$. When simulating the Brownian path $\mathbf{X}(t)$, the Karhunen-Loéve expansion is usually truncated. For instance, we may have

$$\mathbf{X}(t; x_1, x_2, \dots) \sim \sqrt{2/\pi} \sum_{j=1}^{d} x_j \, \frac{\sin((j-1/2)\,\pi\, t/T)}{j-1/2}.$$

Hence, again, the cost depends on d and it would be reasonable to take $\$(d) = \mathcal{O}(d)$ in this case.

Equally importantly,

the value of d should depend on the error demand ε.

More precisely, $d = d(\varepsilon)$ should be a function of ε so that the cost of computing an ε-approximation to the original ∞-variate problem is minimized. As we shall see later, for some problems, $d(\varepsilon)$ increases surprisingly slowly with decreasing ε.

This point is important and shows the difference between the study of ∞-variate problems and the study of tractability of multivariate problems. For ∞-variate problems, we are interested in algorithms that have good properties (e.g., small cost) *only* for the pairs

$$(\varepsilon, d(\varepsilon)) \quad \text{for} \quad \varepsilon \in (0,1),$$

whereas for multivariate problems, good properties should hold for *all* the pairs

$$(\varepsilon, d) \quad \text{for} \quad \varepsilon \in (0,1) \quad \text{and} \quad d = 1, 2, \dots .$$

This is why there are problems with negative tractability results if all pairs (ε, d) are considered and positive results if only pairs $(\varepsilon, d(\varepsilon))$ are of interest.

Such an approach for ∞-variate functions was considered in [24, 31] for approximating Feynman-Kac type of integrals, and more recently in [2, 8–10, 16, 18, 19, 23] for approximating more general integrals, as well as in [33, 34] for function approximation. The presentation of this paper is based on results from [16, 23, 33, 34].

As in the four papers mentioned above, the functions to be integrated or approximated belong to a *quasi-reproducing kernel Hilbert space* (or *Q-RKH space* for short). This means that function evaluation may be a discontinuous functional for some sampling points. We restrict the attention to the *changing dimension algorithms* (or *CD algorithms* for short) introduced in [16] since they provide optimal results, modulo logarithmic terms, with sharp bounds on the tractability exponents. The CD-algorithms approximate only the most important terms from a special Fourier expansion of the function being approximated. Each term depends on a different set of variables. Quite surprisingly, each set contains at most

$$\mathscr{O}(\ln(1/\varepsilon)/\ln(\ln(1/\varepsilon)))$$

variables. This allows efficient algorithms even when the cost function $\$(d)$ is exponential in d.

The approach of using optimal algorithms for approximating the original ∞-variate problem without pre-specifying the value of d is what we call

liberating the dimension.

2 Basic Concepts

In this section, we recall basic definitions/concepts used in the paper. For more detailed discussions, we refer to [16, 23, 33, 34].

2.1 Quasi-reproducing Kernel Hilbert Spaces

We follow here the model introduced in [16] and extended in [33]. The spaces $\mathscr{F}_{\boldsymbol{\gamma}}$ of ∞-variate functions are defined as weighted sums of tensor products of a space of univariate real functions. More precisely, for a Borel measurable set $D \subseteq \mathbb{R}$, let F be a separable reproducing kernel Hilbert space (or RKH space for short) of real functions with the domain D whose kernel is denoted by K.[1] To omit the trivial case, we always assume that $K \neq 0$. To stress that F is generated by K, we will often write

$$F = H(K).$$

We will assume throughout the paper that

$$1 \notin F, \tag{1}$$

where 1 denotes the constant function $f \equiv 1$. When the information used by algorithms is restricted to function values, we will additionally assume that

$$K(a,a) = 0 \tag{2}$$

for a point $a \in D$ called an *anchor*.

We are ready to define the class $\mathscr{F}_{\boldsymbol{\gamma}}$. Let $\mathscr{D}$ be the set of infinite sequences $\mathbf{x} = [x_1, x_2, \dots]$ with $x_i \in D$. For a finite subset $\mathfrak{u}$ of $\mathbb{N}_+ = \{1, 2, \dots\}$, define the reproducing kernel

$$K_{\mathfrak{u}} : \mathscr{D} \times \mathscr{D} \to \mathbb{R} \quad \text{by} \quad K_{\mathfrak{u}}(\mathbf{x}, \mathbf{y}) := \prod_{j \in \mathfrak{u}} K(x_j, y_j)$$
$$\text{for all} \quad \mathbf{x}, \mathbf{y} \in \mathscr{D}, \qquad \text{with} \quad K_{\emptyset} \equiv 1.$$

The RKH space generated by $K_{\mathfrak{u}}$ is denoted by

$$H_{\mathfrak{u}} = H(K_{\mathfrak{u}}),$$

where $H_{\emptyset}$ is the space of constant functions. Although, formally, the functions from $H_{\mathfrak{u}}$ have $\mathscr{D}$ as their domain, they depend only on the variables whose indices are listed in $\mathfrak{u}$. Such variables are referred to as *active variables*.

In a number of important applications, consecutive variables of the functions have *diminishing importance* and/or the spaces of functions have *small effective dimension*, see e.g., [1, 27, 28]. Such function spaces can be modeled by using

[1] The results of [33] hold for general Hilbert spaces F. We restrict the attention to RKH spaces to simplify the presentation.

weights $\gamma = \{\gamma_{\mathfrak{u}}\}_{\mathfrak{u}}$, where $\gamma_{\mathfrak{u}}$ is a non-negative number. The role of $\gamma_{\mathfrak{u}}$ is to quantify the importance of the group of variables with indices in $\mathfrak{u}$; the larger $\gamma_{\mathfrak{u}}$ the more important the group. In particular, $\gamma_{\mathfrak{u}} = 0$ means that the corresponding group of variables does not contribute to the functions. For example, when $\gamma_{\mathfrak{u}} = 0$ if $|\mathfrak{u}| \geq 3$ then f is a sum of functions with each term depending on at most two variables.

Although results of [33, 34] hold for general weights, for simplicity of presentation we will restrict the attention to the *product weights* of the form

$$\gamma_{\mathfrak{u}} = \prod_{j\in\mathfrak{u}} \gamma_j \quad \text{and} \quad \gamma_{\emptyset} = 1,$$

where γ_j are positive numbers. Without loss of generality, we assume that they are ordered,

$$\gamma_j \geq \gamma_{j+1} \quad \text{for } j \geq 1.$$

Consider next $\mathscr{H}_{\boldsymbol{\gamma}}$ as the pre-Hilbert space spanned by the spaces $H_{\mathfrak{u}}$ and equipped with the inner-product

$$\left\langle \sum_{\mathfrak{u}\subset\mathbb{N}_+} f_{\mathfrak{u}}, \sum_{\mathfrak{u}\subset\mathbb{N}_+} g_{\mathfrak{u}} \right\rangle := \sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}}^{-1} \cdot \langle f_{\mathfrak{u}}, g_{\mathfrak{u}} \rangle_{H_{\mathfrak{u}}}$$

for $\sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}}^{-1} \cdot \|f_{\mathfrak{u}}\|_{H_{\mathfrak{u}}}^2 < \infty$ and $\sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}}^{-1} \cdot \|g_{\mathfrak{u}}\|_{H_{\mathfrak{u}}}^2 < \infty$. Finally, the space $\mathscr{F}_{\boldsymbol{\gamma}}$ is the completion of $\mathscr{H}_{\boldsymbol{\gamma}}$ with respect to the inner-product introduced above.

Since $1 \notin H_{\mathfrak{u}}$ for all $\mathfrak{u} \neq \emptyset$, the subspaces $H_{\mathfrak{u}}$ are mutually orthogonal and any function $f \in \mathscr{F}_{\boldsymbol{\gamma}}$ has the unique representation

$$f = \sum_{\mathfrak{u}\subset\mathbb{N}_+} f_{\mathfrak{u}} \quad \text{with} \quad f_{\mathfrak{u}} \in H_{\mathfrak{u}}. \tag{3}$$

Clearly, $\mathscr{F}_{\boldsymbol{\gamma}}$ is also separable. Moreover, it is a RKH space iff

$$\sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}} \cdot K_{\mathfrak{u}}(\mathbf{x}, \mathbf{x}) < \infty \quad \text{for all } \mathbf{x} \in \mathscr{D}. \tag{4}$$

Since $\sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}} \cdot K_{\mathfrak{u}}(\mathbf{x}, \mathbf{x}) = \prod_{j=1}^{\infty}(1 + \gamma_j \cdot K(x_j, x_j))$, the condition (4) holds iff

$$\sup_{x\in D} K(x, x) < \infty \quad \text{and} \quad \sum_{j=1}^{\infty} \gamma_j < \infty,$$

and then

$$\mathscr{K}_{\boldsymbol{\gamma}}(\mathbf{x}, \mathbf{y}) := \sum_{\mathfrak{u}} \gamma_{\mathfrak{u}} \cdot K_{\mathfrak{u}}(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^{\infty}(1 + \gamma_j \cdot K(x_j, y_j))$$

is well defined and it is the reproducing kernel of $\mathscr{F}_{\boldsymbol{\gamma}}$.

If (4) does not hold then function sampling, $L_{\mathbf{x}}(f) := f(\mathbf{x})$, is a discontinuous or ill-defined functional for some $\mathbf{x} \in \mathscr{D}$. However, even then, $L_{\mathbf{x}}$ is continuous when $\mathbf{x}$ has only finitely many components different from the anchor a. Indeed, for given $\mathbf{x} \in \mathscr{D}$ and $\mathfrak{u}$, let $[\mathbf{x};\mathfrak{u}]$ be a short hand notation for the point with active variables listed in $\mathfrak{u}$, i.e.,

$$[\mathbf{x};\mathfrak{u}] := \mathbf{y} = [y_1, y_2, \dots] \quad \text{with} \quad y_j := \begin{cases} x_j & \text{if } j \in \mathfrak{u}, \\ a & \text{if } j \notin \mathfrak{u}. \end{cases} \tag{5}$$

Then

$$f([\mathbf{x};\mathfrak{u}]) = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} f_{\mathfrak{v}}(\mathbf{x}) \quad \text{and} \quad \|L_{[\mathbf{x};\mathfrak{u}]}\|^2 = \sum_{\mathfrak{v}\subseteq\mathfrak{u}} \gamma_{\mathfrak{v}} \cdot K_{\mathfrak{v}}(\mathbf{x},\mathbf{x}) < \infty.$$

Of course, $[\mathbf{x};\emptyset] = \boldsymbol{a} = [a, a, \dots]$ and $f([\mathbf{x};\emptyset]) = f_\emptyset$ for any $\mathbf{x} \in \mathscr{D}$ and any $f \in \mathscr{F}_{\boldsymbol{\gamma}}$.

If (4) does not hold then we refer to such spaces as *quasi-reproducing kernel Hilbert* spaces (*Q-RKH* spaces for short). Important examples of such spaces are provided by those generated by Wiener kernel discussed in the following example.

Example 1. Consider

$$K(x,y) = \min(x,y) \quad \text{with} \quad D = [0,1] \quad \text{or} \quad D = [0,\infty).$$

In this case, F consists of (locally) absolutely continuous functions with $f(0) = 0$ and $f' \in L_2(D)$, and the anchor equals $a = 0$. Clearly, if $\sum_{j=1}^\infty \gamma_j < \infty$, then $\mathscr{F}_{\boldsymbol{\gamma}}$ is a RKH space when $D = [0,1]$, and it is only a Q-RKH space when $D = [0,\infty)$ since $\sup_{x\in[0,\infty)} K(x,x) = \infty$.

2.2 Integration Problem

Let ρ be a given probability density (p. d.) function on D. We are interested in approximating integrals

$$\mathrm{INT}(f) := \lim_{d\to\infty} \int_{D^d} f(x_1,\dots,x_d,a,a,\dots) \cdot \prod_{j=1}^d \rho(x_j)\,\mathrm{d}(x_1,\dots,x_d)$$

for $f \in \mathscr{F}_{\boldsymbol{\gamma}}$. We assume that INT is a well defined and continuous functional on $\mathscr{F}_{\boldsymbol{\gamma}}$. Then

$$\|\mathrm{INT}\|^2 = \sum_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}} \cdot C_0^{|\mathfrak{u}|} = \prod_{j=1}^\infty (1 + \gamma_j \cdot C_0) < \infty, \tag{6}$$

where

$$C_0 := \int_D \rho(x) \int_D K(x,y) \cdot \rho(y) \, \mathrm{d}y \, \mathrm{d}x.$$

Thus $\|\mathrm{INT}\| < \infty$ iff

$$C_0 < \infty \quad \text{and} \quad \sum_{j=1}^{\infty} \gamma_j < \infty \tag{7}$$

This is why we will assume that (7) holds whenever the integration problem is considered. Moreover, we will also assume that

$$C_0 > 0$$

since, otherwise, $\mathrm{INT}(f) = f(\mathrm{a})$ for all functions from $\mathscr{F}_{\boldsymbol{\gamma}}$ which makes the integration problem trivial.

2.3 Function Approximation Problem

As in the previous section, ρ is a given probability density on D. Without loss of generality, we assume that it is positive almost everywhere on D. Then $L_2(D,\rho)$ endowed with the norm

$$\|f\|^2_{L_2(D,\rho)} := \int_D |f(x)|^2 \cdot \rho(x) \, \mathrm{d}(x),$$

is a well defined Hilbert space.

Following [33,34], we assume that $H(K)$ is continuously imbedded in $L_2(D,\rho)$, i.e., $H(K) \subseteq L_2(D,\rho)$ and

$$C_1 := \sup_{f \in H(K)} \frac{\|f\|^2_{L_2(D,\rho)}}{\|f\|^2_{H(K)}} < \infty \quad \text{with the convention} \quad \frac{0}{0} = 0.$$

Next, consider the space $\mathscr{G}$ consisting of functions from $\mathscr{F}_{\boldsymbol{\gamma}}$ with the norm defined by

$$\left\| \sum_{\mathfrak{u} \subset \mathbb{N}_+} f_{\mathfrak{u}} \right\|^2_{\mathscr{G}} := \sum_{\mathfrak{u} \subset \mathbb{N}_+} \|f_{\mathfrak{u}}\|^2_{L_2(\rho_{\mathfrak{u}}, D^{|\mathfrak{u}|})}. \tag{8}$$

Note that the last norm is always finite.

We are interested in approximating the imbedding operator

$$\mathrm{APP} : \mathscr{F}_{\boldsymbol{\gamma}} \to \mathscr{G} \quad \text{given by} \quad \mathrm{APP}(f) = f.$$

The problem is well defined if APP is continuous, and this holds iff

$$\|\mathrm{APP}\|^2 = \sup_{\mathfrak{u}\subset\mathbb{N}_+} \gamma_{\mathfrak{u}} \cdot C_1^{|\mathfrak{u}|} < \infty.$$

It is well known that C_1 is the largest eigenvalue of the integral operator

$$W : H(K) \to H(K) \quad \text{given by} \quad W(f)(x) := \int_D f(t) \cdot K(t,x) \cdot \rho(t)\,\mathrm{d}t. \tag{9}$$

We want to stress that the space $\mathscr{G}$ is very special and, perhaps, not always interesting from a practical point of view. In particular, it can happen that the approximation problem is easier than the integration problem, and that

$$\sup_{f\in\mathscr{F}} \frac{|\mathrm{INT}(f)|}{\|f\|_{\mathscr{G}}} < \infty \quad \text{does } \textit{not} \text{ hold in general.} \tag{10}$$

Indeed, take the reproducing kernel K such that $C_0 > 0$ and $C_1 < \infty$. Note that $\int_D K(x,x) \cdot \rho(x)\ \mathrm{d}(x) < \infty$ implies that $C_1 < \infty$. For $\gamma_j = j^{-\beta}$ with $\beta \in (0,1]$, we then have

$$\|\mathrm{INT}\| = \infty \quad \text{while} \quad \|\mathrm{APP}\| = \max_{k\in\mathbb{N}} C_1^k/(k!)^\beta < \infty$$

We chose such a space $\mathscr{G}$ in [33, 34] as the first step in the study of approximation for ∞-variate functions. The results obtained there will be used in a forthcoming paper [35], see also Sect. 4.3, where function approximation is considered with $\mathscr{G}$ replaced by the Hilbert space $\mathscr{L}_2(\mathscr{D},\rho_\infty)$ whose norm is given by

$$\|f\|^2_{\mathscr{L}_2(\mathscr{D},\rho_\infty)} = \lim_{d\to\infty} \int_{D^d} |f(x_1,\dots,x_d,a,a,\dots)|^2 \cdot \prod_{j=1}^d \rho(x_j)\,\mathrm{d}(x_1,\dots,x_d). \tag{11}$$

Of course, we will need stronger assumptions on $\mathscr{F}_{\boldsymbol{\gamma}}$ for $\|f\|_{\mathscr{L}_2(\mathscr{D},\rho_\infty)}$ to be well defined for all $f \in \mathscr{F}_{\boldsymbol{\gamma}}$. However, then

$$|\mathrm{INT}(f)| \le \|f\|_{\mathscr{L}_2(\mathscr{D},\rho_\infty)} \quad \text{for all} \quad f \in \mathscr{F}_{\boldsymbol{\gamma}},$$

and integration is no harder than approximation.

2.4 Algorithms

Let $\mathscr{S}$ be the solution operator whose values $\mathscr{S}(f)$ we want to approximate; $\mathscr{S} =$ INT for the integration problem, and $\mathscr{S} =$ APP for the approximation problem. Since $\mathscr{F}_{\boldsymbol{\gamma}}$ is a Hilbert space, we may restrict the attention to linear algorithms, see e.g., [26],

$$\mathscr{A}_n(f) = \sum_{i=1}^{n} L_i(f) \cdot g_i \tag{12}$$

Here the L_i's are continuous linear functionals and their values $\{L_i(f)\}_{i=1}^n$ provide *information* about the specific function f. The elements g_i's are numbers for integration and functions from $\mathscr{G}$ for approximation.

If L_i's may be arbitrary continuous linear functionals, then we deal with *unrestricted linear* information. In many applications, including integration, only function samplings $L_i(f) = f(\mathbf{t}_i)$ are allowed. Then

$$\mathscr{A}_n(f) = \sum_{i=1}^{n} f(\mathbf{t}_i) \cdot g_i \quad \text{with} \quad \mathbf{t}_i \in \mathscr{D}$$

and this corresponds to *standard information*. Since in general, $\mathscr{F}_\gamma$ is only a Q-RKH space, the sampling points $\mathbf{t}_i$ used by the algorithms have to be restricted to those that have only finitely many active variables, see (5), i.e.,

$$\mathbf{t}_i = [\mathbf{x}_i, \mathfrak{u}_i]$$

for some $\mathbf{x}_i \in \mathscr{D}$ and $\mathfrak{u}_i$. That is, the algorithms using standard information are of the form

$$\mathscr{A}_n(f) = \sum_{i=1}^{n} f([\mathbf{x}_i, \mathfrak{u}_i]) \cdot g_i. \tag{13}$$

We believe that the cost of evaluating f at $\mathbf{t}_i = [\mathbf{x}_i, \mathfrak{u}_i]$ should depend on the number $|\mathfrak{u}_i|$ of active variables in $\mathbf{t}_i$. That is why we assume that the cost equals $\$(|\mathfrak{u}_i|)$ for a given *cost function*

$$\$: \mathbb{N} \to [1, \infty].$$

At this moment, we only require that $ is monotonically non-decreasing. Examples of $ include

$$\$(d) = (1+d)^\alpha, \quad \$(d) = e^{d^\alpha}, \quad \text{and} \quad \$(d) = e^{e^{d^\alpha}} \quad \text{for} \quad \alpha \ge 0.$$

The (information) cost of $\mathscr{A}_n$ is defined as the total cost of sampling f at the points $\mathbf{t}_i = [\mathbf{x}_i, \mathfrak{u}_i]$, i.e.,

$$\operatorname{cost}(\mathscr{A}_n) := \sum_{i=1}^{n} \$(|\mathfrak{u}_i|).$$

For algorithms that use linear functionals $L_i(f)$, the definition of the cost is extended in a natural way with the cost of evaluating L_i given as follows. Let $L_i(f) = \langle f, h_i \rangle_{\mathscr{F}_\gamma}$, where $h_i \in \mathscr{F}_\gamma$ is the generator of L_i. For any $h \in \mathscr{F}_\gamma$, let

$$\mathrm{Var}(h) := \{\mathfrak{u} \;:\; h_{\mathfrak{u}} \neq 0\} \quad \text{for} \quad h = \sum_{\mathfrak{u} \subset \mathbb{N}_+} h_{\mathfrak{u}}.$$

Then $|\mathrm{Var}(h)|$ is the number of active variables in h and the cost of $L_i(f)$ is defined as $\$(|\mathrm{Var}(h_i)|)$.

We say that an algorithm $\mathscr{A}_n$ of the form (12) with $L_f(f) = \langle f, h_i \rangle_{\mathscr{F}_\gamma}$, is of a *fixed dimension* (FD), if there is a finite set V of $\mathbb{N}_+$ such that

$$\mathrm{Var}(h_i) = V \quad \text{for all} \quad i = 1, 2, \ldots, n.$$

For example, we may have $\mathrm{Var}(h_i) = \{1, \ldots, d\}$, for all i. Otherwise, the algorithm is of a *changing dimension* (CD). As observed in [16], CD algorithms may be significantly superior to FD algorithms.

In the *worst case setting*, the error of $\mathscr{A}_n$ is defined by

$$\mathrm{error}^{\mathrm{wor}}(\mathscr{A}_n) = \mathrm{error}^{\mathrm{wor}}(\mathscr{A}_n; \mathscr{F}_\gamma, \mathscr{T}) := \sup_{\|f\|_{\mathscr{F}_\gamma} \le 1} \|\mathscr{T}(f) - \mathscr{A}_n(f)\|_{\mathscr{G}}.$$

In the *randomized setting*, the choice of the functionals L_i or function sample points $[\mathbf{x}_i; \mathfrak{u}_i]$ may be random. Then the error of a randomized algorithm is defined by

$$\mathrm{error}^{\mathrm{ran}}(\mathscr{A}_n) = \mathrm{error}^{\mathrm{ran}}(\mathscr{A}_n; \mathscr{F}_\gamma, \mathscr{T}) := \sup_{\|f\|_{\mathscr{F}_\gamma} \le 1} \left(\mathbb{E}\, \|\mathscr{T}(f) - \mathscr{A}_n(f)\|_{\mathscr{G}}^2 \right)^{1/2},$$

where $\mathbb{E}$ denotes the expectation with respect to all random parameters in the randomized algorithm $\mathscr{A}_n$.

2.5 Complexity and Tractability

For a given error demand $\varepsilon > 0$, let

$$\mathrm{comp}^{\mathrm{sett}}(\varepsilon) = \mathrm{comp}^{\mathrm{sett}}(\varepsilon; \mathscr{F}_\gamma, \mathscr{T}) := \inf \left\{ \mathrm{cost}(\mathscr{A}_n) \;:\; \mathrm{error}^{\mathrm{sett}}(\mathscr{A}_n) \le \varepsilon \right\}$$

be the minimal cost among algorithms with errors not exceeding ε. Here and elsewhere, sett $\in$ {wor, ran} denotes the setting.

When only standard information is allowed, we consider of course only algorithms that use function values. To distinguish the complexities with standard and unrestricted linear information, we will sometimes write

$$\mathrm{comp}^{\mathrm{sett}}(\varepsilon; \Lambda) \quad \text{or} \quad \mathrm{comp}^{\mathrm{sett}}(\varepsilon; \Lambda, \mathscr{F}_\gamma, \mathscr{T})$$

with $\Lambda = \Lambda^{\text{std}}$ for standard information and $\Lambda = \Lambda^{\text{all}}$ for unrestricted linear information.

We say that the problem is *weakly tractable* if the complexity is not exponential in $1/\varepsilon$, i.e.,

$$\lim_{\varepsilon\to 0} \varepsilon \cdot \ln\left(\text{comp}^{\text{sett}}(\varepsilon)\right) = 0.$$

A stronger notion is *polynomial tractability* which means that there are some non-negative C and p such that

$$\text{comp}^{\text{sett}}(\varepsilon) \le C \cdot \varepsilon^{-p} \quad \text{for all} \quad \varepsilon > 0.$$

The smallest (or more precisely, infimum of) such p is called the *exponent of polynomial tractability*, i.e.,

$$p^{\text{sett}} := \limsup_{\varepsilon\to 0} \frac{\ln(\text{comp}^{\text{sett}}(\varepsilon))}{\ln(1/\varepsilon)}.$$

We sometimes write $p^{\text{sett}} = p^{\text{sett}}(\Lambda)$ or $p^{\text{sett}}(\Lambda, \mathscr{F}_{\boldsymbol{\gamma}}, \mathscr{T})$ with $\Lambda \in \{\Lambda^{\text{all}}, \Lambda^{\text{std}}\}$ to stress what type of information is used.

3 Results for Integration

We present in this section selected results from [23] for CD algorithms. Recall that these algorithms were defined for the first time in [16] and have the following form

$$\mathscr{A}_n(f) = \sum_{i=1}^{n} f([\mathbf{x}_i; \mathfrak{u}_i]) \cdot g_i$$

for some points $\mathbf{x}_i$, sets of active variables $\mathfrak{u}_i$, and the numbers g_i which may depend on n. Moreover, in the randomized setting, all parameters $\mathbf{x}_i, \mathfrak{u}_i$, and g_i may be chosen randomly.

In what follows, the operator I is the integration operator for functions from $H(K)$,

$$I(f) = \int_D f(x) \cdot \rho(x)\, \mathrm{d}x.$$

Theorem 1. *Let* sett $\in$ {wor, ran}. *Suppose that*

- *The product weights satisfy*

$$\gamma_j = \mathscr{O}(j^{-\beta}) \quad \textit{for} \quad \beta > 1, \tag{14}$$

- *There exists a sequence of algorithms $\{A_n\}_n$ for the univariate problem and positive constants α, c such that A_n uses at most n function evaluations and the error of A_n for the univariate integration problem over the space $H(K)$ satisfies*

$$\mathrm{error}^{\mathrm{sett}}(A_n; H(K), I) \le c \cdot n^{-\alpha} \quad \textit{for all} \quad n \in \mathbb{N}. \tag{15}$$

Then there are algorithms $\{\mathscr{A}_\varepsilon\}_\varepsilon$ for the ∞-variate integration problem such that

$$\mathrm{error}^{\mathrm{sett}}(\mathscr{A}_\varepsilon; \mathscr{F}_{\boldsymbol{\gamma}}, \mathrm{INT}) \le \varepsilon \quad \textit{for all} \quad \varepsilon > 0$$

with the following bounds on their cost.

- *If $\$(d) = \mathscr{O}\left(e^{k\cdot d}\right)$ for some $k \ge 0$, then for all $p > \max\left(\frac{1}{\alpha}, \frac{2}{\beta-1}\right)$ there exists a number C depending, in particular, on p such that*

  $$\mathrm{cost}(\mathscr{A}_\varepsilon) \le C \cdot \varepsilon^{-p} \quad \textit{for all} \quad \varepsilon > 0.$$

  This means that the ∞-variate integration problem is polynomially tractable with the exponent at most

  $$\max\left(\frac{1}{\alpha}, \frac{2}{\beta - 1}\right).$$

  Furthermore, in the worst case setting, the exponent is equal to the maximum above if α and β are sharp, and $\$(d) = \Omega(d)$.
- *If $\$(d) = \mathscr{O}\left(e^{e^{k\cdot d}}\right)$ for some $k \ge 0$, then*

  $$\lim_{\varepsilon\to 0} \varepsilon \cdot \ln(\mathrm{cost}(\mathscr{A}_\varepsilon)) = 0.$$

  This means that the ∞-variate integration problem is weakly tractable.

We now comment on this theorem. As shown in [16] for the integration problem in the worst case setting for the Wiener kernel and $D = [0, 1]$, the assumption (14) is necessary for polynomial tractability. If the algorithms A_n are deterministic then so are the algorithms $\mathscr{A}_\varepsilon$. The proof is constructive. Algorithms $\mathscr{A}_\varepsilon$ are based on Smolyak's construction from [25] and results from [30]. Moreover, the algorithms $\mathscr{A}_\varepsilon$ use function values at points that have at most

$$d(\varepsilon) = o\left(\ln(1/\varepsilon)\right) \quad \text{active variables.}$$

This is why the problem is polynomially tractable even when the cost function \$ is exponential, and is weakly tractable even when the cost function \$ is doubly exponential.

Assume now that the complexity of the univariate integration problem over $H(K)$ is $\Theta(\varepsilon^{-p})$. Then we can find algorithms A_n for which (15) holds with

$\alpha = 1/p$ and this value of α is the largest one. Then the exponent of polynomial tractability equals

$$p^{\text{sett}} = \alpha^{-1} = p \quad \text{whenever} \quad \beta \geq 1 + 2/p. \tag{16}$$

In this case, the ∞-variate problem is roughly of the same complexity as the univariate problem.

If $\beta \in (1, 1 + 2/p)$ then the ∞-variate problem is harder than the univariate problem but still we have polynomial tractability. In this case, however, the exponent can be arbitrarily large. The proof that the exponent is sharp also in this case is based on a lower bound from [16] for the ∞-integration problem in the worst case setting.

We illustrate the theorem for the Wiener kernel.

Example 1 (continued). For

$$K(x, y) = \min(x, y), \quad D = [0, 1], \quad \text{and} \quad \rho \equiv 1,$$

the condition (15) holds with $\alpha = 1$ in the worst case setting and with $\alpha = 3/2$ in the randomized setting, and both values are sharp. Hence

$$p^{\text{wor}} = \max\left(1, \frac{2}{\beta - 1}\right) \quad \text{and} \quad \frac{2}{3} \leq p^{\text{ran}} \leq \max\left(\frac{2}{3}, \frac{2}{\beta - 1}\right).$$

Note that the exponent in the randomized setting is smaller then the exponent in the worst case setting if $\beta > 3$. It is open what is the actual value of p^{ran} for $\beta \in (1, 4)$.

4 Results for Approximation

We present in this section selected results from [33] for unrestricted linear information and from [34] for standard information. All of them are for the worst case setting and for the range space $\mathscr{G}$. We next discuss extensions of the results to the randomized setting and to the range space $\mathscr{L}_2(\mathscr{D}, \rho_\infty)$. Recall that for the approximation problem, APP is the imbedding operator from $\mathscr{F}_{\boldsymbol{\gamma}}$ to $\mathscr{G}$. We will also use S to denote the imbedding from $H(K)$ to $L_2(D, \rho)$.

4.1 Unrestricted Linear Information

Consider the operator W defined by (9). It is well known, see e.g., [26], that a necessary condition for polynomial tractability of the approximation problem is a polynomial dependence of the eigenvalues λ_j of W, i.e.,

$$\lambda_j = \mathscr{O}\left(j^{-2\cdot\alpha}\right) \quad \text{for some} \quad \alpha > 0. \tag{17}$$

This is because the errors of optimal algorithms A_n^* for the univariate approximation over $H(K)$ are equal to

$$\text{error}^{\text{wor}}(A_n^*; H(K), S) = \sqrt{\lambda_{n+1}} = \mathscr{O}\left(n^{-\alpha}\right),$$

or equivalently,

$$\text{comp}^{\text{wor}}(\varepsilon; \Lambda^{\text{all}}, H(K), S) = \$(1) \cdot \inf\left\{n \ : \ \lambda_{n+1} \le \varepsilon^2\right\}.$$

One of the results in [33] is the construction of optimal algorithms for the ∞-variate problem which allows to get a necessary and sufficient condition on the polynomial tractability for general weights $\gamma_{\mathfrak{u}}$. Here we state one special result for the product weights.

Theorem 2. *Consider the worst case setting. Suppose that the product weights satisfy*

$$\gamma_j = \mathscr{O}\left(j^{-\beta}\right) \quad \textit{for} \quad \beta > 0 \tag{18}$$

and the eigenvalues satisfy (17)*. Then there are algorithms $\{\mathscr{A}_\varepsilon\}_\varepsilon$ for the ∞-variate approximation problem such that*

$$\text{error}^{\text{wor}}(\mathscr{A}_\varepsilon; \mathscr{F}_{\boldsymbol{\gamma}}, \text{APP}) \le \varepsilon$$

with the following bounds on their cost.

- *If $\$(d) = \mathscr{O}\left(e^{k\cdot d}\right)$ for some $k \ge 0$, then for all $p > \max\left(\frac{1}{\alpha}, \frac{2}{\beta}\right)$ there exists a number C depending, in particular, on p such that*

  $$\text{cost}(\mathscr{A}_\varepsilon) \le C \cdot \varepsilon^{-p} \quad \textit{for all } \varepsilon > 0.$$

  This means that the ∞-variate problem is polynomially tractable with the exponent at most

  $$\max\left(\frac{1}{\alpha}, \frac{2}{\beta}\right).$$

  Furthermore, the exponent is equal to the maximum above if α and β are sharp, and $\$(d) = \Omega(d)$.
- *If $\$(d) = \mathscr{O}\left(e^{e^{k\cdot d}}\right)$ for some $k \ge 0$, then*

  $$\lim_{\varepsilon \to 0} \varepsilon \cdot \ln\left(\text{cost}(\mathscr{A}_\varepsilon)\right) = 0.$$

  This means that the problem is weakly tractable.

As before, the proof is constructive. Moreover, $\mathscr{A}_\varepsilon$ uses inner products with generators having at most $d(\varepsilon)$ active variables, where now

$$d(\varepsilon) = \mathscr{O}\left(\frac{\ln(1/\varepsilon)}{\ln(\ln(/\varepsilon))}\right).$$

Example 1 (continued). For the Wiener kernel, $D = [0,1]$, and $\rho \equiv 1$, we have $\alpha = 1$ and, hence,

$$p^{\mathrm{wor}}(\Lambda^{\mathrm{all}}) = \max\left(1, \frac{2}{\beta}\right).$$

4.2 Standard Information

We have a similar result for algorithms using standard information, see [34, Theorem 7].

Theorem 3. *Consider the worst case setting. Suppose that the product weights satisfy (18) and there exists a sequence of algorithms $\{A_n\}_n$, each using at most n function evaluations, such that their errors for the univariate approximation problem over the space $H(K)$ satisfy*

$$\mathrm{error}^{\mathrm{wor}}(A_n; H(K), S) \le c \cdot n^{-\alpha} \quad \textit{for} \quad \alpha > 0. \tag{19}$$

Then there are algorithms $\{\mathscr{A}_\varepsilon\}_\varepsilon$ for the ∞-variate approximation problem using standard information such that

$$\mathrm{error}^{\mathrm{wor}}(\mathscr{A}_\varepsilon; \mathscr{F}_{\boldsymbol{\gamma}}, \mathrm{APP}) \le \varepsilon$$

with the following bounds on their cost.

- *If $\$(d) = \mathscr{O}\left(e^{k\cdot d}\right)$ for some $k \ge 0$, then for all $p > \max\left(\frac{1}{\alpha}, \frac{2}{\beta}\right)$ there exists a number C depending, in particular, on p such that*

  $$\mathrm{cost}(\mathscr{A}_\varepsilon) \le C \cdot \varepsilon^{-p} \quad \textit{for all} \quad \varepsilon \in (0,1).$$

  This means that the problem is polynomially tractable with the exponent at most

  $$\max\left(\frac{1}{\alpha}, \frac{2}{\beta}\right).$$

  Furthermore, the exponent is equal to the maximum above if α and β are sharp, and $\$(d) = \Omega(d)$.

- *If* $\$(d) = \mathscr{O}\left(e^{e^{k\cdot d}}\right)$ *for some* $k \geq 0$*, then*

$$\lim_{\varepsilon\to 0} \varepsilon \cdot \ln\left(\mathrm{cost}(\mathscr{A}_\varepsilon)\right) = 0.$$

This means that the ∞*-variate problem is weakly tractable.*

Again, the proof is constructive and the sampling points used by $\mathscr{A}_\varepsilon$ have at most $d(\varepsilon)$ active variables with

$$d(\varepsilon) = \mathscr{O}\left(\frac{\ln(1/\varepsilon)}{\ln(\ln(1/\varepsilon))}\right).$$

We stress that the parameters α in (17) and (19) are *not* necessarily the same. The parameter α in (17) describes the power of unrestricted linear information given by the decay of the eigenvalues λ_j. The parameter α in (19) describes the power of standard information given by the best speed of convergence of algorithms using n function evaluations. There are examples of spaces $H(K)$ for which the values of α for unrestricted linear and standard information are different, see [11]. There is still an open problem whether they are the same if we assume that $\alpha > 1/2$ for unrestricted linear information, see [22] for more details.

Example 1 (continued). For the Wiener kernel, $D = [0,1]$, and $\rho \equiv 1$, the conditions (17) and (19) hold with the same $\alpha = 1$. Therefore, the exponent of polynomial tractability for standard information is the same as for unrestricted linear information,

$$p^{\mathrm{wor}}(\Lambda^{\mathrm{std}}) = p^{\mathrm{wor}}(\Lambda^{\mathrm{all}}) = \max\left(1, \frac{2}{\beta}\right).$$

For this particular space, standard and unrestricted linear information are equally powerful. Note that for $\beta \in (0,3)$, the exponent for the approximation problem is smaller than the corresponding exponent for the integration problem. This is due to the special form of the space $\mathscr{G}$, see Sect. 4.3.

4.3 $\mathscr{L}_2$-Approximation

As already mentioned, the space $\mathscr{G}$ was chosen for the approximation problem since it has a relatively simple structure of the eigenpairs of the operator $\mathscr{W} = \mathrm{APP}^* \circ \mathrm{APP}$. In the forthcoming paper [35], we will present results for the $\mathscr{L}_2$-approximation problem with the space $\mathscr{G}$ replaced by the $\mathscr{L}_2 = \mathscr{L}_2(\mathscr{D}, \rho_\infty)$ space whose norm is given by (11). Here are some results for product weights.

It is easy to see that $\mathscr{L}_2 = \mathscr{G}$ if $C_0 = 0$. This is why we assume that

$$C_0 > 0$$

also for the $\mathscr{L}_2$-approximation problem.

The first result of [35] is for $\$(d) = \left(e^{k\cdot d}\right)$. Then the $\mathscr{L}_2$-approximation problem is polynomially tractable iff the exponent β satisfies

$$\beta > 1.$$

Recall that for the $\mathscr{G}$-approximation problem, we only need $\beta > 0$.

Next, if $\beta > 1$ then the exponent of polynomial tractability of the $\mathscr{L}_2$-approximation problem is bounded by

$$p^{\mathrm{wor}}(\Lambda) \le \max\left(\frac{1}{\alpha}, \frac{2}{\beta - 1}\right),$$

where α is from Theorem 2 for $\Lambda = \Lambda^{\mathrm{all}}$ and from Theorem 3 for $\Lambda = \Lambda^{\mathrm{std}}$. Moreover, if α and β are sharp and $\Omega(d) = \$(d) = \mathscr{O}\left(e^{k\cdot d}\right)$ then

$$p^{\mathrm{wor}}(\Lambda^{\mathrm{std}}) = \max\left(\frac{1}{\alpha}, \frac{2}{\beta - 1}\right).$$

Furthermore, if $\$(d) = \mathscr{O}\left(e^{e^{k\cdot d}}\right)$ then the $\mathscr{L}_2$-approximation problem is weakly tractable. In another words, we have similar results for $\mathscr{L}_2$-approximation as for $\mathscr{G}$-approximation with β replaced by $\beta - 1$.

4.4 Randomized Setting

It has been known for quite some time, see [20, 29], that randomization does not help for multivariate approximation defined over Hilbert spaces when unrestricted linear information is allowed. More precisely, for a Hilbert space F_d of d-variate functions and the the space G_d with norm

$$\|f\|_{G_d}^2 = \int_{D_d} |f(\mathbf{x})|^2 \cdot \rho_d(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

consider the problem of approximating the corresponding imbedding operator $S_d : F_d \to G_d$.

Let sett $\in$ {wor, ran}. Denote by $\kappa^{\mathrm{sett}}(\Lambda^{\mathrm{all}}, S_d)$ the order of convergence of optimal algorithms in the worst case and randomized settings, respectively. That is, $\kappa^{\mathrm{sett}}(\Lambda^{\mathrm{all}}, S_d)$ is the supremum of α for which the worst case (or randomized) error of an optimal algorithm using n linear functionals is of order $n^{-\alpha}$. Then the results of [20, 29] imply that

$$\kappa^{\mathrm{ran}}(\Lambda^{\mathrm{all}}, S_d) = \kappa^{\mathrm{wor}}(\Lambda^{\mathrm{all}}, S_d).$$

More recently, it has been constructively proved in [32] that the standard information is as powerful as Λ^{all} in the randomized setting. That is, if $\kappa^{\mathrm{sett}}(\Lambda^{\mathrm{std}}, S_d)$ denotes the order of convergence of optimal algorithms using standard information then

$$\kappa^{\mathrm{ran}}(\Lambda^{\mathrm{std}}, S_d) = \kappa^{\mathrm{ran}}(\Lambda^{\mathrm{all}}, S_d) = \kappa^{\mathrm{wor}}(\Lambda^{\mathrm{all}}, S_d).$$

As already mentioned, the power of standard information in the worst case setting is not yet completely known. In terms of the orders of convergence, it was recently shown, see [11], that there exist reproducing kernel Hilbert spaces F_d for which

$$\kappa^{\mathrm{wor}}(\Lambda^{\mathrm{std}}, S_d) = 0 \quad \text{and} \quad \kappa^{\mathrm{wor}}(\Lambda^{\mathrm{all}}, S_d) = \frac{1}{2}.$$

However, it is still open whether

$$\kappa^{\mathrm{wor}}(\Lambda^{\mathrm{std}}, S_d) = \kappa^{\mathrm{wor}}(\Lambda^{\mathrm{all}}, S_d) \quad \text{if} \quad \kappa^{\mathrm{wor}}(\Lambda^{\mathrm{all}}, S_d) > \frac{1}{2},$$

see [22] for more details.

Since in the study of multivariate problems the cost is measured by the number of linear functional or function values used by an algorithm, these and further results translate into complexity and tractability results. Namely, the complexity and tractability of $\{S_d\}$ in the worst case setting with Λ^{all} are equivalent to complexity and tractability in the randomized setting with Λ^{all} and/or Λ^{std}.

It turns out that similar results hold for ∞-variate approximation problem with the cost depending on the number of active variables. More precisely, we have the following theorem.

Theorem 4. *Assume that*

- *The cost function* $\$(d) = \mathcal{O}\left(e^{k\cdot d}\right)$ *for some* $k \geq 0$*,*
- *The eigenvalues* $\lambda_j = \mathcal{O}\left(j^{-2\cdot\alpha}\right)$ *for some* $\alpha > 0$*,*
- *The product weights are* $\gamma_j = \mathcal{O}(j^{-\beta})$ *with the exponent* $\beta > 0$ *for the* $\mathcal{G}$*-approximation problem, and* $\beta > 1$ *for the* $\mathcal{L}_2$*-approximation problem.*

Then

$$p^{\mathrm{ran}}(\Lambda^{\mathrm{std}}) = p^{\mathrm{ran}}(\Lambda^{\mathrm{all}}) = p^{\mathrm{wor}}(\Lambda^{\mathrm{all}}).$$

We now outline the proof of this theorem. We will do it only for the $\mathcal{G}$-approximation problem since similar arguments and arguments similar to those in [34] can be used for the $\mathcal{L}_2$-approximation. For this purpose, we need to recall some facts about the optimal algorithms for $\mathcal{G}$-approximation in the worst case setting with Λ^{all}. The optimal algorithm $\mathcal{A}_\varepsilon$ whose error is at most ε has the form

$$\mathcal{A}_\varepsilon(f) = \sum_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} A_{\mathfrak{u},n(\mathfrak{u},\varepsilon)}(f_\mathfrak{u}) \quad \text{for} \quad f = \sum_{\mathfrak{u}\in\mathbf{U}_\gamma} f_\mathfrak{u},$$

where $A_{\mathfrak{u},n(\mathfrak{u},\varepsilon)}$ are special projections into $H_{\mathfrak{u}}$ and they use $n(\mathfrak{u},\varepsilon)$ linear functional evaluations. The set $\mathbf{U}(\varepsilon)$ is a special finite subset of $\mathbf{U}_{\boldsymbol{\gamma}}$. In particular, $\mathscr{A}_\varepsilon(f_{\mathfrak{v}}) = 0$ for $f_{\mathfrak{v}}$ with $\mathbf{v} \notin \mathbf{U}(\varepsilon)$. Furthermore, for all $\mathfrak{u} \in \mathbf{U}(\varepsilon)$ we have $|\mathfrak{u}| \le d(\varepsilon)$, where $d(\varepsilon)$ is the maximal number of active variables. As already mentioned, we have

$$d(\varepsilon) = \max_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} |\mathfrak{u}| = \mathscr{O}\left(\frac{\ln(1/\varepsilon)}{\ln(\ln(1/\varepsilon))}\right).$$

The cost of $\mathscr{A}_\varepsilon$ is given by

$$\mathrm{cost}(\mathscr{A}_\varepsilon) = \sum_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} \$(|\mathfrak{u}|)\cdot n(\mathfrak{u},\varepsilon) \le \$(d(\varepsilon))\cdot \sum_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} n(\mathfrak{u},\varepsilon).$$

Each algorithm $A_{\mathfrak{u},\varepsilon}$ can be replaced by the corresponding randomized algorithm that uses standard information due to the already cited result from [32]. However, these randomized algorithms need to evaluate functions $f_{\mathfrak{u}}$ for $\mathfrak{u} \in \mathbf{U}(\varepsilon)$ instead of the whole function f. As shown in [15], a value of $f_{\mathfrak{u}}$ can be obtained by computing at most $2^{|\mathfrak{u}|}$ values of f at points with at most $|\mathfrak{u}|$ active variables. Note that

$$2^{|\mathfrak{u}|} \le 2^{d(\varepsilon)} \quad \text{and} \quad \frac{\ln\left(2^{d(\varepsilon)}\right)}{\ln(1/\varepsilon)} \le (\ln(1/\varepsilon))^{c/\ln(\ln(1/\varepsilon))-1}$$

for a positive constant c. This implies that

$$p^{\mathrm{ran}}(\Lambda^{\mathrm{std}};\mathrm{APP}) \le p^{\mathrm{wor}}(\Lambda^{\mathrm{all}};\mathrm{APP}).$$

To show the opposite inequality, i.e.,

$$p^{\mathrm{wor}}(\Lambda^{\mathrm{all}};\mathrm{APP}) \le p^{\mathrm{ran}}(\Lambda^{\mathrm{all}};\mathrm{APP}),$$

note that

$$\begin{aligned}\limsup_{\varepsilon\to 0} \frac{\ln(\mathrm{cost}(\mathscr{A}_\varepsilon))}{\ln(1/\varepsilon)} &\le \limsup_{\varepsilon\to 0} \frac{\ln\left(\sum_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} n(\mathfrak{u},\varepsilon)\right) + \ln\left(\$(d(\varepsilon))\cdot 2^{d(\varepsilon)}\right)}{\ln(1/\varepsilon)} \\ &= \limsup_{\varepsilon\to 0} \frac{\ln\left(\sum_{\mathfrak{u}\in\mathbf{U}(\varepsilon)} n(\mathfrak{u},\varepsilon)\right)}{\ln(1/\varepsilon)}\end{aligned}$$

since

$$\limsup_{\varepsilon\to 0} \frac{\ln\left(\$(d(\varepsilon))\cdot 2^{d(\varepsilon)}\right)}{\ln(1/\varepsilon)} \le \lim_{\varepsilon\to 0} (\ln(1/\varepsilon))^{c'/\ln(\ln(1/\varepsilon))-1} = 0.$$

This means that the cost function \$ does not contribute to the tractability exponents $p^{\mathrm{sett}}(\Lambda^{\mathrm{all}};\mathrm{APP})$ and we can replace it by $\$(d) \equiv 1$. For such a constant cost function, the worst case ε-complexity is the same as the complexity with respect the space F_d given by the span of $H_{\mathfrak{u}}$ for $\mathfrak{u} \in \mathbf{U}(\varepsilon/2)$ as follows from the proof in [33]. Moreover, the complexity in the randomized setting is bounded from below if $\mathscr{F}_{\boldsymbol{\gamma}}$ is replaced by F_d. Hence the results from [20, 29] complete the proof for the $\mathscr{G}$-approximation.

Acknowledgements I would like to thank Henryk Woźniakowski for valuable comments and suggestions to this paper.

References

1. Caflisch, R. E., Morokoff, M., Owen, A. B.: Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. J. Computational Finance **1**, 27–46 (1997)
2. Creutzig, J., Dereich, S., Müller-Gronbach, T., Ritter, K.: Infinite-dimensional quadrature and approximation of distributions. Found. Comput. Math. **9**, 391–429 (2009)
3. Das, A.: Field Theory: A Path Integral Approach. Lecture Notes in Physics, Vol. 52, World Scientific, Singapore, 1993
4. DeWitt-Morette, C. (editor): Special Issue on Functional Integration. J. Math. Physics **36**, (1995)
5. Duffie, D.: Dynamic Asset Pricing Theory. Princeton University, Princeton, NJ, 1992
6. Egorov, R. P., Sobolevsky, P. I., Yanovich, L. A.: Functional Integrals: Approximate Evaluation and Applications. Kluver Academic, Dordrecht, 1993
7. Feynman, R. P., Hibbs, A. R.: Quantum Mechanics and Path-Integrals. McGraw-Hill, New York, 1965
8. Gnewuch, M.: Infinite-dimensional integration on weighted Hilbert spaces. Math. Comput. **81**, 2175–2205 (2012)
9. Hickernell, F. J., Müller-Gronbach, T., Niu, B., Ritter, K.: Multi-level Monte Carlo algorithms for infinite-dimensional integration on $\mathbb{R}^{\mathbb{N}}$. J. Complexity **26**, 229–254 (2010)
10. Hickernell, F. J., Wang, X.: The error bounds and tractability of quasi-Monte Carlo algorithms in infinite dimension. Math. Comp. **71**, 1641–1661 (2002)
11. Hinrichs, A., Novak, E., Vybiral, J.: Linear information versus function evaluations for L_2-approximation, J. Complexity **153**, 97–107 (2008)
12. Hull, J.: Option, Futures, and Other Derivative Securities. 2nd ed., Prentice Hall, Engelwood Cliffs. NJ, 1993
13. Khandekar, D. C., Lawande, S. V., Bhagwat, K. V.: Path-Integral Methods and their Applications. World Scientific, Singapore, 1993
14. Kleinert, H.: Path Integrals in Quantum Mechanics, Statistics and Polymer Physics. World Scientific, Singapore, 1990
15. Kuo, F. Y., Sloan, I. H., Wasilkowski, G. W., Woźniakowski, H.: On decompositions of multivariate functions. Math. Comp. **79** 953–966 (2010), DOI: 0.1090/S0025-5718-09-02319-9
16. Kuo, F. Y., Sloan, I. H., Wasilkowski, G. W., Woźniakowski, H.: Liberating the dimension. J. Complexity **26**, 422–454 (2010)
17. Merton, R.: Continuous–Time Finance, Basil Blackwell, Oxford, 1990
18. Niu, B., Hickernell, F. J.: Monte Carlo simulation of stochastic integrals when the cost function evaluation is dimension dependent. In: Ecuyer, P. L., Owen, A. B. (eds) Monte Carlo and Quasi-Monte Carlo Methods 2008, pp. 545–569, Springer (2008)
19. Niu, B., Hickernell, F. J., Müller-Gronbach, T., Ritter, K.: Deterministic multi-level algorithms for infinite-dimensional integration on $\mathbb{R}^{\mathbb{N}}$. Submitted (2010)
20. Novak, E.: Optimal linear randomized methods for linear operators in Hilbert spaces, J. Complexity **8**, 22–36, (1992)

21. Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems, European Mathematical Society, Zürich (2008)
22. Novak, E., Woźniakowski, H.: On the power of function values for the approximation problem in various settings. Submitted (2010)
23. Plaskota, L., Wasilkowski, G. W.: Tractability of infinite-dimensional integration in the worst case and randomized settings. J. Complexity **27**, 505–518 (2011)
24. Plaskota, L., Wasilkowski, G. W., Woźniakowski, H.: A new algorithm and worst case complexity for Feynman-Kac path integration. J. Computational Physics **164**, 335–353 (2000)
25. Smolyak, S. A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Dokl. Acad. Nauk SSSR **4** 240–243 (1963)
26. Traub, J. F., Wasilkowski, G. W., Woźniakowski, H.: Information-Based Complexity, Academic Press, New York (1988)
27. Wang, X., Fang, K. -T.: Effective dimensions and quasi-Monte Carlo integration. J. Complexity **19**, 101–124 (2003)
28. Wang, X., Sloan, I. H.: Why are high-dimensional finance problems often of low effective dimension? SIAM J. Sci. Comput. **27**, 159–183 (2005)
29. Wasilkowski, G. W.: Randomization for continuous problems, J. Complexity **5**, 195–218 (1989)
30. Wasilkowski, G. W., Woźniakowski, H.: Explicit cost bounds for multivariate tensor product problems. J. Complexity **11**, 1–56 (1995)
31. Wasilkowski, G. W., Woźniakowski, H.: On tractability of path integration, J. Math. Physics **37**, 2071–2088 (1996)
32. Wasilkowski, G. W., Woźniakowski, H.: The power of standard information for multivariate approximation in the randomized setting, *Mathematics of Computation* **76**, 965–988 (2007)
33. Wasilkowski, G. W., Woźniakowski, H.: Liberating the dimension for function approximation. J. Complexity **27**, 86–110 (2011)
34. Wasilkowski, G. W., Woźniakowski, H.: Liberating the dimension for function approximation: standard information. J. Complexity **27**, 417–440 (2011)
35. G. W. Wasilkowski, Liberating the dimension for L_2 approximation, J. Complexity **28**, 304–319 (2012)
36. Wiegel, F. W.: Path Integral Methods in Physics and Polymer Physics, World Scientific, Singapore (1986)

A Component-by-Component Construction for the Trigonometric Degree

Nico Achtsis and Dirk Nuyens

— In memory of James Lyness (1932–2010) —

Abstract We propose an alternative to the algorithm from Cools, Kuo, Nuyens (Computing 87(1–2):63–89, 2010), for constructing lattice rules with good trigonometric degree. The original algorithm has construction cost $O(|\mathscr{A}_d(m)| + dN \log N)$ for an N-point lattice rule in d dimensions having trigonometric degree m, where the set $\mathscr{A}_d(m)$ has exponential size in both d and m (in the "unweighted degree" case, which is what we consider here). We reduce the cost to $O(dN(\log N)^2)$ with an implicit constant governing the needed precision (which is dependent on N and d).

1 Introduction

Consider d-dimensional integrand functions f having absolutely convergent Fourier series representation

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h})\, \mathrm{e}^{2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}},$$

then the error of integration by means of a *rank*-1 *lattice rule* [19, 24] is given by

N. Achtsis · D. Nuyens
Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
e-mail: nico.achtsis@cs.kuleuven.be; dirk.nuyens@cs.kuleuven.be

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_10, © Springer-Verlag Berlin Heidelberg 2012

$$Q(f; z, N) - I(f) = \frac{1}{N} \sum_{k=0}^{N-1} f\left(\frac{kz \bmod N}{N}\right) - \int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$
$$= \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \setminus \{\boldsymbol{0}\} \\ \boldsymbol{h} \cdot z \equiv 0 \pmod N}} \hat{f}(\boldsymbol{h}), \tag{1}$$

where $I(f)$ is the integral of f and $Q(f; z, N)$ its approximation by an N-point (rank-1) lattice rule with integer *generating vector* z. The set $\Lambda^\perp := \{\boldsymbol{h} \in \mathbb{Z}^d : \boldsymbol{h} \cdot z \equiv 0 \pmod N)\}$, appearing in (1), is called the *dual lattice* (for the lattice Λ with generator $z/N + \mathbb{Z}^d$). We want to construct lattice rules which integrate exactly all Fourier coefficients which are at most a distance m from the origin measured by the 1-norm. The largest such m, for a fixed rule Q, then denotes the *trigonometric degree* of the lattice rule. Figure 1 shows the Fourier space for the trigonometric degree, as well as for the *product trigonometric degree*, which measures the distance in the ∞-norm, to be used in the next section. The trigonometric degree and similar quantities, originating in the Russian literature, have been studied in many Western publications, some of them by Lyness [6, 16–18]; other references are [1–3, 5, 7–11, 23].

One is able to easily write down the reproducing kernel of such a (finite) dimensional reproducing kernel Hilbert space (RKHS) in terms of an orthonormal basis. For a space of functions of trigonometric degree at most m we get

$$K_m(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 \le m}} \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x})\, \overline{\exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{y})} = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 \le m}} \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot (\boldsymbol{x} - \boldsymbol{y})). \tag{2}$$

The squared *worst-case error* using a rank-1 lattice rule in this RKHS is then given by

$$e^2(z, N; K_m) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 \le m}} \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot (kz)/N), \tag{3}$$

see, e.g., [15] for expressing worst-case errors in a RKHS. The *worst-case error* for a quadrature/cubature rule Q in a Banach space $\mathscr{H}$ is defined as

$$e(Q; \mathscr{H}) := \sup_{\substack{f \in \mathscr{H} \\ \|f\|_{\mathscr{H}} \le 1}} |I(f) - Q(f)|.$$

If the rank-1 rule specified by z and N has trigonometric degree m, then its worst-case error in the RKHS with kernel K_m will be zero. The latter form for the squared worst-case error (3) is, for $d \gg 1$, far from convenient for construction purposes as the sum over the Fourier indices $\boldsymbol{h}$ cannot be written in a "product form". A kernel which can be written in a product form (or a small sum of product forms) is a necessary condition for the current fast component-by-component algorithms, see, e.g., [4, 21] for some example kernels.

For comparison, the classical infinite dimensional function space which takes all Fourier coefficients into account, the so-called Korobov space, has reproducing kernel, for $\alpha > 1$,

$$K_\alpha(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \frac{\exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot (\boldsymbol{x} - \boldsymbol{y}))}{\prod_{j=1}^d \max(1, |h_j|^\alpha)} = \prod_{j=1}^d \left(1 + \sum_{0 \neq h \in \mathbb{Z}} \frac{\exp(2\pi \mathrm{i}\, h(x_j - y_j))}{|h|^\alpha} \right),$$

where the infinite sum reduces to a Bernoulli polynomial B_α in case α is even. The squared worst-case error using a rank-1 lattice rule is then

$$e^2(\boldsymbol{z}, N; K_\alpha) = -1 + \frac{1}{N} \sum_{k=0}^{N-1} \prod_{j=1}^d \left(1 + c_\alpha\, B_\alpha \left(\frac{k z \bmod N}{N} \right) \right), \tag{4}$$

for some easily determined constant c_α.

The kernels we consider here are all in terms of Fourier series, therefore they are what is called *shift-invariant* or periodic, i.e., $K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{0})$. In general the squared worst-case error for a shift-invariant space with kernel K using a lattice rule is given by

$$e^2(\boldsymbol{z}, N; K) = -\int_{[0,1)^d} K(\boldsymbol{x}, \boldsymbol{0})\, \mathrm{d}\boldsymbol{x} + \frac{1}{N} \sum_{k=0}^{N-1} K(\boldsymbol{z} k/N, \boldsymbol{0}). \tag{5}$$

Using the Fourier expansion of $K(\boldsymbol{x}, \boldsymbol{0}) = K_{\boldsymbol{0}}(\boldsymbol{x})$, i.e., the kernel with one leg fixed, we arrive at

$$e^2(\boldsymbol{z}, N; K) = -\widehat{K}_{\boldsymbol{0}}(\boldsymbol{0}) + \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \widehat{K}_{\boldsymbol{0}}(\boldsymbol{h}) \frac{1}{N} \sum_{k=0}^{N-1} \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{z} k/N) = \sum_{\substack{\boldsymbol{0} \neq \boldsymbol{h} \in \mathbb{Z}^d \\ \boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod{N}}} \widehat{K}_{\boldsymbol{0}}(\boldsymbol{h}),$$

where the latter sum is over the *dual lattice*. If one compares with (1) then it is clear that (5) is the integration error of the function $K_{\boldsymbol{0}}(\boldsymbol{x})$ using the lattice rule $Q(\cdot; \boldsymbol{z}, N)$. In other words: the squared worst-case error of a lattice rule (in a shift-invariant space) is given as the sum of the Fourier coefficients of the kernel (with one leg fixed to $\boldsymbol{0}$) over the dual lattice. Therefore, *the Fourier coefficients attach a weight to the dual lattice points in the squared worst-case error*; this will be the point of view we will use in the following.

2 Embedding by a Tensor Product Function Space

In [5] a new algorithm was introduced to construct rank-1 lattice rules using a component-by-component procedure that obtains a prescribed *weighted* degree of exactness and, at the same time, achieve the near optimal worst-case error in a Korobov space. (The algorithm in [5] is presented for N prime, but can be modified for composite N as well. Also, the algorithm there is presented for different kinds of degrees of exactness, here we are only concerned with the trigonometric degree.) This algorithm explicitly constructs a set of Fourier indices $\mathscr{A}_d(m)$ associated with the degree of exactness, i.e., all integer points at a distance smaller than or equal to m to the origin. The construction cost of that algorithm is $O(|\mathscr{A}_d(m)| + dN \log N)$. To make this algorithm feasible the degree of exactness is weighted by weights β_j w.r.t. the different coordinate axes $j = 1, \dots, d$. If all these weights are put equal to 1 then one obtains the classical trigonometric degree and the size of the set $\mathscr{A}_d(m)$ increases exponentially in d and m, making the construction intractable. More precisely, it can be shown, see, e.g., [11], that

$$|\mathscr{A}_d(m)| = |\mathscr{A}_m(d)| = \sum_{s\geq 0} 2^s \binom{d}{s}\binom{m}{s} \leq \begin{cases} (1+2m)^d = O((2m)^d), & \text{if } d \leq m, \\ (1+2d)^m = O((2d)^m), & \text{if } m \leq d, \end{cases} \tag{6}$$

where we used the Binomial theorem and the easy estimate $\binom{n}{k} \leq n^k/k! \leq n^k$. (Note that the sum in (6) always has a finite summation range as both d and m are finite positive integers and $\binom{n}{k} = 0$ for $k \notin \{0, \dots, n\}$, $k, n \in \mathbb{Z}$, $n \geq 0$.)

In [5] the theoretical basis starts off by modifying the classical Korobov space to incorporate the kernel of the finite dimensional space, which is (2) for the trigonometric degree. The unfortunate form of this kernel plays no part there as one constructs the set $\mathscr{A}_d(m)$ explicitly and thus no calculations have to be done using this kernel. Here we propose to walk the other way: we will not build the (exponentially growing) set $\mathscr{A}_d(m)$, but will try to calculate the worst-case error for a modified trigonometric space.

Incorporating an idea from [13] we build a function space with exponentially decaying Fourier coefficients, and, extending what is studied in [13], make it finite dimensional. Our first attempt at an efficient kernel is

$$K_{m,p}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 \leq m}} p^{\|\boldsymbol{h}\|_1} \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot (\boldsymbol{x} - \boldsymbol{y})), \tag{7}$$

for $0 < p < 1$. Note that the part inside the sum is now of "product form", however the multiple sums are a dependent chain. If one strives for exactness, i.e., integrate all these Fourier coefficients exactly, then there is no difference in using kernel (2) or (7). A rule which is exact for all trigonometric polynomials up to degree m will have a squared worst-case error equal to zero for both of these choices. Moreover, as the sum still involves the 1-norm, we still fail to have an efficient computable form.

Now we enlarge the index set of Fourier coefficients to take a tensor product form. Again for $0 < p < 1$, now consider the kernel

$$\begin{aligned} K'_{m,p}(\boldsymbol{x},\boldsymbol{y}) &= \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_\infty\le m}} p^{\|\boldsymbol{h}\|_1} \exp(2\pi\mathrm{i}\, \boldsymbol{h}\cdot(\boldsymbol{x}-\boldsymbol{y})) \\ &= \prod_{j=1}^{d} \sum_{h=-m}^{m} p^{|h|} \exp(2\pi\mathrm{i}\, h(x_j-y_j)) \\ &= \prod_{j=1}^{d} \left(1 + 2\sum_{h=1}^{m} p^h \cos(2\pi\, h(x_j-y_j))\right). \end{aligned} \tag{8}$$

The last form is suitable to be directly used in the fast component-by-component algorithm [20–22].

The problem with (8) however is that we are now in fact looking at a *product* trigonometric degree (i.e., a tensor product form degree) instead of the plain trigonometric degree: that is, if the squared worst-case error for this kernel is zero, then the rule has *product* trigonometric degree at least m (and by extension also trigonometric degree at least m), if it is non-zero however, then we could still have trigonometric degree at least m. This simple embedding can be seen in Fig. 1. We want to obtain bounds on the value for the squared worst-case error such that we can determine, when it is non-zero, if the Fourier coefficients for $\|\boldsymbol{h}\|_\infty \le m$, which we don't integrate exactly (i.e., the dual lattice points), actually have $\|\boldsymbol{h}\|_1 > m$, i.e., all in the shaded area in Fig. 1 (but not on the border of the inner diamond). If so, then the rule has trigonometric degree at least m (right image; with actual trigonometric degree m), if not, then the rule has smaller degree (left image).

For ease of presentation one often uses the concept of the *enhanced trigonometric degree* [6] which is defined as the trigonometric degree plus one. In other words, the enhanced trigonometric degree is the distance of the closest non-zero point to the origin of the dual lattice measured in the 1-norm.

We can rewrite kernel (8), getting rid of the sum, using

$$1 + 2\sum_{h=1}^{m} p^h \cos(2\pi h t) = \frac{1 - p^2 - 2p^{m+1}\cos(2\pi(m+1)t) + 2p^{m+2}\cos(2\pi m t)}{1 + p^2 - 2p\cos(2\pi t)}$$

which can be obtained by tedious calculations starting from the exponential form or using easy manipulations starting from [14, 1.353/3]. However, care must be taken to evaluate this function (in whatever form), especially as p will be chosen small. In Fig. 2 one can see what the one-dimensional kernel looks like. (A similar remark is also in place for the kernel used in [13] which has $m = \infty$.)

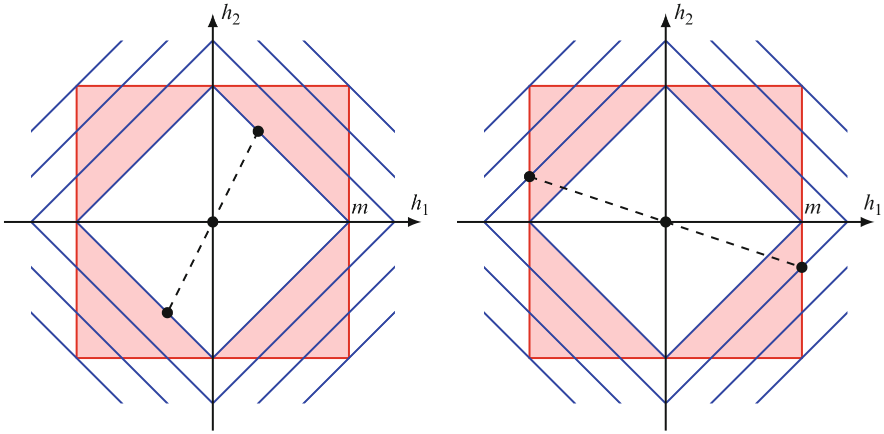


Fig. 1 The trigonometric degree iso-lines (1-norm: diamond shaped iso-lines) versus the product trigonometric degree iso-lines (∞-norm: iso-lines parallel to the axes). Note that, in contrast to what this 2-dimensional figures suggest, the difference in volume for the enclosing product degree shape increases exponentially with the dimension. *Left view*: dual lattice points $\boldsymbol{h} \neq \mathbf{0}$ on the 1-norm iso-line of distance m, i.e., $\|\boldsymbol{h}\|_1 = m$; the picture shows an *enhanced trigonometric degree* of m, i.e., a trigonometric degree of $m - 1$. *Right view*: no dual lattice points $\boldsymbol{h} \neq \mathbf{0}$ with $\|\boldsymbol{h}\|_1 \leq m$, i.e., having trigonometric degree at least m; in the picture the enhanced trigonometric degree is $m + 1$ and thus the trigonometric degree is m.

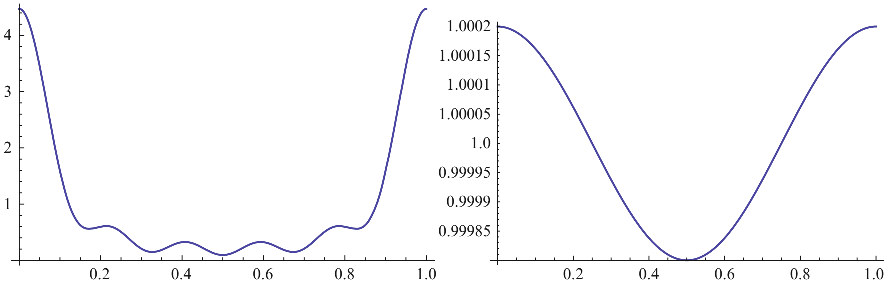


Fig. 2 The one-dimensional kernel $K'_{m,p}(x, 0)$, see (8), of the finite dimensional product space which weights Fourier coefficients by the 1-norm. *Left*: the kernel for $p = 2/3$ and $m = 5$. *Right*: the kernel for $p = 10^{-4}$ and $m = 5$.

3 Distinguishing Dual Lattice Points

Kernel (8) can be analyzed for the trigonometric degree by looking at the different cases where the squared worst-case error for kernel $K'_{m,p}$ is non-zero. We analyze the cases under the premise that the rule has trigonometric degree m.

First assume the rule really has trigonometric degree at least m, i.e., $\|\boldsymbol{h}\|_1 > m$ for all dual lattice points $\boldsymbol{h} \neq \mathbf{0}$, and also has dual lattice points for which $\|\boldsymbol{h}\|_\infty \leq m$, i.e., dual lattice points in the shaded area of Fig. 1. The first 1-norm iso-line on which these points could fall is the one where the 1-norm equals $m+1$ (right image).

All points on this iso-line account for a weight of p^{m+1} in the squared worst-case error. Naturally, if there are $1/p$ dual lattice points on this line, then the squared worst-case error will be at least $1/p \times p^{m+1} = p^m$, which is the weight of the iso-line $\|\boldsymbol{h}\|_1 = m$.

Conversely, assume the rule actually has degree less than m. If there are no dual lattice points on the mth iso-line then the worst-case error is at least p^{m-1}. On the other hand, if there would be any dual lattice points on the mth iso-line then the squared worst-case error would as well have a value of order p^m. Further, note that if there is one dual lattice point at distance m, then there is a second one as well as trivially $\|\boldsymbol{h}\|_1 = \|-\boldsymbol{h}\|_1$, thus the squared worst-case error would at least be $2\,p^m$. So, in the case above, where we have trigonometric degree at least m, we would need at least $2/p$ dual lattice points on the iso-line of weight $m+1$ to have a squared worst-case error of at least $2\,p^m$.

Unsurprisingly, this shows that the contribution of the dual lattice points with $\|\boldsymbol{h}\|_1 > m$ and $\|\boldsymbol{h}\|_\infty \leq m$ could raise above the level of the dual lattice points with $\mathbf{0} \neq \|\boldsymbol{h}\|_1 \leq m$. This problem can be avoided by choosing p small enough since there is a maximum of integer points which can fall inside the shaded region in Fig. 1. A naive but straightforward way is by weighting all points in $\{\boldsymbol{h} \in \mathbb{Z}^d : \|\boldsymbol{h}\|_1 > m \text{ and } \|\boldsymbol{h}\|_\infty \leq m\}$ by the same factor p^{m+1} which then have a combined weight smaller than two points on the edge of the cross-polytope.

Lemma 1. *Given integer $m, d > 1$, if one chooses p such that*

$$\frac{1}{p} > 2^{d-1}\left((m+1)^d - \frac{(d+1)\cdots(d+m)}{m!}\right)$$

then

$$\sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 > m \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{m+1} < 2\,p^m.$$

Proof. We will count the integer points by subtracting the points with $\|\boldsymbol{h}\|_1 \leq m$ from the points with $\|\boldsymbol{h}\|_\infty \leq m$. As we are counting all integer points (instead of only the dual lattice points) we can simplify the count to $\boldsymbol{h}$ with non-negative coordinates and then multiply by a factor of 2^d. Doing so we count all points on the interface between adjacent hypercubes twice, so this is just an approximation.

The number of integer points in $[0, m]^d$ is trivially $(m+1)^d$. To find the number of integer points in the simplex with vertices $(0, 0, \ldots, 0)$, $(m, 0, \ldots, 0)$, $(0, m, \ldots, 0)$, $\ldots$, $(0, 0, \ldots, m)$ we can use the theory of Ehrhart polynomials, see, e.g., [12], from which we find the generating function

$$\sum_{m\geq 0} a_m\, x^m = \frac{1}{(1-x)^{d+1}}.$$

The mth Maclaurin coefficient is given by

$$\frac{(d+1)\cdots(d+m)}{m!},$$

which is the number of integer points inside the simplex. It follows that

$$\#\{\boldsymbol{h} \in \mathbb{Z}^d : \|\boldsymbol{h}\|_1 > m \text{ and } \|\boldsymbol{h}\|_\infty \leq m\} \leq 2^d \left((m+1)^d - \frac{(d+1)\cdots(d+m)}{m!} \right),$$

which is sharp for $d = 2$. We now want

$$2^d \left((m+1)^d - \frac{(d+1)\cdots(d+m)}{m!} \right) p^{m+1} < 2\, p^m,$$

from which the stated result follows. □

Given such a choice of p we show that the squared worst-case error in the RKHS with kernel $K'_{m,p}$ gives us information on the trigonometric degree.

Lemma 2. *Given an N-point rank-1 lattice rule $Q(f; \boldsymbol{z}, N)$ with generating vector $\boldsymbol{z}$ then for integer $m > 1$ and $0 < p < 1$ chosen as in Lemma 1 we have*

0. *$e^2(\boldsymbol{z}, N; K'_{m,p}) = 0$ if Q has (product) trigonometric degree at least m;*
   and if $e^2(\boldsymbol{z}, N; K'_{m,p}) \neq 0$
1. *$\left\lfloor \log_p \frac{e^2(\boldsymbol{z}, N; K'_{m,p})}{2} \right\rfloor \leq m$ if Q has trigonometric degree less than m;*
2. *$\left\lfloor \log_p \frac{e^2(\boldsymbol{z}, N; K'_{m,p})}{2} \right\rfloor > m$ if Q has trigonometric degree at least m.*

Proof. The case of $e^2(\boldsymbol{z}, N; K'_{m,p}) = 0$ is trivial.

Now assume there are no non-zero dual lattice points for which $\|\boldsymbol{h}\|_1 \leq m$ then

$$e^2(\boldsymbol{z}, N; K'_{m,p}) = \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod N \\ \|\boldsymbol{h}\|_1 > m \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} \leq \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 > m \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{m+1} < 2\, p^m$$

due to Lemma 1.

On the other hand if there are non-zero dual lattice points with $\|\boldsymbol{h}\|_1 \leq m$ then

$$\begin{aligned} e^2(\boldsymbol{z}, N; K'_{m,p}) &= \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod N \\ 0 < \|\boldsymbol{h}\|_1 \leq m}} p^{\|\boldsymbol{h}\|_1} + \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod N \\ \|\boldsymbol{h}\|_1 > m \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} \\ &\geq \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod N \\ 0 < \|\boldsymbol{h}\|_1 \leq m}} p^{\|\boldsymbol{h}\|_1} \\ &\geq 2\, p^m. \end{aligned}$$

From these the result follows. □

Note that generally it will not be possible to check what the trigonometric degree is when it is larger than m. There is always the possibility of a $\boldsymbol{h} \in \Lambda^{\perp}$ such that $\|\boldsymbol{h}\|_1 = m + \ell$ but $\|\boldsymbol{h}\|_\infty > m$ for some $0 < \ell < m$: a dual lattice point outside of $[-m, m]^d$ but on a 1-norm iso-line through this hypercube. However, by modifying the choice of p we can determine what the trigonometric degree is when it is smaller than m, as stated in the following corollary.

Corollary 1. *Given an N-point rank-1 lattice rule Q with generating vector $\boldsymbol{z}$ then for integer $m > 1$ and $0 < p < 1$ chosen as*

$$\frac{1}{p} > 2^{d-1}\left((m+1)^d - (d+1)\right),$$

we have the additional property that

$$\left\lfloor \log_p \frac{e^2(\boldsymbol{z}, N; K'_{m,p})}{2} \right\rfloor = m - \ell + 1$$

if Q has trigonometric degree $m - \ell$ where $0 < \ell < m$.

Proof. Following the same reasoning as in the proof of Lemma 1, we find that for $\ell > 0$

$$\begin{aligned}\#\{\boldsymbol{h} \in \mathbb{Z}^d : \|\boldsymbol{h}\|_1 > m - \ell \text{ and } \|\boldsymbol{h}\|_\infty \le m\} \\ \le 2^d\left((m+1)^d - \frac{(d+1)\cdots(d+m-\ell)}{(m-\ell)!}\right).\end{aligned}$$

We now want for all possible $0 < \ell < m$

$$2^d\left((m+1)^d - \frac{(d+1)\cdots(d+m-\ell)}{(m-\ell)!}\right) p^{m+1-\ell} < 2\,p^{m-\ell},$$

from which the stated condition on p follows.

Using this condition, suppose the trigonometric degree is $m - \ell$ for some $0 < \ell < m$. Then we find

$$e^2(\boldsymbol{z}, N; K'_{m,p}) = \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z} \equiv 0 \pmod N \\ \|\boldsymbol{h}\|_1 > m-\ell \\ \|\boldsymbol{h}\|_\infty \le m}} p^{\|\boldsymbol{h}\|_1} \le \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 > m-\ell \\ \|\boldsymbol{h}\|_\infty \le m}} p^{m-\ell+1} < 2\,p^{m-\ell}$$

and

$$e^2(z, N; K'_{m,p}) \geq \sum_{\substack{\boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod N \\ \|\boldsymbol{h}\|_1 = m-\ell+1 \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} \geq 2\, p^{m-\ell+1}.$$

□

Above we have always lumped together the points to get weighted all by the same weight of p^{m+1}. A more careful analysis is possible if we weight each $\boldsymbol{h}$ exactly. This is possible, but we were unable to get such a nice expression as in Lemma 1. The following result could however be used in an algorithmic way to find a p greater than or equal to the one obtained by Lemma 1.

Lemma 3. *Given integer $m, d > 1$, if one chooses p such that*

$$\sum_{s=1}^{d} 2^s \binom{d}{s} \left(\left(\frac{p^{m+1} - p}{p-1} \right)^s - \sum_{k=1}^{m} p^k \binom{k-1}{s-1} \right) < 2\, p^m$$

then

$$\sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 > m \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} < 2\, p^m.$$

Proof. In similar spirit as the previous results we need

$$\sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} - \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_1 \leq m}} p^{\|\boldsymbol{h}\|_1} < 2\, p^m.$$

The weighted integer points inside the hypercube are easy to express as all sums are independent:

$$\sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \|\boldsymbol{h}\|_\infty \leq m}} p^{\|\boldsymbol{h}\|_1} = \left(\sum_{h=-m}^{m} p^{|h|} \right)^d = \left(1 + 2 \sum_{h=1}^{m} p^h \right)^d = \left(1 + 2\, \frac{p^{m+1} - p}{p-1} \right)^d.$$

This can also be written as

$$\begin{aligned} \left(1 + 2\, \frac{p^{m+1} - p}{p-1} \right)^d &= \sum_{s=0}^{d} \binom{d}{s} \left(2\, \frac{p^{m+1} - p}{p-1} \right)^s \\ &= 1 + \sum_{s=1}^{d} 2^s \binom{d}{s} \left(\frac{p^{m+1} - p}{p-1} \right)^s. \end{aligned}$$

For the weighted points inside the cross-polytope we have the number of points at distance k to be

$$\begin{aligned}\sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d\\ \|\boldsymbol{h}\|_1=k}} 1 &= \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d\\ \|\boldsymbol{h}\|_1\le k}} 1 - \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d\\ \|\boldsymbol{h}\|_1\le k-1}} 1\\ &= \sum_{s\ge 0} 2^s \binom{d}{s}\left(\binom{k}{s} - \binom{k-1}{s}\right)\\ &= \sum_{s\ge 1} 2^s \binom{d}{s}\binom{k-1}{s-1}\end{aligned}$$

for $k \geq 1$, cf. (6). As such, the weighted integer points inside the cross-polytope are given by

$$1 + \sum_{k=1}^{m} p^k \sum_{s\ge 1} 2^s \binom{d}{s}\binom{k-1}{s-1} = 1 + \sum_{s=1}^{d} 2^s \binom{d}{s}\left(\sum_{k=1}^{m} p^k \binom{k-1}{s-1}\right).$$

(Depending on the choice of d and m the sum over k might vanish partly or even completely because of the properties of the binomial coefficient, cf. (6).) From here the result follows. □

4 A Modification of the CKN Weighted-Degree Algorithm

The algorithm in [5] is a component-by-component algorithm, see, e.g., [25]. This means that one constructs the generating vector z one component at a time, first generating a one dimensional vector, then a two dimensional, etc, always keeping the previous choices fixed.

Using the results from the previous section, we can modify the algorithm from [5] as follows. Starting from a d-dimensional generating vector with trigonometric degree m_d, we "guess" (as explained below) the trigonometric degree $\tilde{m}_{d+1}$ that can be achieved in $d + 1$ dimensions. We then use kernel $K'_{\tilde{m}_{d+1},p}$, given in (8), with an appropriate choice for p, e.g., given by Lemma 1, to calculate the squared worst-case error for each possible choice of z_{d+1}. For this we consider all $z \in \mathbb{Z}_N^\times$ (where $\mathbb{Z}_N^\times$ are all positive integers relatively prime to N and smaller than N, i.e., the multiplicative group modulo N). This step might possibly be repeated for different choices of $\tilde{m}_{d+1}$ if our initial guess turned out to be incorrect, making use of Lemma 2. As we have chosen a tensor product form kernel, the calculation of the worst-case error for all possible choices $z \in \mathbb{Z}_N^\times$ can be done using Fast Fourier Transformations (FFTs) using the techniques from [20–22] in time $O(N \log N)$ for each guess of $\tilde{m}_{d+1}$. The final trigonometric degree that we settle on will be denoted by m_{d+1}.

As in [5] we try to achieve a good trigonometric degree *and* at the same time obtain an almost optimal worst-case error in a Korobov space. For this we also calculate the worst-case error using kernel K_α, see (4), and find the $z \in \mathbb{Z}_N^\times$ which minimizes this worst-case error and at the same time achieves the trigonometric degree m_{d+1} (found by the calculations based on $K'_{m_{d+1},p}$). The final choice of z is then fixed as z_{d+1}. The calculation of the worst-case error in the Korobov space might also be done using FFTs in time $O(N \log N)$.

When the number of points is sufficiently large, then [5, Theorem 3] shows that such lattice rules exist and can be found in a component by component way. For completeness we repeat that result here (which is stated for a prime number of points due to technicalities), slightly adjusted to the context of the (unweighted) trigonometric degree here. (The subsequent theorem also uses "product weights" γ_j to build a weighted function space. Such weighted spaces are a standard tool in tractability analysis but are of no real concern in this paper and can be safely ignored. Further information can be found in, e.g., [26].)

Theorem 1 (From [5, Theorem 3]). *Let $c > 1$ be fixed, m be given, and let N be a prime number satisfying*

$$N > \max\left(m, 1 + \frac{c}{c-1} \frac{|\mathscr{A}_{d+1}(m)| - |\mathscr{A}_d(m)| - 2m}{2}\right).$$

Suppose we already have a $\boldsymbol{z} \in (\mathbb{Z}_N^\times)^d$ for which

$$e^2(\boldsymbol{z}, N; K_m) = 0,$$

i.e., the rule has trigonometric degree at least m, and

$$e^2(\boldsymbol{z}, N; K_\alpha) \leq \left(\frac{c}{N-1} \prod_{j=1}^{d} \left(1 + 2\gamma_j^\lambda \zeta(\alpha\lambda)\right)\right)^{1/\lambda} \qquad \text{for all } \lambda \in (1/\alpha, 1],$$

i.e., the rule has near optimal worst-case error in the Korobov space with smoothness α. Then there is "at least one" $z_{d+1} \in \mathbb{Z}_N^\times$ such that we achieve trigonometric degree m

$$e^2((\boldsymbol{z}, z_{d+1}), N; K_m) = 0,$$

and near optimal worst-case error

$$e^2((\boldsymbol{z}, z_{d+1}), N; K_\alpha) \leq \left(\frac{c}{N-1} \prod_{j=1}^{d+1} \left(1 + 2\gamma_j^\lambda \zeta(\alpha\lambda)\right)\right)^{1/\lambda} \qquad \text{for all } \lambda \in (1/\alpha, 1].$$

This choice of N however was argued to be much higher than necessary, so in the practical implementation the condition was omitted. Here we will follow the same argument: we omit the condition on N and try to achieve the highest possible trigonometric degree possible. From [11] we note the known minimum number of points needed to achieve a prescribed trigonometric degree m in d dimensions:

$$N_{\min}(m,d) \geq \left|\mathscr{A}_d\left(\left\lfloor \frac{m}{2} \right\rfloor\right)\right| = \begin{cases} O(m^d), & \text{if } d \leq m, \\ O((2d)^{m/2}), & \text{if } m \leq d. \end{cases} \tag{9}$$

More specifically, the attainable lower bound in 2 dimensions, again, see [11], is given by

$$N_{\min}(m,2) = \begin{cases} 2k^2 + 2k + 1, & \text{for } m = 2k, \\ 2k^2 + 4k + 2, & \text{for } m = 2k + 1. \end{cases}$$

This brings us back to the "guessing" of the trigonometric degree. First note that the range of possible trigonometric degrees is quite limited. As an estimate in two dimensions we could use $m < \sqrt{2N}$. It follows that for a fixed N and increasing d (as in a component-by-component algorithm) the achievable trigonometric degree will decrease exponentially, see (9). This enables us to guess the trigonometric degree rather easily. To start off the process we use that the trigonometric degree in the first dimension always equals $N - 1$ (under the condition that z_1 is relatively prime to N), for the second dimension we can start from the explicit lower bound, i.e., guess $m < \sqrt{2N}$, and from then on we can assume exponential decrease. Moreover, if we never underestimate m, then by choosing p as in Corollary 1 we can determine the trigonometric degree from the squared worst-case error. Summarizing, we have the following algorithm:

Algorithm 1. For given $d_{\max}$, wanted degree $\hat{m}_{d_{\max}} \geq 1$ in $d_{\max}$ dimensions, $\alpha > 1$ and choosing $N \geq |\mathscr{A}_{d_{\max}}(\lfloor \hat{m}_{d_{\max}}/2 \rfloor)|$, then:

1. Set $z_1 = 1$.
2. For each $d = 1, \ldots, d_{\max} - 1$ with $\boldsymbol{z} = (z_1, \ldots, z_d)$ already fixed do the following:
   (a) Guess the trigonometric degree m_{d+1} (preferably do not underestimate), and choose a p small enough, e.g., as in Lemma 1 or Corollary 1.
   (b) For each possible component $z \in \mathbb{Z}_N^\times$, calculate $e^2((\boldsymbol{z}, z_{d+1}), N; K'_{m_{d+1},p})$. If there is no choice with trigonometric degree m_{d+1} then guess again and repeat this step, otherwise set the trigonometric degree m_{d+1}.
   (c) Set z_{d+1} to be the $z \in \mathbb{Z}_N^\times$ that minimizes $e^2((\boldsymbol{z}, z_{d+1}), N; K_\alpha)$ and has degree m_{d+1}.

Corollary 2. *Given Algorithm* 1*, we have that the complexity of construction up to d dimensions is $O(dN(\log N)^2)$.*

Proof. Assuming we need to repeat the guess T times, the cost per iteration is $O(TN \log N)$. As the relation between the number of points N and the achievable trigonometric degree is exponential for increasing d, see (9), we can assume $T = O(\log N)$ worst case. Consequently the cost is $O(N(\log N)^2)$ and the complexity of construction up to d dimensions is $O(dN(\log N)^2)$. □

We remark that the construction cost given assumes unit cost for all basic arithmetic operations on the computing device. Based on the smallness of p this will almost always mean arbitrary precision calculations for which this assumption is not quite correct (depending on the needed precision the deviation will become larger). An analysis of the practical implications for an actual implementation of this algorithm is therefore left for future research; but we make some developments in this area in the remainder of the paper. (We note that the full study of this would imply a numerical analysis of the computation of the worst-case error. As far as we know, this has not been studied yet.) To give an example of the technical complications: if the needed precision is very high, then it will become necessary to use FFTs which minimize the number of multiplications; or to use other algorithms to execute the underlying circular convolution.

We remark as well that in each iteration of the algorithm, we are in fact more or less computing the shortest vector in circa $N \approx |\mathbb{Z}_N^\times|$ dual lattices. So somewhere we expect to get bitten by the exponential complexity in d of the general problem of shortest vector computations. In that respect, the proposed algorithm looks quite good and it seems we can reduce the complexity by exploiting the specifics of our problem.

5 An Improvement on *p*

It is clear that our choice of p is far too conservative; it was a very crude underestimate based on a worst case argument. We simulated 10^3 random numbers N between 100 and 4,001, together with 5-dimensional integer vectors $\boldsymbol{z}$ with elements between 1 and N. Then, for each dimension between 2 and 5 we calculated the *enhanced trigonometric degree* explicitly (which we denote in this section by m for ease of notation), after which we checked for all $\boldsymbol{h}$ in $\{\boldsymbol{h} : \|\boldsymbol{h}\|_\infty \leq m \text{ and } \|\boldsymbol{h}\|_1 > m\}$ whether they satisfy $\boldsymbol{h} \cdot \boldsymbol{z} \equiv 0 \pmod N$. This gave us 10^3 trigonometric degrees and the corresponding number of dual lattice points in $\Lambda_m^\perp := \{\boldsymbol{h} : \boldsymbol{h} \in \Lambda^\perp, \|\boldsymbol{h}\|_\infty \leq m \text{ and } \|\boldsymbol{h}\|_1 > m\}$ for each dimension. In Table 1, the maximum $|\Lambda_m^\perp|$ that was found for each trigonometric degree encountered is reported. We also calculated the theoretical bound

$$\Omega_m = 2^d \left((m+1)^d - \frac{(d+1)\cdots(d+m)}{m!} \right)$$

used in Lemma 1 for constructing p to compare against the numerical experiment. It seems that Ω_m greatly overestimates the possible number of dual lattice points. More specifically, we have the following lemma for the two-dimensional case.

Lemma 4. *For $d = 2$, the maximum number of points in $\Lambda_m^\perp$ is bounded by 6 for any m.*

Proof. This lemma can be proven by noticing that for rank-1 lattice rules all dual lattice points lie equidistantly on equidistant parallel hyperplanes. □

This lemma illustrates, at least for $d = 2$, that the number of possible dual lattice points is fixed regardless of the trigonometric degree, whereas Ω_m, used for calculating p in our algorithm, increases with m. From Table 1 there seems to be some evidence that the maximum number of points in $\Lambda_m^\perp$ is much smaller than Ω_m also in higher dimensions.

Lemma 1 has been written in a general sense, without using any information on the actual underlying point set. If we specialize to rank-1 rules we can get a better estimate. We start from the following easy result.

Lemma 5. *Given an N-point rank-1 lattice rule with generating vector $\boldsymbol{z} \in (\mathbb{Z}_N^\times)^d$ modulo N, then there are N^{d-1} dual lattice points modulo N (i.e., in $[0, N)^d$).*

Proof. An integer point $\boldsymbol{h} \in \mathbb{Z}^d$ is part of the dual lattice if

$$h_1 z_1 + h_2 z_2 + \cdots + h_d z_d \equiv 0 \pmod{N}.$$

Now fix any choice of $h_j \in \mathbb{Z}_N$ except one, say h_1, then for $a = (h_2 z_2 + \cdots + h_d z_d)$ there is a unique solution, since $z_1 \in \mathbb{Z}_N^\times$, for h_1 in

$$h_1 z_1 + a \equiv 0 \pmod{N}.$$

The same conclusion could be drawn if fixing any other $d-1$ components. As there were N^{d-1} choices for the other h_j the dual lattice has N^{d-1} points in $[0, N)^d$. □

Similarly to the previous lemma we obtain an estimate for the dual lattice points inside $[-m, m]^d$.

Corollary 3. *Under the same conditions as for Lemma 5, there are $(2m + 1)^{d-1}$ dual lattice points in $[-m, m]^d$ for $m < N$.*

Note that this seems always smaller than Ω_m. However, this result is only valid for rank-1 lattice rules, whereas Lemma 1 remains valid for higher rank lattice rules. Therefore, we opted to keep Lemma 1 as a guideline although this estimate will get us a smaller p. A practical implementation for rank-1 rules could however make use of Corollary 3.

In closing this section we want to remark that, apart from making p larger by using a more careful analysis, we can also make p larger by sorting out bad cases as we go. E.g., the following lemma shows that as soon as we have fixed a component

Table 1 The enhanced trigonometric degrees found in a sample of random generating vectors up to 5 dimensions, together with the corresponding maximum number of dual lattice points $|\Lambda_m^\perp|$ found and the theoretical bound Ω_m, used for constructing p in Lemma 1.

	$d=2$		$d=3$		$d=4$		$d=5$	
m	$\|\Lambda_m^\perp\|$	Ω_m	$\|\Lambda_m^\perp\|$	Ω_m	$\|\Lambda_m^\perp\|$	Ω_m	$\|\Lambda_m^\perp\|$	Ω_m
2	2	12	10	136	14	1,056	62	7,104
3	0	24	8	352	88	3,536	312	30,976
4	4	40	20	720	122	8,880	630	95,968
5	4	60	22	1,280	136	18,720	1,416	240,768
6	4	84	20	2,072	174	35,056	1,350	523,040
7	4	112	32	3,136	202	60,256	1,298	1,023,232
8	4	144	36	4,512	250	97,056	1,346	1,848,384
9	4	180	28	6,240	222	148,560	1,560	3,135,936
10	4	220	32	8,360	210	218,240	1,508	5,057,536
11	4	264	36	10,912	222	309,936	1,660	7,822,848
12	4	312	34	13,936	222	427,856		
13	4	364	32	17,472	220	576,576		
14	6	420	30	21,560	210	761,040		
15	4	480	32	26,240				
16	6	544	32	31,552				
17	4	612	36	37,536				
18	6	684	38	44,232				
19	4	760	34	51,680				
20	4	840	32	59,920				
21	4	924	34	68,992				
22	4	1,012	34	78,936				
23	4	1,104	36	89,792				
24	4	1,200	36	101,600				
25	4	1,300	34	114,400				
26	4	1,404						
27	4	1,512						
28	4	1,624						
29	4	1,740						
30	4	1,860						
31	4	1,984						
32	4	2,112						
33	4	2,244						
34	6	2,380						
35	4	2,520						
36	6	2,664						
37	4	2,812						
38	6	2,964						
39	4	3,120						
40	4	3,280						

of z, then several multiples modulo N must not be considered again as valid choices in the next dimensions; as such we should not care about the actual p value we would need for these very bad rules.

Lemma 6. *Given an N-point rank-1 lattice rule with generating vector $z \in \mathbb{Z}_N^d$ in $d \geq 2$ dimensions, then as soon as there is a repeated component (modulo N) in z, the trigonometric degree is just 1. Moreover, if one component, say z_j, is $-t$ times a multiple of another (modulo $N/(z_i, N)$), say z_i, and $t \not\equiv 0 \pmod{N/(z_i, N)}$, then the trigonometric degree is at most t.*

Proof. We just prove the most general case. Consider the vector $\boldsymbol{h}$ which is zero everywhere except for the two components where $z_i \equiv a \pmod{N}$ and $z_j \equiv -t\,a \pmod{N/(a, N)}$. We get the equation

$$h_i z_i + h_j z_j \equiv 0 \pmod{N}$$

which, with (a, N) the greatest common divisor of a and N, is equivalent to

$$\frac{a}{(a, N)} h_i - \frac{t\,a}{(a, N)} h_j \equiv 0 \pmod{N/(a, n)},$$

where we have to assert that $t \not\equiv 0 \pmod{N/(a, n)}$ such that the problem still involves h_i and h_j. Multiplying by the multiplicative inverse of $a/(a, N)$ we obtain

$$h_i - t\,h_j \equiv 0 \pmod{N/(a, n)}$$

which clearly has a non-trivial solution $h_i = t$ and $h_j = 1$. It follows that the enhanced trigonometric degree is at most $\|\boldsymbol{h}\|_1 = t + 1$ and thus the trigonometric degree can be at most t. □

As a consequence of this last lemma we note that in Algorithm 1 as the algorithm progresses from dimension to dimension and as we have fixed N from the beginning—thus limiting the achievable trigonometric degree—the possible choices for the next z_{d+1} are much less than the elements of $\mathbb{Z}_N^\times$. We would hope that exploiting this knowledge would enable us to take much larger choices of p, as the bad choices will be the ones with the most points close to $\mathbf{0}$.

6 Conclusion and Future Work

We proposed a component-by-component algorithm to construct rules of good trigonometric degree by making use of a finite dimensional, exponentially decaying, reproducing kernel Hilbert space. The analysis of the algorithm has been tackled from an "existence" point of view, that is, we have proven that such an algorithm exist, and even explicitly given the algorithm outline, but we did not consider practical implementation aspects. Working out the technical details of the algorithm is of considerable complexity and left for future work. Some initial results in that direction have been included.

Acknowledgements The authors would like to thank the two anonymous referees for useful comments on the manuscript.

References

1. M. Beckers and R. Cools. A relation between cubature formulae of trigonometric degree and lattice rules. In H. Brass and G. Hämmerlin, editors, *Numerical integration IV (Oberwolfach, 1992)*, pages 13–24. Birkhäuser Verlag, 1993.
2. R. Cools. More about cubature formulas and densest lattice packings. *East Journal on Approximations*, 12(1):37–42, 2006.
3. R. Cools and H. Govaert. Five- and six-dimensional lattice rules generated by structured matrices. *J. Complexity*, 19(6):715–729, 2003.
4. R. Cools, F. Y. Kuo, and D. Nuyens. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comput.*, 28(6):2162–2188, 2006.
5. R. Cools, F. Y. Kuo, and D. Nuyens. Constructing lattice rules based on weighted degree of exactness and worst case error. *Computing*, 87(1–2):63–89, 2010.
6. R. Cools and J. N. Lyness. Three- and four-dimensional K-optimal lattice rules of moderate trigonometric degree. *Math. Comp.*, 70(236):1549–1567, 2001.
7. R. Cools, E. Novak, and K. Ritter. Smolyak's construction of cubature formulas of arbitrary trigonometric degree. *Computing*, 62(2):147–162, 1999.
8. R. Cools and D. Nuyens. A Belgian view on lattice rules. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 3–21. Springer-Verlag, 2008.
9. R. Cools and D. Nuyens. Extensions of Fibonacci lattice rules. In P. L'Écuyer and A. B. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 1–12. Springer-Verlag, 2009.
10. R. Cools and A. V. Reztsov. Different quality indexes for lattice rules. *J. Complexity*, 13(2):235–258, 1997.
11. R. Cools and I. H. Sloan. Minimal cubature formulae of trigonometric degree. *Math. Comp.*, 65(216):1583–1600, 1996.
12. J. A. De Loera, J. Rambau, and F. Santos. *Triangulations*, volume 25 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2010.
13. J. Dick, F. Pillichshammer, G. Larcher, and H. Woźniakowski. Exponential convergence and tractability of multivariate integration for Korobov spaces. *Math. Comp.*, 80(274):905–930, 2011.
14. I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series and Products*. Academic Press, 7th edition, 2007.
15. F. J. Hickernell. Lattice rules: How well do they measure up? In P. Hellekalek and G. Larcher, editors, *Random and Quasi-Random Point Sets*, pages 109–166. Springer-Verlag, Berlin, 1998.
16. J. N. Lyness. Notes on lattice rules. *J. Complexity*, 19(3):321–331, 2003.
17. J. N. Lyness and T. Sørevik. Four-dimensional lattice rules generated by skew-circulant matrices. *Math. Comp.*, 73(245):279–295, 2004.
18. J. N. Lyness and T. Sørevik. Five-dimensional K-optimal lattice rules. *Math. Comp.*, 75(255):1467–1480, 2006.
19. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Number 63 in Regional Conference Series in Applied Mathematics. SIAM, 1992.

20. 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(254):903–920, 2006.
21. D. Nuyens and R. Cools. Fast component-by-component construction, a reprise for different kernels. In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 371–385. Springer-Verlag, 2006.
22. D. Nuyens and R. Cools. Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. *J. Complexity*, 22(1):4–28, 2006.
23. N. N. Osipov, R. Cools, and M. V. Noskov. Extremal lattices and the construction of lattice rules. *Appl. Math. Comput.*, 217(9):4397–4407, 2011.
24. I. H. Sloan and S. Joe. *Lattice Methods for Multiple Integration*. Oxford Science Publications, 1994.
25. I. H. Sloan and A. V. Reztsov. Component-by-component construction of good lattice rules. *Math. Comp.*, 71(237):263–273, 2002.
26. I. H. Sloan and H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *J. Complexity*, 14(1):1–33, 1998.

Scrambled Polynomial Lattice Rules for Infinite-Dimensional Integration

Jan Baldeaux

Abstract In the random case setting, scrambled polynomial lattice rules, as discussed in Baldeaux and Dick (Numer. Math. 119:271–297, 2011), enjoy more favorable strong tractability properties than scrambled digital nets. This short note discusses the application of scrambled polynomial lattice rules to infinite-dimensional integration. In Hickernell et al. (J Complex 26:229–254, 2010), infinite-dimensional integration in the random case setting was examined in detail, and results based on scrambled digital nets were presented. Exploiting these improved strong tractability properties of scrambled polynomial lattice rules and making use of the analysis presented in Hickernell et al. (J Complex 26:229–254, 2010), we improve on the results that were achieved using scrambled digital nets.

1 Introduction

In a recent series of papers, [2, 4, 6, 7, 9, 11, 13], the problem of infinite-dimensional quadrature has been studied. Such problems have many applications, e.g. in mathematical finance, see [8, 10], where series expansions are used to represent particular random variables.

This short note focuses on the random case setting, which was addressed in [4]. In the latter paper, a complete and general analysis was presented, clearly showing the reader how to employ a given quadrature rule. Furthermore, results based on the scrambled Niederreiter sequence were presented. Recently, [1], it was shown that for multivariate integration in the random case setting, scrambled polynomial lattice rules possess more favorable strong tractability properties than any scrambled digital sequence.

J. Baldeaux
School of Finance and Economics, University of Technology, Sydney, NSW, Australia
e-mail: JanBaldeaux@gmail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_11,

This raises the natural question, whether the results based on the scrambled Niederreiter sequence can be improved on using scrambled polynomial lattice rules. In this short note, we give an affirmative answer to this question. In particular, the contribution of this note is the following: Using the analysis presented in [4], we employ the multivariate integration result from [1] to improve on the results presented in [4]. Finally, regarding lattice rules, we remark that results on multivariate integration using lattice rules in the random case setting have not appeared in the literature.

We place ourselves in the same setting as discussed in [4], use the same algorithms, but employ a different quadrature rule, in particular one with better strong tractability properties. In the interest of giving due credit to the authors of [4] and also of saving space, we have decided to proceed as follows: We very briefly recall the function space introduced in [4] and the sampling regimes, but introduce cost and worst-case errors under the premise that algorithms are based on scrambled polynomial lattice rules. The interested reader is referred to [4] for a complete and general treatment of the problem studied in this note.

2 The Setting

In this section, we briefly recall the function space and the sampling regimes, cost and errors as introduced in [4]. Regarding notation, as in [4], v is used to denote finite subsets of $\mathbb{N}$, the set $\{1, \ldots, s\}$ is denoted by $1:s$, and lastly, we write $x_k \preceq y_k$ for sequences of positive real numbers x_k and y_k, if $x_k \leq c y_k$ is valid for $k \in \mathbb{N}$ and some constant $c > 0$.

2.1 The Function Space

We briefly remind the reader how to construct functions of infinitely many variables, as presented in [4]. Essentially, we start with a one-dimensional reproducing kernel Hilbert space, construct spaces of finitely many variables as tensor product spaces and take limits to allow for infinitely many variables. Coordinate weights $\gamma_v = \prod_{j \in v} \gamma_j$, $v \subset \mathbb{N}$, which indicate the importance of the variables x_j, $j \in v$, ensure convergence of the relevant quantities.

In particular, we consider the reproducing kernel

$$k(x, y) = \frac{1}{3} + (x^2 + y^2)/2 - \max(x, y),$$

$x, y \in [0, 1]$, and consider the Hilbert space $H(1 + \gamma k)$, for a weight $\gamma > 0$, whose norm satisfies

$$\|f\|^2 = \left(\int_0^1 f(y)dy\right)^2 + \gamma^{-1}\int_0^1 (f')^2(y)dy.$$

To allow for functions of finitely many variables, we consider the reproducing kernel

$$K_v(x,y) = \prod_{j\in v}(1+\gamma_j k(x_j,y_j))$$

and of course the associated Hilbert space $H(K_v)$ is of tensor product form

$$H(K_v) = \bigotimes_{j\in v} H(1+\gamma_j k). \tag{1}$$

To define functions of infinitely many variables, we define the measurable kernel K on $[0,1]^{\mathbb{N}} \times [0,1]^{\mathbb{N}}$

$$K(x,y) = \sum_v \gamma_v K_v(x,y) = \sum_v \gamma_v \prod_{j\in v} k(x_j,y_j),$$

for $x, y \in [0,1]^{\mathbb{N}}$ and denote the associated space by $H(K)$, which, see [4, Lemma 6], consists of all functions

$$f = \sum_v f_v, \quad f_v \in H_v,$$

for which

$$\sum_v \gamma_v^{-1}\|f_v\|^2_{k_v} < \infty,$$

and, in case of convergence,

$$\|f\|^2_K = \sum_v \gamma_v^{-1}\|f\|^2_{k_v}.$$

2.2 Sampling Regimes, Cost, and Worst-Case Error

In this subsection, we introduce randomized algorithms for the integration of functions $f : [0,1]^{\mathbb{N}} \to \mathbb{R}$; the reader is referred to [2, 15] for a detailed discussion. For the remainder of the note, we assume that $\gamma_j \asymp j^{-\alpha}$, $\alpha > 1$. Following [2, 4], two sampling regimes, which specify the domains from which the integration nodes can be chosen, are introduced.

Fixed subspace sampling restricts this domain to a finite-dimensional affine subspace

$$\mathscr{X}_{v,a} = \left\{x \in [0,1]^{\mathbb{N}} : x_j = a \text{ for } j \in \mathbb{N} \setminus v\right\}$$

for a finite set $\emptyset \neq v \subset \mathbb{N}$ and $a \in [0,1]$. Since we will employ scrambled polynomial lattice rules, we will deal with the case $v = 1:s$. We remind the reader that essentially one only specifies those coordinates included in v, the remaining ones are specified via the anchor point a.

Variable subspace sampling generalizes this idea to a sequence of finite-dimensional affine subspaces

$$\mathscr{X}_{v_1,a} \subset \mathscr{X}_{v_2,a} \subset \ldots,$$

where $v = (v_i)_{i \in \mathbb{N}}$ is a given increasing sequence, $v_i \subset \mathbb{N}$, and $a \in [0,1]$; again, the case $v_i = 1 : s_i$ will turn out to be relevant when dealing with scrambled polynomial lattice rules. This sampling scheme allows us to choose integration nodes from subspaces of different dimensionality. To be able to compare algorithms, we want to be able to quantify how costly it is to evaluate the integrand f at the integration nodes. Following [2, 4], we formulate the cost of evaluating f at the integration node x in terms of the dimension of the finite-dimensional subspace from which the integration node is chosen. This means for fixed subspace sampling we obtain the cost

$$c_{v,a} = \begin{cases} |v|, & \text{if } x \in \mathscr{X}_{v,a} \\ \infty, & \text{otherwise.} \end{cases} \tag{2}$$

For variable subspace sampling, which allows for the integration nodes to be chosen from a sequence of finite-dimensional subspaces, we choose the subspace with the smallest dimension in which the node lies, to obtain the cost

$$c_{v,a}(x) = \inf\{\dim(\mathscr{X}_{v_i,a}) : x \in \mathscr{X}_{v_i,a}\} \tag{3}$$

and set $\inf \emptyset = \infty$.

The randomized quadrature formulas employed in this note are based on scrambled polynomial lattice rules,

$$Q_{m,b,1:s}(f) = \frac{1}{b^m} \sum_{i=1}^{b^m} f(x_i), \tag{4}$$

where $x_i \in [0,1)^s$ are obtained by scrambling a polynomial lattice rule, see [1]. Defining

$$(\Psi_{v,a} f)(x) = f(x_v, a), \tag{5}$$

we denote the randomized quadrature formulas of interest in this note by

$$Q_{n,s,a} = Q_{\lfloor \log_b(n) \rfloor, b, 1:s} \circ \Psi_{1:s,a}, \tag{6}$$

where n denotes the number of points the quadrature rule is comprised of. Intuitively speaking, we carefully specify dimensions 1 to s via scrambled polynomial lattice rules and employ the anchor $a \in [0, 1]$ for the subsequent dimensions. It is clear from the previous discussion, that for fixed subspace sampling, the notation introduced in Eq. 6 is sufficient. For variable subspace sampling, however, we allow our integration nodes to be chosen from subspaces of different dimensions, in this sense the letter s is not sufficient. We remark that this is addressed in Sect. 4.

Next, we wish to discuss the cost of the randomized algorithms. As we only discuss randomized algorithms based on scrambled polynomial lattice rules in this note, we define the cost of fixed and variable subspace sampling under the premise that the randomized algorithm is based on a scrambled polynomial lattice rule. This simplifies the discussion, the cost model employed in [4], which stems from [2], allows for a much more general class of algorithms, see also [9] for an even more general cost model.

Essentially, the cost of evaluating a randomized algorithm Q is given by the sum of the costs of evaluating the function at the integration nodes chosen from the finite-dimensional subspaces. For fixed subspace sampling, assuming that nodes are chosen from $\mathscr{X}_{1:s,a}$

$$\mathrm{cost}_{\mathit{fix}}(Q) = ns.$$

For variable subspace sampling, we choose our integration nodes from a sequence of finite-dimensional affine subspaces, indexed by $(v_i)_{i=1}^m$, where $m \leq n$, as we use an n-point quadrature rule, where $v_i = 1 : s_i$, $i = 1, \ldots, m$. For the subspace indexed by v_i, the integration nodes would be based on scrambled polynomial lattice rules whose integration nodes lie in $[0, 1)^{s_i}$, and we denote the number of those integration nodes by n_{v_i}, where of course $\sum_{i=1}^m n_{v_i} = n$. Consequently, we have

$$\mathrm{cost}_{var}(Q) = \sum_{i=1}^{m} s_i n_{v_i}. \tag{7}$$

We use $B(K)$ to denote the unit ball in $H(K)$, and remark that all integrands considered in this note lie in $B(K)$. For $f \in B(K)$, we use the notation

$$I(f) = \int_{\mathscr{X}} f(x)dx,$$

where $\mathscr{X} \subset [0, 1]^{\mathbb{N}}$, and denote the worst-case error of a randomized algorithm Q, used to approximate integrands f in $B(K)$, by

$$e(Q, B(K)) = \sup_{f \in B(K)} \left(E\,(I(f) - Q(f))^2 \right)^{1/2}.$$

Lastly, minimal errors, which are of great importance in information-based complexity, [12, 14, 15], are defined by

$$e_{N,fix}(B(K)) = \inf\left\{e(Q, B(K)) : \ \mathrm{cost}_{fix}(Q, B(K)) \leq N\right\}$$

and

$$e_{N,var}(B(K)) = \inf\{e(Q, B(K)) : \ \mathrm{cost}_{var}(Q, B(K)) \leq N\}.$$

The following result on numerical integration in $H(K_{1:s})$, see Eq. 1, stems from [1].

Theorem 1. *Assume* $\sum_{j=1}^{\infty} \gamma_j^{\frac{1}{3-2\varepsilon}} < \infty$*, for* $0 < \varepsilon \leq 1$*, then*

$$e\left(Q_{b,m,1:s}, H(K_{1:s})\right) \leq c_\varepsilon n^{-(3/2-\varepsilon)},$$

where $n = b^m$ *and* $Q_{b,m,1:s}$ *is a scrambled polynomial lattice rule as defined in Eq. 4.*

Proof. From the proof of [16, Lemma 7], it is clear that the function space $H(K_{1:s})$ can be embedded in the space $V_{1,s,\gamma}$, as defined in [1], from which the result follows immediately.

3 Results on Fixed Subspace Sampling

To fully specify the fixed subspace sampling algorithm, we only need to specify the dimension of the finite-dimensional subspace employed for sampling, and the number of integration nodes, which are based on a scrambled polynomial lattice rule. As we wish to minimize worst-case errors for a fixed bound on the cost, say N, both, the dimension and the number of integration nodes, are functions of N.

Corollary 1 (Corollary 1, [4]). *Let* $0 < \varepsilon \leq 1$*,* $\gamma_j \asymp j^{-\alpha}$*,* $\alpha \geq 3$*, and* $a \in [0, 1]$*. Choose*

$$n \asymp N^{\frac{\alpha-1}{\alpha+2-\varepsilon}}$$

and

$$s \asymp N^{\frac{3-\varepsilon}{\alpha+2-\varepsilon}}$$

for $N \in \mathbb{N}$*. Then, for* $Q_N = Q_{n,s,a}$

$$e\left(Q_N, B(K)\right) \preceq N^{-\frac{(3-\varepsilon)/2(\alpha-1)}{\alpha+2-\varepsilon}}$$

and

$$cost_{fix}(Q_N, B(K)) \preceq N.$$

Proof. The result follows immediately from [4, Theorem 1], where we set $\alpha \geq 3$.

Remark 1. In [4], the same result was established under the stronger assumption on the weights $\gamma_j \asymp j^{-\alpha}$, $\alpha > 4$. The result presented in Corollary 1 is optimal for $\alpha \geq 3$, see [4, Corollary 3].

4 Results on Variable Subspace Sampling

We carry out variable subspace sampling using the so-called multi-level approach, which was first introduced in [3], see also [5]. The idea underlying the multi-level approach is the following: We fix a sequence of sets

$$v_1 \subset \cdots \subset v_L$$

and the associated finite-dimensional affine subspaces

$$\mathscr{X}_{v_1} \subset \cdots \subset \mathscr{X}_{v_L}.$$

We use the integral associated with the finite-dimensional subspace of the largest dimension, $I(\Psi_{v_L,a} f)$ to approximate $I(f)$. However, we rewrite $I(\Psi_{v_L,a} f)$ as follows

$$I(\Psi_{v_L,a} f) = \sum_{l=1}^{L} I\left(\Psi_{v_l,a} f - \Psi_{v_{l-1},a} f\right)$$

setting $\Psi_{v_0,a} f = 0$. Each of the integrals $I(\Psi_{v_l,a} f - \Psi_{v_{l-1},a} f)$ is now approximated using an independent randomized algorithm, based on a scrambled polynomial lattice rule, in particular, we use a randomized algorithm

$$Q(f) = \sum_{l=1}^{L} Q_{n_l,s_l,a}(f - \Psi_{1:s_{l-1},a} f), \tag{8}$$

so at level l, we use an algorithm based on a scrambled polynomial lattice rule consisting of $b^{\lfloor \log_b(n_l) \rfloor}$ points, which lie in $[0,1)^{s_l}$. The error associated with this algorithm can be split into bias and variance,

$$E\left(I(f) - Q(f)\right)^2 = \left(I(f) - I(\Psi_{1:s_L,a} f)\right)^2 + \mathrm{Var}(Q(f)),$$

in particular

$$\mathrm{Var}(Q(f)) = \sum_{l=1}^{L} \mathrm{Var}\left(Q_{n_l,s_l,a}(f - \Psi_{1:s_{l-1},a} f)\right),$$

see [4]. Regarding the cost, from Eq. 7,

$$\mathrm{cost}_{\,var}(Q, B(K)) = \sum_{l=1}^{L} s_l n_l.$$

By definition of variable subspace sampling, the dimension s_l increases with l, but one would expect the variances $\mathrm{Var}\left(Q_{n_l,s_l,a}(f - \Psi_{1:s_{l-1},a} f)\right)$ to decrease as l increases; the challenge is to trade off these effects well.

Corollary 2 (Corollary 4, [4]). *Assume that* $\gamma_j \asymp j^{-\alpha}$*, for* $\alpha > 3$*, let* $0 < \varepsilon < \min(1, \alpha - 3)$ *and put*

$$\rho_1 = \frac{\alpha - 1}{3 - \varepsilon/2}, \quad \rho_2 = \frac{\alpha - 4 - \varepsilon}{3 - \varepsilon/2}.$$

Choose L, s_l*,* n_l *according to Eqs. 26–28 in [4], and let* $a \in [0,1]$*. Take the corresponding multi-level algorithm* Q_N *according to Eq. 8 based on the scrambled polynomial lattice rule. Then*

$$e(Q_N, B(K)) \preceq \begin{cases} N^{-(3-\varepsilon)/2}, & \mathit{if}\, \alpha \geq 10, \\ N^{-(3-\varepsilon)/2 \frac{\alpha-1}{9}}, & \mathit{if}\, \alpha < 10, \end{cases}$$

and

$$cost_{var}(Q_N, B(K)) \preceq N.$$

Proof. The proof follows immediately from [4, Theorem 4], with $\alpha' = 3 + \varepsilon$.

Remark 2. The same error bounds were established in [4], but the rate $N^{(3-\varepsilon)/2}$ was only established for $\alpha \geq 11$, whereas here it is achieved for $\alpha \geq 10$, due to an improved strong tractability result. Of course, for $\alpha \geq 10$, this result is optimal. Furthermore, we conclude that for (at least) $\alpha > 7$, variable subspace sampling improves on fixed subspace sampling.

Remark 3. We alert the reader to [6, 11], where infinite-dimensional integration in the worst-case setting is studied. In [6, 11], rank-1 lattice rules are employed as a basis for the algorithms, and we remark that in the worst-case setting, polynomial lattice rules have not been shown to improve on rank-1 lattice rules.

Remark 4. For both, fixed subspace and variable subspace sampling, this note provided optimal convergence rates assuming that $\alpha \geq 3$ and $\alpha \geq 10$, respectively. It is not known if these assumptions on α are optimal in general, it is not even known if these assumptions on α are optimal for scrambled polynomial lattice rules. Furthermore, since we construct polynomial lattice rules using the component-by-component algorithm, the resulting subspaces for the variable subspace sampling regime are necessarily nested and satisfy $v_i = 1 : s_i$. Whereas the model presented in [2, 4] requires the subspaces to be nested, it would be interesting to check if one could weaken the assumption on α by choosing different v_i.

References

1. Baldeaux, J., Dick, J., A construction of polynomial lattice rules with small gain coefficients, Numerische Mathematik, 119, 271–297, 2011.
2. Creutzig, J., Dereich, S., Müller-Gronbach, T., Ritter, K., Infinite-dimensional quadrature and approximation of distributions, Foundations of Computational Mathematics, 9, 391–429, 2009.
3. Heinrich, S., Monte Carlo complexity of global solution of integral equations, Journal of Complexity, 14, 151–175, 1998.
4. Hickernell, F.J., Müller-Gronbach, T., Niu, B., Ritter, K., Multi-level Monte Carlo Algorithms for Infinite-Dimensional Integration on $\mathbb{R}^{\mathbb{N}}$, Journal of Complexity, 26, 229–254, 2010.
5. Giles, M.B., Multilevel Monte Carlo path simulation, Operations Research, 56, 607–617, 2008.
6. Gnewuch, M., Infinite-dimensional Integration on Weighted Hilbert Spaces, Mathematics of Computation, 2012.
7. Gnewuch, M., Weighted geometric discrepancies and numerical integration on reproducing kernel Hilbert spaces, Journal of Complexity 28, 2–17, 2012.
8. Imai, J., Kawai, R., Quasi-Monte Carlo Method for Infinitely Divisible Random Vectors via Series Representations, SIAM Journal on Scientific Computing, 32, 1879–1897, 2010.
9. Kuo, F.Y., Sloan, I.H., Wasilkowski, G.W., Woźniakowski, H., Liberating the dimension, Journal of Complexity, 26, 422–454, 2010.
10. Niu, B., Hickernell, F.J., *Monte Carlo simulation of stochastic integrals when the cost of function evaluation is dimension dependent*, Monte Carlo and Quasi-Monte Carlo Methods 2008 (P. L'Ecuyer and A.Owen, eds.), Springer-Verlag, Berlin, 545–560, 2010.
11. Niu, B., Hickernell, F.J., Müller-Gronbach, T., Ritter, K., Deterministic Multi-level Algorithms for Infinite-Dimensional Integration on $\mathbb{R}^{\mathbb{N}}$, Journal of Complexity, 26, 229–254, 2010.
12. Novak, E., Deterministic and stochastic error bounds in numerical analysis, Lecture Notes in Mathematics, 1349, Springer-Verlag, Berlin, 1988.
13. Plaskota, L., Wasilkowski, G.W., Tractability of infinite-dimensional integration in the worst case and randomized settings, Journal of Complexity, 27, 505–518, 2011.
14. Ritter, K., Average-case analysis of numerical problems, Lecture Notes in Mathematics, 1733, Springer-Verlag, Berlin, 2000.
15. Traub, J., Wasilkowski, G.W., Woźniakowski, H., Information-based Complexity, Academic Press, New York, 1988.
16. Yue, R.-X., Hickernell, F.J., Strong tractability of integration using scrambled Niederreiter points, Mathematics of Computation, 74, 1871–1893, 2005.

Geometric and Statistical Properties of Pseudorandom Number Generators Based on Multiple Recursive Transformations

L.Yu. Barash

Abstract The equidistribution property is studied for the generators of the MRG type. A new algorithm for generating uniform pseudorandom numbers is proposed. The theory of the generator, including detailed study of its geometric and statistical properties, in particular, proofs of periodic properties and of statistical independence of bits at distances up to logarithm of mesh size, is presented. Extensive statistical testing using available test packages demonstrates excellent results, while the speed of the generator is comparable to other modern generators.

1 Introduction

Pseudorandom number generation is an important component of any stochastic simulations such as molecular dynamics and Monte Carlo simulations [4, 5, 15, 23, 27]. The problem of design of reliable and fast generators is of great importance and attracts much attention [14].

The present approach extends the method of pseudorandom number generation of Ref. [1, 2], which is based on evolution of the ensemble of dynamical systems (see Sect. 2). Several generalizations are carried out. The connection between the statistical properties of a generator and geometric properties of the corresponding map is uncovered. New pseudorandom number generators are proposed. Using SSE2 technology, which is supported by all Intel and AMD processors fabricated later than in 2003 [13, 32], effective implementations are developed.

Among several statistical test suites available in the literature, TestU01 is known to contain very stringent batteries of tests for empirical testing of pseudorandom numbers. At present there are not so many pseudorandom number generators that pass all the tests even in the sense that no p-value is outside the interval

L. Yu. Barash
Landau Institute for Theoretical Physics, 142432, Chernogolovka, Russia
e-mail: barash@itp.ac.ru

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_12, © Springer-Verlag Berlin Heidelberg 2012

$[10^{-10}, 1 - 10^{-10}]$ [22]. In Sect. 3 it is shown that statistical testing with TestU01 confirms excellent statistical properties of the proposed realizations.

One of the most important properties that characterize the quality of pseudorandom sequence of numbers, is high-dimensional uniformity and the corresponding equidistribution property [7, 9, 17, 33, 34]. In contrast to other important characteristics of pseudorandom number generators such as the period length, which is studied in detail for almost all known generators, there are only a few examples when high-dimensional equidistribution property was proved [7, 9, 17, 19, 24, 26, 33, 34].

In this paper the proper choice of parameters is established, which results in the validity of the equidistribution property for the proposed generator. In particular, it is shown that the determinant of the transformation has to be an even integer in order for the property to hold. This signifies that applying dissipative dynamical systems to pseudorandom number generation can result in substantially preferable statistical behavior of the corresponding pseudorandom number sequences, compared to applying conservative dynamical systems. The equidistribution is established on length up to a characteristic length ℓ: for $n \leq \ell$, each combination of successive n bits taken from the RNG output occurs exactly the same number of times and has a corresponding probability $1/2^n$. The length ℓ turns out to depend linearly on t, where the mesh size g (i.e. the modulus of the basic recurrence) is equal to $p \cdot 2^t$ and p is an odd prime (see Propositions 7 and 8 in Sect. 5). In other words, for given p, one has $\ell \propto \log g$. Numerical results show that the equidistribution property still approximately holds with high accuracy beyond the region of its strict validity under the condition $n < 6.8 \log p$.

I have constructed several realizations for the proposed generator (see Table 1). It is shown in Sect. 5 that for the realizations either $\ell = 2t - 1$ or $\ell = (t - 1)/2$ takes place. The speed and statistical properties of the constructed generators are compared with those of other modern generators (see Tables 1 and 2). Practically, the generators with smaller values of t (e.g. with prime g) also have very good properties for a particular choice of parameters, while the generator period is not less than $p^2 - 1$ and increases significantly with increasing p. For this reason two realizations with small t are also thoroughly tested.

The paper is organized as follows. The generator is introduced in Sect. 2. In Sect. 3, the results of speed tests and statistical tests are presented. In Sect. 4 the geometric and statistical properties of transformations with unitary determinant associated with hyperbolic automorphisms of two-dimensional torus, are studied. The five-dimensional equidistribution never takes place in this case. In Sect. 5 the choice of parameters is established for the high-dimensional equidistribution property to hold for the proposed generator. The main new results of the present paper are contained in Sects. 4 and 5. They complement the results of the short letter [3].

Table 1 Parameters of the new generators and numbers of failed tests for the batteries of tests SmallCrush, Crush, BigCrush [21], and Diehard [21]. Testing was performed with package TestU01 version TestU01-1.2.3. For each battery of tests, we present three numbers: the number of statistical tests with p-values outside the interval $[10^{-3}, 1-10^{-3}]$, number of tests with p-values outside the interval $[10^{-5}, 1-10^{-5}]$, and number of tests with p-values outside the interval $[10^{-10}, 1-10^{-10}]$.

Generator	g	k	q	v	Period	SmallCrush	Diehard	Crush	BigCrush
GM29.1-SSE	$2^{29}-3$	4	2	1	$= 2.8 \cdot 10^{17}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
GM55.4-SSE	$16(2^{51}-129)$	256	176	4	$\geq 5.1 \cdot 10^{30}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
GQ58.1-SSE	$2^{29}(2^{29}-3)$	8	48	1	$\geq 2.8 \cdot 10^{17}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
GQ58.3-SSE	$2^{29}(2^{29}-3)$	8	48	3	$\geq 2.8 \cdot 10^{17}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
GQ58.4-SSE	$2^{29}(2^{29}-3)$	8	48	4	$\geq 2.8 \cdot 10^{17}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
MRG32k3a	–	–	–	–	$= 3.1 \cdot 10^{57}$	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
LFSR113	–	–	–	–	$= 1.0 \cdot 10^{34}$	0, 0, 0	1, 0, 0	6, 6, 6	6, 6, 6
MT19937	–	–	–	–	$= 4.3 \cdot 10^{6001}$	0, 0, 0	0, 0, 0	2, 2, 2	2, 2, 2

Table 2 CPU time (s) for generating 10^9 random numbers. Processors: Intel Core i7-940 and AMD Turion X2 RM-70. Compilers: gcc 4.3.3, icc 11.0.

Intel Core i7-940	gcc -O0	gcc -O1	gcc -O2	gcc -O3	icc -O0	icc -O1	icc -O2	icc -O3	Source
MT19937	13.7	5.7	6.9	2.6	17.5	6.5	2.9	2.9	[24]
MT19937-SSE	5.2	4.8	5.5	2.0	4.9	4.7	2.4	2.0	[2]
LFSR113	10.4	4.8	6.8	3.1	10.2	5.0	4.6	4.5	[19]
LFSR113-SSE	8.0	6.8	6.8	6.9	7.3	6.9	6.6	6.5	[2]
MRG32k3a	47.9	36.3	35.3	25.0	56.1	33.1	22.8	28.1	[20]
MRG32k3a-SSE	9.1	7.4	5.8	5.8	8.8	7.4	6.0	5.9	[2]
GM29.1-SSE	22.6	19.6	17.5	18.1	21.2	18.7	18.2	18.1	[6]
GM55.4-SSE	18.0	16.8	15.4	15.4	17.7	16.3	15.8	15.7	[6]
GQ58.1-SSE	50.5	49.2	47.4	47.3	50.5	48.1	48.0	47.7	[6]
GQ58.3-SSE	22.0	21.2	19.0	20.1	22.5	20.4	19.5	19.5	[6]
GQ58.4-SSE	16.1	14.7	12.8	13.8	15.5	13.9	13.3	13.3	[6]
AMD Turion X2 RM-70									
MT19937	31.0	17.8	10.8	7.1	31.0	18.7	5.2	4.9	[24]
MT19937-SSE	11.3	10.3	11.1	6.6	10.8	9.9	6.0	6.0	[2]
LFSR113	14.6	8.7	9.6	5.3	14.9	9.1	6.9	6.8	[19]
MRG32k3a	89.0	60.9	60.9	47.0	89.1	69.2	41.5	41.6	[20]
MRG32k3a-SSE	25.9	22.3	18.4	18.3	25.6	22.3	19.0	19.0	[2]
GM29.1-SSE	68.5	64.4	60.7	60.7	67.8	63.1	61.7	61.7	[6]
GM55.4-SSE	59.8	54.8	53.1	53.0	58.2	53.6	52.8	52.8	[6]
GQ58.1-SSE	179.6	179.6	178.3	177.8	183.1	178.3	178.5	178.5	[6]
GQ58.3-SSE	75.5	73.9	70.6	71.1	74.2	71.9	70.4	70.1	[6]
GQ58.4-SSE	51.9	51.0	48.2	48.1	53.1	49.4	48.2	48.1	[6]

2 The Generator, Its Initialization and Period

It is suggested in [1, 2] to construct RNGs based on an ensemble of sequences generated by multiple recursive method. The state of the generator consists of the values $x_i^{(n-1)}, x_i^{(n-2)} \in \mathbb{Z}_g$, $i = 0, 1, \ldots, s-1$. The transition function of the generator is defined by the recurrence relation

$$x_i^{(n)} = kx_i^{(n-1)} - qx_i^{(n-2)} \pmod{g}, \tag{1}$$

where $i = 0, 1, \ldots, s-1$. The values $x_i^{(n-1)}, i = 0, 1, \ldots, s-1$ can be considered as x-coordinates of s points $(x_i^{(n-1)}, y_i^{(n-1)})^T, i = 0, 1, \ldots, s-1$ of the $g \times g$ lattice on the two-dimensional torus, then each recurrence relation (1) describes the dynamics of x-coordinate of a point on the two-dimensional torus:

$$\begin{pmatrix} x_i^{(n)} \\ y_i^{(n)} \end{pmatrix} = M \begin{pmatrix} x_i^{(n-1)} \\ y_i^{(n-1)} \end{pmatrix} \pmod{g}, \tag{2}$$

where matrix $M = \binom{m_1\ m_2}{m_3\ m_4}$ is a matrix with integer elements, and $k = \operatorname{Tr} M$, $q = \det M$, where $\operatorname{Tr} M$ is a trace of matrix M [1, 12, 16]. Indeed, it follows from (2) that $kx_i^{(n-1)} - qx_i^{(n-2)} = (m_1 + m_4)x_i^{(n-1)} - (m_1m_4 - m_2m_3)x_i^{(n-2)} = (x_i^{(n)} - m_2y_i^{(n-1)}) + m_4x_i^{(n-1)} - m_1m_4x_i^{(n-2)} + m_2m_3x_i^{(n-2)} = x_i^{(n)} - m_2(y_i^{(n-1)} - m_3x_i^{(n-2)}) + m_4(x_i^{(n-1)} - m_1x_i^{(n-2)}) = x_i^{(n)} - m_2m_4y_i^{(n-2)} + m_2m_4y_i^{(n-2)} = x_i^{(n)} \pmod{g}$. The basic recurrence (1) is therefore closely related to so-called matrix generator of pseudorandom numbers studied in [12, 14, 25].

The output function is defined as follows:

$$a^{(n)} = \sum_{i=0}^{s-1} \lfloor 2x_i^{(n)}/g \rfloor \cdot 2^i, \tag{3}$$

where $i = 0, 1, \ldots, s-1$, i.e. each bit of the output corresponds to its own recurrence, and $s = 32$ recurrences are calculated in parallel.

For $g = p \cdot 2^t$, where p is an odd prime, the characteristic polynomial $f(x) = x^2 - kx + q$ is chosen to be primitive over $\mathbb{Z}_p$. Primitivity of the characteristic polynomial guarantees maximal possible period $p^2 - 1$ of the output sequence for $g = p$. It is straightforward to prove that taking $g = p \cdot 2^t$ instead of $g = p$ does not reduce the value of the period.

There is an easy algorithm to calculate $x^{(n)}$ in (1) very quickly from $x^{(0)}$ and $x^{(1)}$ for any large n. Indeed, if $x^{(2n)} = k_nx^{(n)} - q_nx^{(0)} \pmod{g}$, then $x^{(4n)} = (k_n^2 - 2q_n)x^{(2n)} - q_n^2x^{(0)} \pmod{g}$. As was mentioned already in [1], this helps to initialize the generator. To initialize all s recurrences, the following initial conditions are used: $x_i^{(0)} = x^{(iA)}$, $x_i^{(1)} = x^{(iA+1)}, i = 0, 1, \ldots, s-1$. Here A is a value of the order of $(p^2-1)/s$. The author has tested realizations with various values of A of the order of $(p^2-1)/s$ and found in all cases that the specific choice of A was not of importance for the properties studied in the next sections. Short cycles and, in particular, the cycle consisting of zeroes, are avoided if at least one of $x^{(0)}$ and $x^{(1)}$ is not divisible by p. As a result of the initialization, all s initial points belong to the same orbit on the torus of the period p^2-1, while the minimal distance A between the initial points along the orbit is chosen to be very large.

In addition to the realizations based on the output function (3) that takes a single bit from each linear recurrence, I have also constructed realizations based on a more general output function

$$a^{(n)} = \sum_{i=0}^{s-1} \lfloor 2^v x_i^{(n)}/g \rfloor \cdot 2^{iv}, \qquad (4)$$

where v bits are taken from each recurrence at each step and $i = 0, 1, \ldots, s-1$. For example, GM55.4-SSE realization calculates only $s = 8$ recurrence relations in parallel and takes $v = 4$ bits from each number. Pseudorandom 32-bit numbers can be generated if $sv \geq 32$. The sequence of bits $\{\lfloor 2^v x_i^{(n)}/g \rfloor\}$, where i is fixed and $\{x_i^{(n)}\}$ is generated with relation (2) will be designated below as a stream of v-bit blocks generated with matrix M. The pairs $x_i^{(0)}, x_i^{(1)} \in \mathbb{Z}_g$ for the recurrence (1) and $x_i^{(0)}, y_i^{(0)} \in \mathbb{Z}_g$ for the recurrence (2) represent seeds for the streams of v-bit blocks generated with (1) and (2) respectively. Consider the set of admissible seeds containing all seeds such that at least one of the two values is not divisible by p. Selecting the seed at random from a uniform distribution over the set of admissible seeds determines the probability measure for output subsequences of a stream of v-bit blocks. Such probabilities are considered below in Sects. 4 and 5.

The parameters for the particular constructed realizations of the generator are shown in Table 1. The parameters are chosen in order for the characteristic polynomial $x^2 - kx + q$ to be primitive over $\mathbb{Z}_p$. In addition, as is shown in Sect. 5, q must be divisible by 2^v in order for the high-dimensional equidistribution property to hold. Also the value of $(k+q)g$ should not exceed either 2^{32} or 2^{64} in order to effectively calculate four 32-bit recurrences or two 64-bit recurrences in parallel within SIMD arithmetic. In the particular case $t = 0$ and $v = 1$ the method reduces to that studied earlier in [1, 2]. Program codes for the new generators and proper initializations are available in [6].

3 Statistical Testing and Speed of the Generator

Table 1 shows the results of applying the SmallCrush, PseudoDiehard, Crush and BigCrush batteries of tests taken from [21], to the generators. For each battery of tests, Table 1 displays three characteristics: the number of statistical tests with p-values outside the interval $[10^{-3}, 1-10^{-3}]$, number of tests with p-values outside the interval $[10^{-5}, 1 - 10^{-5}]$, and number of tests with p-values outside the interval $[10^{-10}, 1 - 10^{-10}]$. Table 1 also contains the results of statistical tests for Mersenne Twister generator of Matsumoto and Nishimira [24], combined Tausworthe generator of L'Ecuyer [19] and combined multiple recursive generator proposed in [20]. These generators are modern examples of fast RNG implementations with good statistical properties (see Sect. 4.5.4 and 4.6.1 in [18]). Both LFSR113 and MT19937 fail the test `scomp_LinearComp` that is a linear complexity test for the binary sequences (see [21]), because the bits of LFSR113 and MT19937 have a linear structure by construction. Also LFSR113 fails the test `smarsa_MatrixRank` (see [21]). The period lengths for the generators MRG32K3A, LFSR113 and MT19937 are $3.1 \cdot 10^{57}$, $1.0 \cdot 10^{34}$ and $4.3 \cdot 10^{6001}$ respectively.

Libraries SmallCrush, PseudoDiehard, Crush and BigCrush contain 15, 126, 144 and 160 tests respectively.

The usefulness of a RNG for a specific application in physics depends on, possibly dangerous interferences of the correlations in the specific problem and those of the RNG. Modern statistical test suites contain tests that reveal known types of correlations for the RNGs, in particular, the types that are known to result in systematic errors in Monte-Carlo simulations and that were studied in [8,11,28–31]. One concludes that the new realizations described in this paper possess excellent statistical properties.

I have tested the CPU times needed for generating 10^9 random numbers. The results are shown in Table 2 for Intel Core i7-940 and AMD Turion X2 RM-70 processors respectively. The results are presented for different compilers and optimization options. The compilers in use are GNU C compiler gcc version 4.3.3 and Intel C compiler icc version 11.0. The CPU times for the realizations GM29.1-SSE, GM55.4-SSE, GQ58.1-SSE, GQ58.3-SSE and GQ58.4-SSE introduced in Table 1 are compared with those for Mersenne Twister generator of Matsumoto and Nishimira [24], combined Tausworthe generator of L'Ecuyer [19] and combined multiple recursive generator proposed in [20].

4 Geometric and Statistical Properties for Matrices with Unitary Determinant

Let $X = \{(x,y)^T \in \mathbb{R}^2 | 0 \le (x/g) < 1/2, 0 \le (y/g) < 1\}$, $Y = \{(x,y)^T \in \mathbb{R}^2 | 1/2 \le (x/g) < 1, 0 \le (y/g) < 1\}$, i.e. X is the left half of the torus and Y is the right half of the torus. Let the initial point be $(x_0^{(0)}, y_0^{(0)})^T$. For the first bits of the first five outputs of the generator to be 10011, it is necessary and sufficient to have $(x_0^{(0)}, y_0^{(0)})^T \in Z_{10011} = Y \cap R^{-1}(X) \cap R^{-2}(X) \cap R^{-3}(Y) \cap R^{-4}(Y)$. Here, R is the action of the cat map. Therefore, the set Z_{10011} is a subset of $[0,g)^2$ and consists of filled polygons. Below $S(D)$ will designate the area of any set $D \subset [0,g)^2$ after division by g^2, i.e., the area of a set D is equal to $g^2 S(D)$. It will be demonstrated

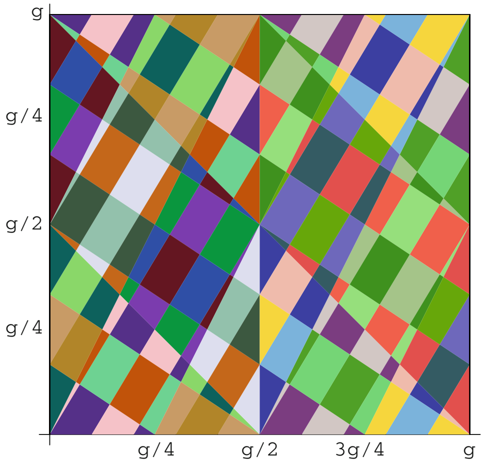


Fig. 1 The regions on the torus obtained in [1] for the third points of sequences of length 5 for the matrix $\left(\begin{smallmatrix}1 & 1\\ 1 & 2\end{smallmatrix}\right)$. These regions correspond to the sequences of length 5 of the first bits generated by the corresponding RNG. Each region is drawn with its own color.

that the nature of the correlations is connected with the geometric properties of the transformation.

Figure 1 represents the polygons corresponding to the subsequences of length five for the cat map with $M = \left(\begin{smallmatrix}1 & 1\\ 1 & 2\end{smallmatrix}\right)$. Each set of polygons represents the regions on the torus for the third point of the generator, e.g., $\tilde{Z}_{01001} = R^{-2}(X) \cap R^{-1}(Y) \cap X \cap R(X) \cap R^2(Y)$, and is drawn with its own color. Of course, $S(\tilde{Z}_{01001}) = S(Z_{01001})$ because the cat maps are area preserving. The geometric structures in Fig. 1 show the regions $\tilde{Z}_{00000}, \dots, \tilde{Z}_{11111}$ and illustrate the geometric approach to calculating the probabilities. The exact areas $S(Z_{00000}), \dots, S(Z_{11111})$ can be calculated.

The image of the lattice $g \times g$ of the torus with the transformation $M = \left(\begin{smallmatrix}m_1 & m_2\\ m_3 & m_4\end{smallmatrix}\right)$, where $\det M = 1$, is the same lattice $g \times g$ on the torus, because in this case the inverse matrix $M^{-1} = \left(\begin{smallmatrix}m_4 & g-m_2\\ g-m_3 & m_1\end{smallmatrix}\right) \pmod g$ is also a matrix with integer elements. If g is even and $q = 1$, then the numbers of points of the lattice inside X and Y are equal.

Proposition 1. *If* $M = \left(\begin{smallmatrix}m_1 & m_2\\ m_3 & m_4\end{smallmatrix}\right)$ *is a matrix with integer elements* m_1, m_2, m_3, m_4, $q = \det M = 1$, *then* $S(Z_{i_1 i_2}) = 1/4$, *for all* $i_1, i_2 \in \{0, 1\}$.

Proof. $S(Z_{00}) = S(Z_{10})$ because the corresponding areas pass into each other with the shift $(x/g, y/g)^T \to (x/g, \{y/g + 1/(2m_2)\})^T$, where $\{y/g + 1/(2m_2)\}$ is a fractional part of $y/g + 1/(2m_2)$. On the other hand, $S(Z_{00}) = S(Z_{11})$ because the corresponding areas pass into each other with the 180-degree turn with respect to the point $(x_R/g, y_R/g)^T = (1/2, 1/2)^T$. Proposition 1 is proved.

Figure 2 illustrates the structure of the set $R^{-1}(X)$ for odd m_1 (left panel) and for even m_1 (right panel). If m_1 is even then the areas Z_{00} and Z_{10} also pass into each other with the 180-degree turn with respect to the point $(x'_R/g, y'_R/g)^T = (1/4, 1/2)^T$.

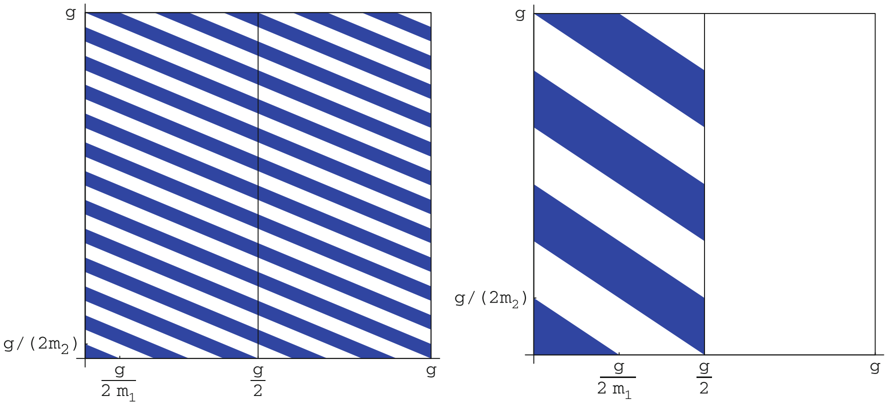


Fig. 2 *Left*: The set $R^{-1}(X)$ for matrix $M = \left(\begin{smallmatrix}5 & 12\\ 2 & 5\end{smallmatrix}\right)$. This set is similar for an arbitrary cat map with positive entries. The torus is divided into two halves for convenience. *Right*: The set $R^{-1}(X) \cap X$ for matrix $M = \left(\begin{smallmatrix}2 & 3\\ 3 & 5\end{smallmatrix}\right)$.

Proposition 2. *If M is a matrix with integer elements, $q = \det M = 1$, then $S(Z_{i_1 i_2 i_3}) = 1/8$, for all $i_1, i_2, i_3 \in \{0, 1\}$.*

Proof. Let $S(Z_{000}) = \alpha$ and $S(Z_{001}) = \beta$. Then $\alpha + \beta = S(Z_{00}) = 1/4$. Consequently, $S(Z_{011}) = 1/4 - S(Z_{001}) = \alpha$ and $S(Z_{111}) = 1/4 - S(Z_{011}) = \beta$. On the other hand, $S(Z_{111}) = S(Z_{000}) = \alpha$ because $R^{-2}(Y) \cap R^{-1}(Y) \cap Y$ passes into $R^{-2}(X) \cap R^{-1}(X) \cap X$ with the 180-degree rotation with respect to the point $(x_R/g, y_R/g)^T = (1/2, 1/2)^T$. Therefore, $\alpha = \beta = 1/8$. Proposition 2 is proved.

Proposition 3. *If M is a matrix with integer elements, $q = \det M = 1$, $k = \operatorname{Tr} M$ is an odd integer, then $S(Z_{i_1 i_2 i_3 i_4}) = 1/16$, for all $i_1, i_2, i_3, i_4 \in \{0, 1\}$.*

Proof. Let $M = \left(\begin{smallmatrix}a_1 & b_1\\ c_1 & d_1\end{smallmatrix}\right)$, $M^2 = \left(\begin{smallmatrix}a_2 & b_2\\ c_2 & d_2\end{smallmatrix}\right)$, and $M^3 = \left(\begin{smallmatrix}a_3 & b_3\\ c_3 & d_3\end{smallmatrix}\right)$. Here $k = \operatorname{Tr} M = a_1 + d_1$.

Let $S(Z_{0000}) = \alpha$ and $S(Z_{0001}) = \beta$. Then $\alpha + \beta = S(Z_{000}) = 1/8$. Because $q = 1$, we have $a_2 = ka_1 - 1$ and $a_3 = ka_2 - a_1 = a_1(k^2 - 1) - k$. Hence, there are two possibilities.

1. If a_1 is even, then a_2 and a_3 are odd. Taking the 180-degree rotations with respect to the points $(1/2, 1/2)^T$, $(1/4, 1/2)^T$, and Proposition 2 into account, we have $S(Z_{0100}) = S(Z_{0000}) = S(Z_{1111}) = S(Z_{1011}) = \alpha$, $S(Z_{0101}) = S(Z_{0001}) = S(Z_{1110}) = S(Z_{1010}) = \beta$, $S(Z_{1010}) = S(Z_{1000}) = 1/8 - S(Z_{0000}) = \beta$, $S(Z_{0110}) = S(Z_{0010}) = 1/8 - S(Z_{1010}) = \alpha$. Therefore, $1/4 = S(Z_{00}) = S(Z_{0000}) + S(Z_{0010}) + S(Z_{0100}) + S(Z_{0110}) = 4\alpha$, i.e., $\alpha = \beta = 1/16$.
2. If a_1 is odd, then a_2 is even and a_3 is odd. Taking the 180-degree rotations with respect to the points $(1/2, 1/2)^T$, $(1/4, 1/2)^T$ and Proposition 2 into account, we have $S(Z_{0010}) = S(Z_{0000}) = S(Z_{1111}) = S(Z_{1101}) = \alpha$, $S(Z_{0011}) = S(Z_{0001}) = S(Z_{1110}) = S(Z_{1100}) = \beta$, $S(Z_{0100}) = 1/8 - S(Z_{1100}) = \alpha$,

$S(Z_{0110}) = 1/8 - S(Z_{1110}) = \alpha$. Therefore, $1/4 = S(Z_{00}) = S(Z_{0000}) + S(Z_{0010}) + S(Z_{0100}) + S(Z_{0110}) = 4\alpha$, i.e., $\alpha = \beta = 1/16$.

Proposition 3 is proved.

The notion of probability of subsequences of a stream of v-bit blocks, that will be used in the Propositions below, is introduced in the end of Sect. 2.

Proposition 4. *If* $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ *is a matrix with integer elements,* $q = \det M = 1$*,* m_1 *and* g *are divisible by* 2^v*, then every sequence of length two in a stream of* v*-bit blocks generated with matrix* M*, has the same probability* $1/2^{2v}$.

Proof. Let $X_i = \{(x,y)^T \in \mathbb{R}^2 | i/2^v \le x/g < (i+1)/2^v, 0 \le (y/g) < 1\}$, i.e. the torus is divided into 2^v vertical stripes $X_0, X_1, \ldots, X_{2^v-1}$ of equal area. Consider the shift $S : (x,y)^T \to (x + g/2^v, y)^T \pmod g$, i.e. $S(X_i) = X_{(i+1) \pmod{2^v}}$. Then $MS(x,y)^T = M(x+g/2^v, y)^T = M(x,y) + (0, m_3 g/2^v) \pmod g$. Therefore, the set of points A of the $g \times g$-lattice, such that $A \in X_i$ and $M(A) \in X_j$ passes with the shift S into the set of points A of the same lattice such that $A \in X_{(i+1) \pmod{2^v}}$ and $M(A) \in X_j$. Let $P(i,j)$ be the probability of the sequence (i,j) of length two, where $i, j \in \{0, 1, \ldots, 2^v - 1\}$. Then $P(0,j) = P(1,j) = \cdots = P(2^v-1, j)$ and $\sum_{i=0}^{2^v-1} P(i,j) = 1/2^v$, because each point of the lattice has a single preimage. Proposition 4 is proved.

Proposition 5. *If* $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ *is a matrix with integer elements,* $m_2 = 2^u \cdot w$*, where* w *is odd,* g *is divisible by* 2^{u+v}*, then every sequence of length two in a stream of* v*-bit blocks generated with matrix* M*, has the same probability* $1/2^{2v}$.

Proof. Let $X_i = \{(x,y)^T \in \mathbb{R}^2 | i/2^v \le x/g < (i+1)/2^v, 0 \le (y/g) < 1\}$, i.e. the torus is divided into 2^v vertical stripes $X_0, X_1, \ldots, X_{2^v-1}$ of equal area. Consider the shift $S : (x,y)^T \to (x, y + g/2^{u+v} \pmod g)^T$, in this case $S(X_i) = X_i$. Then $MS(x,y)^T = M(x, y + g/2^{u+v})^T = M(x,y) + (gw/2^v, m_4 g/2^{u+v}) \pmod g$. Therefore, the set of points A of the $g \times g$-lattice, such that $A \in X_i$ and $M(A) \in X_j$ passes with the shift S into the set of points A of the same lattice such that $A \in X_i$ and $M(A) \in X_{(j+w) \pmod{2^v}}$. Therefore $P(i,0) = P(i,1) = \cdots = P(i, 2^v - 1)$ and $\sum_{j=0}^{2^v-1} P(i,j) = 1/2^v$. Proposition 5 is proved.

Proposition 6. *If* $M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix}$ *is a matrix with integer elements,* $k = 2^m \cdot r$*,* $m_2 = 2^u \cdot w$*, where* r, w *are odd,* $q = 1$ *and* g *is divisible by* 2^{u+m+v}*, then (i) every sequence of length three in a stream of bits generated with matrix* M *has the same probability* $1/8$*; (ii) if* $m = 0$*, then every sequence of length four in a stream of bits generated with matrix* M *has the same probability* $1/16$.

Proof. (i) It follows from Proposition 5 that every subsequence of length two is equiprobable because g is divisible by 2^{u+m+v} and $m_2^{(2)} = km_2$ where $M^2 = \begin{pmatrix} m_1^{(2)} & m_2^{(2)} \\ m_3^{(2)} & m_4^{(2)} \end{pmatrix} \pmod g$. The rest of proof is essentially the same as the proof of Proposition 2, where the numbers of points of the $g \times g$-lattice inside each region are considered at each step instead of considering the areas of the regions.

(ii) It follows from above that every subsequence of length three is equiprobable in this case. The rest of proof is essentially the same as the proof of Proposition 3, where the numbers of points of the $g \times g$-lattice inside each region are considered at each step instead of considering the areas of the regions. Proposition 6 is proved.

The following statements are also valid for $q = 1$ (see [1]): (i) if k is odd, then $S(Z_{00000})$ depends only on the trace k of matrix M of the cat map and equals $S = S_0(1+1/(3k^2-6))$, where $S_0 = 1/32$; (ii) if k is even, then $S(Z_{0000})$ depends only on the trace k of matrix M of the cat map and equals $S = S_0 \cdot k^2/(k^2-1)$, where $S_0 = 1/16$. The condition $S > S_0$ signifies that the 5-dimensional equidistribution never takes place for $q = 1$, i.e. for conservative hyperbolic automorphisms of the torus.

5 Geometric and Statistical Properties for $q \neq 1$

In [1] the connection between statistical properties, results of the random walk test, and geometric properties of the cat maps is established. Cat maps are simple chaotic dynamical systems that correspond to transformations (2) for $q = \det M = 1$, i.e. hyperbolic automorphisms of the two-dimensional torus. In particular, it is discussed in [1] and in the end of the previous section that the 5-dimensional equidistribution never takes place for $q = 1$, i.e. for conservative hyperbolic automorphisms of the torus. In this section another case $q \neq 1$ involving dissipative dynamical systems is studied.

Let $X_i = \{(x,y)^T \in \mathbb{R}^2 | i/2^v \leq x/g < (i+1)/2^v, 0 \leq (y/g) < 1\}$, i.e. the torus is divided into 2^v vertical stripes $X_0, X_1, \ldots, X_{2^v-1}$ of equal area. Suppose that g is divisible by 2^v and consider the shift $S : (x,y)^T \to (x+g/2^v, y) \pmod g$, i.e. $S(X_i) = X_{(i+1) \pmod{2^v}}$. The shift S is a superposition of two rotations: $S = R_2 R_1$, where R_1 is a 180-degree rotation with respect to the point $(1/2^{v+1}, 1/2)^T$ and R_2 a 180-degree rotation with respect to the point $(1/2^v, 1/2)^T$.

Proposition 7. *If (i) $M = \left(\begin{smallmatrix} m_1 & m_2 \\ m_3 & m_4 \end{smallmatrix}\right)$ is a matrix with integer values m_1, m_2, m_3, m_4, (ii) m_1, $q = \det M$ and g are divisible by 2^v, (iii) the image of the lattice $g \times g$ with the transformation M^j is invariant with respect to the shift S for $j = 0, 1, \ldots, n$, then all the sequences of length n in a stream of v-bit blocks generated with matrix M are equiprobable.*

Proof. In this case the element $m_1^{(n)}$ of matrix

$$M^n = \begin{pmatrix} m_1^{(n)} & m_2^{(n)} \\ m_3^{(n)} & m_4^{(n)} \end{pmatrix} \pmod g \tag{5}$$

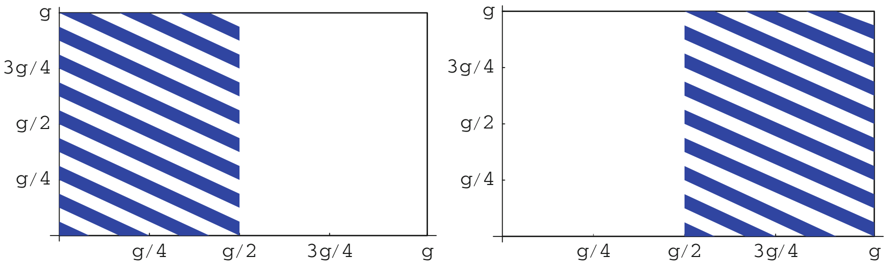


Fig. 3 The set of points A such that $A \in X_0$ and $M^2(A) \in X_0$ (*left panel*) and the set of points A such that $A \in X_1$ and $M^2(A) \in X_0$ (*right panel*) for $M = \binom{2\ 2}{1\ 2}$ and $v = 1$.

satisfies the recurrence relation $m_1^{(n)} = km_1^{(n-1)} - qm_1^{(n-2)} \pmod g$. Hence $m_1^{(n)}$ is divisible by 2^v for any integer $n \geq 1$.

Since $m_1^{(n)}$ is divisible by 2^v, one has $M^n S(x,y)^T = M^n(x + g/2^v \pmod g, y)^T = M^n(x,y)^T + (0, m_3^{(n)} g/2^v)^T$. Hence, the set of points A such that $A \in X_i$ and $M^n(A) \in X_j$ passes with the shift S into the set of points A such that $A \in X_{(i+1) \pmod{2^v}}$ and $M^n(A) \in X_j$.

Let's now prove by induction that all sequences of length n are equiprobable. Obviously, if g is divisible by 2^v, sequences of length 1 are equiprobable: $P(0) = P(1) = \cdots = P(2^v - 1) = 1/2^v$. Assume that all sequences of length $n-1$ are equiprobable. Let $\alpha_i = P(ix_1 \dots x_{n-1})$, $i = 0, 1, \dots, 2^v - 1$ be probabilities of sequences of length n. Then $\alpha_i = \alpha_{i+1}$, $i = 0, 1, \dots, 2^v - 2$ because the set of points A of the lattice $g \times g$ such that $A \in X_i$, $M(A) \in X_{x_1}, \dots, M^{n-1}(A) \in X_{x_{n-1}}$ passes with the shift S into the set of points A of the lattice $g \times g$ such that $A \in X_{(i+1) \pmod{2^v}}$, $M(A) \in X_{x_1}, \dots, M^{n-1}(A) \in X_{x_{n-1}}$. On the other hand, $\sum_{i=0}^{2^v-1} \alpha_i$ is the probability of sequence $x_1 \dots x_{n-1}$ of length $n-1$ and equals $1/2^{v(n-1)}$. Therefore, $\alpha_i = 1/2^{vn}$, $i = 0, 1, \dots, 2^v - 1$, and all sequences of length n are equiprobable. Proposition 7 is proved.

The condition that the image of the lattice $g \times g$ with the transformation M^j is invariant with respect to the shift S for $j = 0, 1, \dots, n$, is used in the above consideration and is necessary for the Proposition 7. For $j = 0$ the invariance means that g is divisible by 2^v. If g and $m_1^{(n)}$ are divisible by 2^v, then the number of points A of the lattice $g \times g$ such that $A \in X_0$ and $M^n(A) \in X_0$ is equal to the number of points A of the same lattice such that $A \in X_1$ and $M^n(A) \in X_0$. If g is not divisible by 2^v then these numbers are approximately equal because the corresponding areas are equal and g is a large number, and the exact equality holds only if g is divisible by 2^v. Figure 3 shows the sets of points $\{A | A \in X_0, M^2(A) \in X_0\}$ and $\{A | A \in X_1, M^2(A) \in X_0\}$ for $M = \binom{2\ 2}{1\ 2}$ and $v = 1$.

Example 1. For $M = \binom{2\ 2}{1\ 2}$, $M = \binom{10\ \ 17}{-4\ -2}$ and $M = \binom{244\ 43}{32\ 12}$ the sequences of length $1, 2, \dots, \ell$ in a stream of bits generated with matrix M are equiprobable, where $\ell = 2t - 1$, $\ell = (t-1)/2$ and $\ell = (t-1)/2$ respectively. Here $g = p \cdot 2^t$, where p is an odd prime, and the matrices correspond to the realizations GM29-SSE, GM58-SSE and GM55-SSE respectively.

Let's now prove this statement, i.e., let's check that the image of the lattice $g \times g$ with the transformation M^j is invariant with respect to the shift for $j = 0, 1, \ldots, n$ and $n \leq \ell$. In particular, the invariance takes place if there are integers $r, l < t$ such that the distance between integer vectors $(x + g/2^{r+1}, y + g/2^{l+1})^T$ and $(x, y)^T$ after applying transformation M^j is equal to $(g/2, 0)^T$ modulo g. This results in $(m_1^{(j)}/2^r + m_2^{(j)}/2^l, m_3^{(j)}/2^r + m_4^{(j)}/2^l)^T \equiv (1, 0)^T \pmod 2$. For the matrix $M = \left(\begin{smallmatrix} 2 & 2 \\ 1 & 2 \end{smallmatrix}\right)$ the condition is satisfied when $r = j/2$, $l = j/2 - 1$ for even j and $r = (j-1)/2$, $l = (j+1)/2$ for odd j. Thus $\ell = j_{max} + 1 = 2t - 1$. Similarly, for each of the matrices $M = \left(\begin{smallmatrix} 10 & 17 \\ -4 & -2 \end{smallmatrix}\right)$ and $M = \left(\begin{smallmatrix} 244 & 43 \\ 32 & 12 \end{smallmatrix}\right)$ the condition is satisfied for $\ell = (t-1)/2$.

Proposition 8. *Consider a matrix M with integer elements and the following integer quantities: $g = p \cdot 2^t$, $q = \det M = 2^u w \pmod g$, $k = \mathrm{Tr}\, M = 2^m r \pmod g$, $u \geq 1$, $t \geq v$, $m \geq 0$. Here w, r are odd integers and p is an odd prime. Then (i) all 2^{vj} sequences of length j in a stream of v-bit blocks generated with recurrence relation (1) are equiprobable for $j = 1, 2, \ldots, \ell$. Here $\ell = \lceil (t-v)/\lceil u/2 \rceil \rceil$ for $u \leq 2m$ and $\ell = \lceil (t-v)/(u-m) \rceil$ for $u > 2m$; (ii) if k is even, then the image of the lattice $g \times g$ with the transformation M^{2t} is the lattice $p \times p$ on the torus; (iii) if k is odd, then the image of the lattice $g \times g$ with the transformation $M^{\lceil t/u \rceil}$ is not invariant with respect to the shift S.*

Proof. (i) Let $X_i' = \{x | ig/2^v \leq x < (i+1)g/2^v\}$, $i = 0, 1, \ldots, 2^v - 1$. Let $k_0 = 1$, $k_1 = k \pmod g$, $k_{i+1} = kk_i - qk_{i-1} \pmod g$, $i = 1, 2, \ldots$. Consider the expressions

$$\xi^{(h-i)} = p \cdot 2^{t-iu-v} w^{h-i} k_i \pmod g. \tag{6}$$

Let the expressions define integer values of $\xi^{(h-i)}$ for $i = 0, 1, 2, \ldots, i_{max}$ for some i_{max}. Then it is straightforward to ascertain that the following relations are satisfied: $\xi^{(j)} = k\xi^{(j-1)} - q\xi^{(j-2)} \pmod g$, $j = h, h-1, \ldots, h - i_{max} + 2$. Also, it is easy to check that $\xi^{(h+i)} = 0$ for $i = 1, 2, \ldots$, where $\xi^{(h+i)}$ is defined as $k\xi^{(h+i-1)} - q\xi^{(h+i-2)}$.

It is easy to show by induction the following: if $u \leq 2m$ then k_i is divisible by $2^{\min(fi,t)}$, where $f = \lfloor u/2 \rfloor$; if $u > 2m$ then k_i is divisible by $2^{\min(mi,t)}$. Therefore, expressions (6) define integer values of $\xi^{(h-i)}$ for $i = 0, 1, 2, \ldots, \ell - 1$, where $\ell = \lceil (t-v)/\lceil u/2 \rceil \rceil$ for $u \leq 2m$ and $\ell = \lceil (t-v)/(u-m) \rceil$ for $u > 2m$.

Let's now prove that every sequence of length $n \leq \ell$ has the same probability $1/2^{vn}$. Obviously, since g is divisible by 2^v, sequences of length 1 are equiprobable and $P(i) = 1/2^v$ for $i = 0, 1, \ldots, 2^v - 1$. Let $P(x_h \ldots x_{n-1})$ be a probability that last $n - h$ elements of a sequence of length n are $x_h, \ldots, x_{n-1}$, where $x_i \in \{0, 1, \ldots, 2^v - 1\}$, $h < n$, $i = h, \ldots, n-1$. Then $P(x_h \ldots x_{n-1}) = |B|/g^2$, where $B = \{(x^{(0)}, x^{(1)})^T | x^{(h)} \in X'_{x_h}, \ldots, x^{(n-1)} \in X'_{x_{n-1}}\}$, $x^{(i)}$ is defined as $kx^{(i-1)} - qx^{(i-2)} \pmod g$ for $i \geq 2$, and $|B|$ is the number of elements of the set B.

Therefore, if $h \leq \ell - 1$ then $P(x_h x_{h+1} \dots x_{n-1}) = P(x'_h x_{h+1} \dots x_{n-1})$ where $x'_h = x_h + w^h \pmod{g}$. Indeed, $\{(x^{(0)} + \xi^{(0)}, x^{(1)} + \xi^{(1)})^T | x^{(h)} \in X'_{x_h}, x^{(h+1)} \in X'_{x_{h+1}}, \dots, x^{(n-1)} \in X'_{x_{n-1}}\} = \{(x^{(0)}, x^{(1)})^T | x^{(h)} \in X'_{x'_h}, x^{(h+1)} \in X'_{x_{h+1}}, \dots, x^{(n-1)} \in X'_{x_{n-1}}\}$, where $\xi^{(0)}$ and $\xi^{(1)}$ are defined in (6) and are integer values for $h \leq \ell$. Because w^h is an odd integer, one obtains $\beta_0 = \beta_1 = \dots = \beta_{2^v - 1}$ where $\beta_i = P(ix_{h+1} \dots x_{n-1})$, $i = 0, 1, \dots, 2^v - 1$.

In particular, for $h = n - 1$ and $n \leq \ell$ one has $P(i) = 1/2^v$, $i = 0, 1, \dots, 2^v - 1$, and one obtains by induction that $P(i x_{h+1} \dots x_{n-1}) = 1/2^{v(n-h)}$, $i = 0, 1, \dots, 2^v - 1$ for $h \leq \ell - 1$. In particular, if $n \leq \ell$, $P(i x_1 \dots x_{n-1}) = 1/2^{vn}$, $i = 0, 1, \dots, 2^v - 1$, and, therefore, $P(x_0 x_1 \dots x_{n-1}) = 1/2^{vn}$.

(ii) In this case $x^{(n+2)} = kx^{(n+1)} - qx^{(n)} \pmod{g}$, $n = 0, 1, 2, \dots$. Therefore if $(x^{(0)}, y^{(0)})^T$ belongs to the $g \times g$ lattice, then $x^{(2)}$ and $y^{(2)}$ are even integers, $x^{(4)}$ and $y^{(4)}$ are divisible by 4, etc. $x^{(2t)}$ and $y^{(2t)}$ are divisible by 2^t and therefore $(x^{(2t)}, y^{(2t)})^T$ belongs to the lattice $p \times p$ on the torus.

(iii) Let L be the image of the $g \times g$-lattice with the transformation M^n, where $n = \lceil t/u \rceil$. Then $(0,0)^T \in L$ because $M^n(0,0)^T = (0,0)^T$. If L is invariant with respect to the shift, then $(g/2^v, 0)^T \in L$. Therefore there exists a point $(x, y)^T$ of the $g \times g$-lattice such that $M^n(x,y)^T = (g/2^v, 0)^T \pmod{g}$. Because $m_1^{(n)}$ is divisible by 2^v, one has $M^n(g/2^v, 0)^T = (0, m_3^{(n)} g/2^v)^T \pmod{g}$. Therefore, $0 = k'_n g/2^v - q^n x \pmod{g}$ where $k'_n = \operatorname{Tr} M^n$ is an odd integer for $n \geq 1$. This is impossible because $q^n = 0 \pmod{2^t}$, $k'_n g/2^v \neq 0 \pmod{2^t}$.

Proposition 8 is proved.

Although the exact equidistribution property does not hold when distance between some points of the sequence $\geq 2t$, numerical results demonstrate that the equidistribution holds approximately with high accuracy for the sequences of bits of length n, where $n < 6.8 \log p$. Also, one can take n points with arbitrary distances (not exceeding $p^2 - 1$) between them along the orbit (i.e. not necessarily successive points of the orbit), where $n < 6.8 \log p$, and still the approximate equidistribution will hold with a high accuracy. The output value $a^{(n)}$ in (3) consists of high-order bits of s successive points along the orbit of matrix M^A, where A is introduced in Sect. 2, therefore, according to the numerical results, the output value $a^{(n)}$ has a uniform distribution with a very high accuracy.

In most cases the image of the lattice $g \times g$ on the torus with M^j where $j \geq 2t$ is the $p \times p$-lattice, therefore it is most interesting to study the deviations from the equidistribution for the $p \times p$-lattice. I have calculated the exact areas on the torus which correspond to each of the sequences for $M = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}$. The calculations were carried out on a PC using Class Library for Numbers [10] for exact rational arithmetics. For each of the 2^n sequences of length $n = 1, 2, \dots$, the corresponding set of points on the unit two-dimensional torus consists of filled polygons. Exact rational coordinates of all the vertices of each filled polygon were found. Also, the

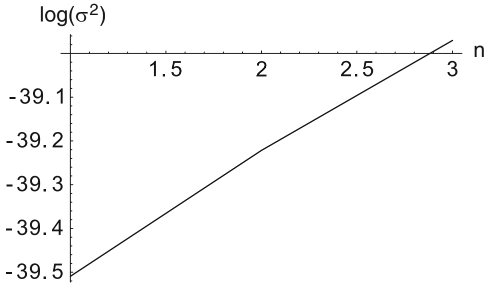


Fig. 4 Variance of the numbers of points of the $p \times p$-lattice corresponding to sequences of length n versus n. The values are normalized such that $\langle A_n \rangle = 1$.

exact number of points of the $p \times p$ lattice inside each polygon was calculated. The total area of the polygons for each of the 2^n sequences of length n was found to equal $1/(2^n)$. Such equality of the areas for different sequences of the same length was observed for matrices with even determinant and was not observed for matrices with odd determinant. Let $A_{n,0}, A_{n,1}, \dots, A_{n,2^n-1}$ be the numbers of points of the $p \times p$-lattice inside the sets of filled polygons which correspond to the sequences of length n. Then $\sum_{i=0}^{2^n-1} A_{n,i} = p^2$. Therefore, if A_n is the set of numbers $A_n = \{2^n A_{n,0}/p^2, 2^n A_{n,1}/p^2, \dots, 2^n A_{n,2^n-1}/p^2\}$, then $\langle A_n \rangle = 1$, where $\langle A_n \rangle$ is the average value of A_n. The dependence of logarithm of variance of A_n on n is shown in Fig. 4 for $p = 2^{29} - 3$. The calculations for smaller values of p and larger values of n demonstrate that the dependence of $\log(\sigma^2)$ on n is almost linear. Calculations show that the deviations from equidistribution are negligibly small in the sense that $\sigma(A_n)$ is much smaller than $\langle A_n \rangle = 1$, for $n < 6.8 \log p$. In particular, for $p = 2^{29} - 3$ the deviations are small for $n < 130$.

The variance for the several points of the orbit of matrix M on the $p \times p$-lattice on the torus, is found to substantially depend on the number of points and on the value of p, and only weakly depend (within several percent) on the distances between the points along the orbit.

This work was supported by Russian Foundation for Basic Research.

References

1. L. Barash, L.N. Shchur, Physical Review E **73** (2006), 036701.
2. L.Yu. Barash, L.N. Shchur, Computer Physics Communications **182** (2011) 1518–1527.
3. L.Yu. Barash, Europhysics Letters 95 (2011) 10003.
4. K.S.D. Beach, P.A. Lee, P. Monthoux, Physical Review Letters **92** (2004) 026401.
5. A.R. Bizzarri, Journal of Physics: Condensed Matter **16** (2004) R83–R110.
6. http://www.comphys.ru/barash/rng_sse2.zip
7. R. Couture, P. L'Ecuyer, S. Tezuka, Mathematics of Computation **60**, 749–761 (1993).
8. A.M. Ferrenberg, D.P. Landau, Y.J. Wong, Physical Review Letters **69** (1992) 3382–3384.
9. M. Fushimi, S. Tezuka, Communications of the ACM **26(7)**, 516–523 (1983).
10. http://www.ginac.de/CLN/
11. P. Grassberger Physics Letters A **181** (1993) 43–46.
12. H. Grothe, Statistische Hefte, **28** (1987) 233–238.

13. http://www.intel.com/support/processors/pentium4/sb/CS-029967.htm
14. D.E. Knuth, *The Art of Computer Programming*, Vol. 2 (Addison-Wesley, Reading, Mass., 3rd edition, 1997).
15. D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2000).
16. P. L'Ecuyer, Communications of the ACM, **33(10)** (1990) 85–98.
17. P. L'Ecuyer, Mathematics of Computation **65**, 203–213 (1996).
18. P. L'Ecuyer, Chapter 4 of the Handbook of Simulation, Jerry Banks Ed., Wiley, 1998, pp. 93–137.
19. P. L'Ecuyer, Mathematics of Computation, **68** (1999) 261–269.
20. P. L'Ecuyer, Operations Research, **47** (1999) 159–164.
21. P. L'Ecuyer, R. Simard, *TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators (2002)*, Software user's guide, http://www.iro.umontreal.ca/∼simardr/testu01/tu01.html
22. P. L'Ecuyer, R. Simard, *TestU01: A C Library for Empirical Testing of Random Number Generators*, ACM Transactions On Mathematical Software, **33(4)** (2007) article 22.
23. A. Lüchow, Annual Review of Physical Chemistry, **51** (2000) 501–526.
24. M. Matsumoto and T. Nishimura, ACM Transactions on Modeling and Computer Simulation, **8** (1998) 3–30.
25. H. Niederreiter, in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, ed. H. Niederreiter and P. J.-S. Shiue, Lecture Notes in Statistics, (Springer-Verlag, 1995).
26. F. Panneton, P. L'Ecuyer, M. Matsumoto, ACM Transactions on Mathematical Software, **32(1)** (2006) 1–16.
27. S.C. Pieper and R.B. Wiring, Annual Review of Nuclear and Particle Science, **51** (2001) 53–90.
28. F. Schmid, N.B. Wilding, International Journal of Modern Physics C **6** (1995) 781–787.
29. L.N. Shchur, Computer Physics Communications **121–122** (1999) 83–85.
30. L.N. Shchur, J.R. Heringa, H.W.J. Blöte, Physica A **241** (1997) 579–592.
31. L.N. Shchur, H.W.J. Blöte, Physical Review E **55** (1997) R4905–R4908.
32. http://support.amd.com/us/Embedded_TechDocs/24592.pdf
33. S. Tezuka, P. L'Ecuyer, ACM Transactions on Modeling and Computer Simulation **1**, 99–112 (1991).
34. J.P.R. Tootil, W.D. Robinson, D.J. Eagle, Journal of the ACM **20(3)**, 469–481 (1973).

Computing Greeks Using Multilevel Path Simulation

Sylvestre Burgos and Michael B. Giles

Abstract We investigate the extension of the multilevel Monte Carlo method (M.B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, In A. Keller, S. Heinrich, and H. Niederreiter, editors, Monte Carlo and Quasi-Monte Carlo Methods 2006, 343–358, Springer-Verlag, 2007; M.B. Giles, Oper Res 56(3):607–617, 2008) to the calculation of Greeks. The pathwise sensitivity analysis (P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer, New York, 2004) differentiates the path evolution and effectively reduces the smoothness of the payoff. This leads to new challenges: the use of naive algorithms is often impossible because of the inapplicability of pathwise sensitivities to discontinuous payoffs.

These challenges can be addressed in three different ways: payoff smoothing using conditional expectations of the payoff before maturity (P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer, New York, 2004); an approximation of the above technique using path splitting for the final timestep (S. Asmussen and P. Glynn, Stochastic Simulation, Springer, New York, 2007); the use of a hybrid combination of pathwise sensitivity and the Likelihood Ratio Method (M.B. Giles, Vibrato Monte Carlo sensitivities, In P. L'Ecuyer and A. Owen, editors, Monte Carlo and Quasi-Monte Carlo Methods 2008, 369–382, Springer, 2009). We discuss the strengths and weaknesses of these alternatives in different multilevel Monte Carlo settings.

S. Burgos · M.B. Giles
Oxford-Man Institute of Quantitative Finance, University of Oxford, Oxford, UK
e-mail: sylvestre.burgos@maths.ox.ac.uk; mike.giles@maths.ox.ac.uk

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_13, © Springer-Verlag Berlin Heidelberg 2012

1 Introduction

In mathematical finance, Monte Carlo methods are used to compute the price of an option by estimating the expected value $\mathbb{E}(P)$. P is the payoff function that depends on an underlying asset's scalar price $S(t)$ which satisfies an evolution SDE of the form

$$\mathrm{d}S(t) = a(S,t)\,\mathrm{d}t + b(S,t)\,\mathrm{d}W_t, \quad 0 \le t \le T, \quad S(0) \text{ given.} \tag{1}$$

This is just one use of Monte Carlo in finance. In practice the prices are often quoted and used to calibrate our market models; the option's sensitivities to market parameters, the so-called Greeks, reflect the exposure to different sources of risk. Computing these is essential to hedge portfolios and is therefore even more important than pricing the option itself. This is why our research focuses on getting fast and accurate estimates of Greeks through Monte Carlo simulations.

1.1 Multilevel Monte Carlo

Let us first recall important results from [4] and [5]. We consider a standard Monte Carlo method using a discretisation with first order weak convergence (e.g. the Milstein scheme). Achieving a root-mean square error of $O(\varepsilon)$ requires a variance of order $O(\varepsilon^2)$, hence $O(\varepsilon^{-2})$ independent paths. It also requires a discretisation bias of order $O(\varepsilon)$, thus $O(\varepsilon^{-1})$ timesteps, giving a total computational cost $O(\varepsilon^{-3})$.

Giles' multilevel Monte Carlo technique reduces this cost to $O(\varepsilon^{-2})$ under certain conditions. The idea is to write the expected payoff with a fine discretisation using 2^L uniform timesteps as a telescopic sum. Let $\widehat{P}_\ell$ be the simulated payoff with a discretisation using 2^ℓ uniform timesteps,

$$\mathbb{E}(\widehat{P}_L) = \mathbb{E}(\widehat{P}_0) + \sum_{\ell=1}^{L} \mathbb{E}(\widehat{P}_\ell - \widehat{P}_{\ell-1}) \tag{2}$$

We then use Monte Carlo estimators using N_ℓ independent samples

$$\mathbb{E}(\widehat{P}_\ell - \widehat{P}_{\ell-1}) \approx \widehat{Y}_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(\widehat{P}_\ell^{(i)} - \widehat{P}_{\ell-1}^{(i)} \right) \tag{3}$$

The small corrective term $\widehat{P}_\ell^{(i)} - \widehat{P}_{\ell-1}^{(i)}$ comes from the difference between a fine and a coarse discretisation of the same driving Brownian motion. Its magnitude depends on the strong convergence properties of the scheme used. Let V_ℓ be the variance of a single sample $\widehat{P}_\ell^{(i)} - \widehat{P}_{\ell-1}^{(i)}$. The next theorem shows that what determines the efficiency of the multilevel approach is the convergence rate of V_ℓ as $\ell \to \infty$.

To ensure a better efficiency we may modify (3) and use different estimators of $\widehat{P}$ on the fine and coarse levels of $\widehat{Y}_\ell$,

$$\mathbb{E}(\widehat{P}_L) = \mathbb{E}(\widehat{P}_0) + \sum_{\ell=1}^{L} \mathbb{E}\left(\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c\right) \tag{4}$$

$\widehat{P}_\ell^f$, $\widehat{P}_{\ell-1}^c$ are the estimators using respectively 2^ℓ and $2^{\ell-1}$ steps in the computation of $\widehat{Y}_\ell$. The telescoping sum property is maintained provided that

$$\mathbb{E}\left(\widehat{P}_\ell^f\right) = \mathbb{E}\left(\widehat{P}_\ell^c\right). \tag{5}$$

Theorem 1. *Let P be a function of a solution to (1) for a given Brownian path $W(t)$; let $\widehat{P}_\ell$ be the corresponding approximation using the discretisation at level ℓ, i.e. with 2^ℓ steps of width $h_\ell = 2^{-\ell}\,T$.*

If there exist independent estimators $\widehat{Y}_\ell$ of computational complexity C_ℓ based on N_ℓ samples and there are positive constants $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ such that

1. $\mathbb{E}(\widehat{Y}_\ell) = \begin{cases} \mathbb{E}(\widehat{P}_0) & \text{if } l = 0 \\ \mathbb{E}(\widehat{P}_\ell - \widehat{P}_{\ell-1}) & \text{if } \ell > 0 \end{cases}$
2. $|\mathbb{E}(\widehat{P}_\ell - P)| \leq c_1 h_\ell^\alpha$
3. $\mathbb{V}(\widehat{Y}_\ell) \leq c_2 h_\ell^\beta\, N_\ell^{-1}$
4. $C_\ell \leq c_3 N_\ell\, h_\ell^{-1}$

Then there is a constant c_4 such that for any $\varepsilon < e^{-1}$, there are values for L and N_ℓ resulting in a multilevel estimator $\widehat{Y} = \sum_{\ell=0}^{L} \widehat{Y}_\ell$ with a mean-square-error $MSE = \mathbb{E}((\widehat{Y} - \mathbb{E}(P))^2) < \varepsilon^2$ with a complexity C bounded by

$$C \leq \begin{cases} c_4\varepsilon^{-2} & \text{if } \beta > 1 \\ c_4\varepsilon^{-2}(\log\varepsilon)^2 & \text{if } \beta = 1 \\ c_4\varepsilon^{-2-(1-\beta)/\alpha} & \text{if } 0 < \beta < 1 \end{cases} \tag{6}$$

Proof. See [5].

We usually know α thanks to the literature on weak convergence. Results in [9] give $\alpha = 1$ for the Milstein scheme, even in the case of discontinuous payoffs. β is related to strong convergence and is in practice what determines the efficiency of the multilevel approach. Its value depends on the payoff and may not be known *a priori*.

1.2 Monte Carlo Greeks

Let us briefly recall two classic methods used to compute Greeks in a Monte Carlo setting: the pathwise sensitivities and the Likelihood Ratio Method. More details can be found in [2, 3] and [8].

1.2.1 Pathwise Sensitivities

Let $\widehat{S} = (\widehat{S}_k)_{k\in[0,N]}$ be the simulated values of the asset at the discretisation times and $\widehat{W} = (\widehat{W}_k)_{k\in[1,N]}$ be the corresponding set of independent Brownian increments. The value of the option V is estimated by $\widehat{V}$ defined as

$$V = \mathbb{E}\left[P(S)\right] \approx \widehat{V} = \mathbb{E}\left[P(\widehat{S})\right] = \int P(\widehat{S})p(\theta,\widehat{S})\,\mathrm{d}\widehat{S}$$

Assuming that the payoff $P(\widehat{S})$ is Lipschitz, we can use the chain rule and write

$$\frac{\partial\widehat{V}}{\partial\theta} = \frac{\partial}{\partial\theta}\int P\left(\widehat{S}(\theta,\widehat{W})\right)\,p(\widehat{W})\mathrm{d}\widehat{W} = \int \frac{\partial P(\widehat{S})}{\partial\widehat{S}}\frac{\partial\widehat{S}(\theta,\widehat{W})}{\partial\theta}\,p(\widehat{W})\mathrm{d}\widehat{W}$$

where $\mathrm{d}\widehat{W} = \prod_{k=1}^{N}\mathrm{d}\widehat{W}_k$ and $p(\widehat{W}) = \prod_{k=1}^{N}p(\widehat{W}_k)$ is the joint probability density function of the normally distributed independent increments $(\widehat{W}_k)_{k\in[1,N]}$.

We obtain $\dfrac{\partial\widehat{S}}{\partial\theta}$ by differentiating the discretisation of (1) with respect to θ and iterating the resulting formula.The limitation of this technique is that it requires the payoff to be Lipschitz and piecewise differentiable.

1.2.2 Likelihood Ratio Method

The Likelihood Ratio Method starts from

$$\widehat{V} = \mathbb{E}\left[P(\widehat{S})\right] = \int P(\widehat{S})p(\theta,\widehat{S})\,\mathrm{d}\widehat{S} \tag{7}$$

The dependence on θ comes through the probability density function $p(\theta,\widehat{S})$; assuming some conditions discussed in [3] and in Sect. 7 of [8], we can write

$$\begin{aligned}\frac{\partial\widehat{V}}{\partial\theta} &= \int P(\widehat{S})\frac{\partial p(\widehat{S})}{\partial\theta}\,\mathrm{d}\widehat{S} = \int P(\widehat{S})\frac{\partial\log p(\widehat{S})}{\partial\theta}p(\widehat{S})\,\mathrm{d}\widehat{S}\\ &= \mathbb{E}\left[P(\widehat{S})\frac{\partial\log p(\widehat{S})}{\partial\theta}\right]\end{aligned} \tag{8}$$

$$\text{with}\qquad \mathrm{d}\widehat{S} = \prod_{k=1}^{N}\mathrm{d}\widehat{S}_k \qquad\text{and}\qquad p(\widehat{S}) = \prod_{k=1}^{N}p\left(\widehat{S}_k\,\middle|\,\widehat{S}_{k-1}\right)$$

The main limitation of the method is that the estimator's variance is $O(N)$, increasing without limit as we refine the discretisation.

1.3 Multilevel Monte Carlo Greeks

By combining the elements of Sects. 1.1 and 1.2 together, we write

$$\frac{\partial V}{\partial \theta} = \frac{\partial \mathbb{E}(P)}{\partial \theta} \approx \frac{\partial \mathbb{E}(\widehat{P}_L)}{\partial \theta} = \frac{\partial \mathbb{E}(\widehat{P}_0)}{\partial \theta} + \sum_{\ell=1}^{L} \frac{\partial \mathbb{E}(\widehat{P}_\ell - \widehat{P}_{\ell-1})}{\partial \theta} \tag{9}$$

As in (3), we define the multilevel estimators

$$\widehat{Y}_0 = N_0^{-1} \sum_{i=1}^{M} \frac{\partial \widehat{P}_0^{(i)}}{\partial \theta} \qquad \text{and} \qquad \widehat{Y}_\ell = N_\ell^{-1} \sum_{i=1}^{N_\ell} \left(\frac{\partial \widehat{P}_\ell^{(i)}}{\partial \theta} - \frac{\partial \widehat{P}_{\ell-1}^{(i)}}{\partial \theta} \right) \tag{10}$$

where $\frac{\partial \widehat{P}_0}{\partial \theta}, \frac{\partial \widehat{P}_{\ell-1}}{\partial \theta}, \frac{\partial \widehat{P}_\ell}{\partial \theta}$ are computed with the techniques presented in Sect. 1.2.

2 European Call

We consider the Black-Scholes model: the asset's evolution is modelled by a geometric Brownian motion $\mathrm{d}S(t) = r\, S(t)\mathrm{d}t + \sigma\, S(t)\mathrm{d}W_t$. We use the Milstein scheme for its good strong convergence properties. For timesteps of width h,

$$\widehat{S}_{n+1} = \widehat{S}_n \cdot \left(1 + r\, h + \sigma\, \Delta W_n + \frac{\sigma^2}{2} \left(\Delta W_n^2 - h \right) \right) := \widehat{S}_n \cdot D_n \tag{11}$$

The payoff of the European call is $P = (S_T - K)^+ = \max(0, S_T - K)$. We illustrate the techniques by computing Delta and Vega, the sensitivities to the asset's initial value S_0 and to its volatility σ. We take a time to maturity $T = 1$.

2.1 Pathwise Sensitivities

Since the payoff is Lipschitz, we can use pathwise sensitivities. The differentiation of (11) gives

$$\frac{\partial \widehat{S}_0}{\partial S_0} = 1, \qquad \frac{\partial \widehat{S}_{n+1}}{\partial S_0} = \frac{\partial \widehat{S}_n}{\partial S_0} \cdot D_n$$

$$\frac{\partial \widehat{S}_0}{\partial \sigma} = 0, \qquad \frac{\partial \widehat{S}_{n+1}}{\partial \sigma} = \frac{\partial \widehat{S}_n}{\partial \sigma} \cdot D_n + \widehat{S}_n \left(\Delta W_n + \sigma (\Delta W_n^2 - h) \right)$$

To compute $\widehat{Y}_\ell$ we use a fine and a coarse discretisation with $N_f = 2^\ell$ and $N_c = 2^{\ell-1}$ uniform timesteps respectively.

$$\widehat{Y}_\ell = \frac{1}{N_\ell}\sum_{i=1}^{N_\ell}\left[\left(\frac{\partial P}{\partial S_{N_f}}\frac{\partial \widehat{S}^{(i)}_{N_f}}{\partial \theta}\right)^{(\ell)} - \left(\frac{\partial P}{\partial S_{N_c}}\frac{\partial \widehat{S}^{(i)}_{N_c}}{\partial \theta}\right)^{(\ell-1)}\right] \tag{12}$$

We use the same driving Brownian motion for the fine and coarse discretisations: we first generate the fine Brownian increments $\widehat{W} = (\Delta W_0, \Delta W_2, \dots, \Delta W_{N_f-1})$ and then use $\widehat{W}^c = (\Delta W_0 + \Delta W_1, \dots, \Delta W_{N_f-2} + \Delta W_{N_f-1})$ as the coarse level's increments.

To assess the order of convergence of $\mathbb{V}(\widehat{Y}_L)$, we take a sufficient number of samples so that the Monte Carlo error of our simulations will not influence the results. We plot $\log(\mathbb{V}(\widehat{Y}_\ell))$ as a function of $\log(h_\ell)$ and use a linear regression to measure the slope for the different estimators. The theoretical results on convergence are asymptotic ones, therefore the coarsest levels are not relevant: hence we perform the linear regression on levels $\ell \in [3, 8]$. This gives a numerical estimate of the parameter β in Theorem 1. Combining this with the theorem, we get an estimated complexity of the multilevel algorithm. This gives the following results :

Estimator	β	MLMC complexity
Value	≈ 2.0	$O(\varepsilon^{-2})$
Delta	≈ 0.8	$O(\varepsilon^{-2.2})$
Vega	≈ 1.0	$O(\varepsilon^{-2}\log\varepsilon^2)$

Giles has shown in [4] that $\beta = 2$ for the value's estimator. For Greeks, the convergence is degraded by the discontinuity of $\frac{\partial P}{\partial S} = \mathbf{1}_{S>K}$: a fraction $O(h_\ell)$ of the paths has a final value $\widehat{S}$ which is $O(h_\ell)$ from the discontinuity K. For these paths, there is a $O(1)$ probability that $\widehat{S}^{(\ell)}_{N_f}$ and $\widehat{S}^{(\ell-1)}_{N_c}$ are on different sides of the strike K, implying $\left(\frac{\partial P}{\partial S_{N_f}}\frac{\partial \widehat{S}_{N_f}}{\partial \theta}\right)^{(\ell)} - \left(\frac{\partial P}{\partial S_{N_c}}\frac{\partial \widehat{S}_{N_c}}{\partial \theta}\right)^{(\ell-1)}$ is $O(1)$. Thus $\mathbb{V}(\widehat{Y}_\ell) = O(h_\ell)$, and $\beta = 1$ for the Greeks.

2.2 Pathwise Sensitivities and Conditional Expectations

We have seen that the payoff's lack of smoothness prevents the variance of Greeks' estimators $\widehat{Y}_\ell$ from decaying quickly and limits the potential benefits of the multilevel approach. To improve the convergence speed, we can use conditional expectations as explained in Sect. 7.2 of [8]. Instead of simulating the whole path, we stop at the penultimate step and then for every fixed set $\widehat{W} = (\Delta W_k)_{k\in[0,N-2]}$, we consider the full distribution of $\left(\widehat{S}_N|\widehat{W}\right)$. With $a_n = a\left(\widehat{S}_{N-1}(\widehat{W}), (N-1)h\right)$ and $b_n = b\left(\widehat{S}_{N-1}(\widehat{W}), (N-1)h\right)$, we can write

$$\widehat{S}_N(\widehat{W}, \Delta W_{N-1}) = \widehat{S}_{N-1}(\widehat{W}) + a_n(\widehat{W})h + b_n(\widehat{W})\,\Delta W_{N-1} \tag{13}$$

We hence get a normal distribution for $\left(\widehat{S}_N|\widehat{W}\right)$.

$$p(\widehat{S}_N|\widehat{W}) = \frac{1}{\sigma_{\widehat{W}}\sqrt{2\pi}}\,\exp\left(-\frac{\left(\widehat{S}_N - \mu_{\widehat{W}}\right)^2}{2\sigma^2_{\widehat{W}}}\right) \tag{14}$$

with

$$\mu_{\widehat{W}} = \widehat{S}_{N-1} + a\left(\widehat{S}_{N-1}, (N-1)h\right)h$$

$$\sigma_{\widehat{W}} = b\left(\widehat{S}_{N-1}, (N-1)h\right)\sqrt{h}$$

If the payoff function is sufficiently simple, we can evaluate analytically $\mathbb{E}\left[P\left(\widehat{S}_N\right)|\widehat{W}\right]$. Using the tower property, we get

$$\widehat{V} = \mathbb{E}\left[P(\widehat{S}_N)\right] = \mathbb{E}_{\widehat{W}}\left[\mathbb{E}_{\Delta W_N}\left[P(\widehat{S}_N)\middle|\widehat{W}\right]\right] \approx \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[P\left(\widehat{S}_N^{(m)}\right)\middle|\widehat{W}^{(m)}\right] \tag{15}$$

In the particular case of geometric Brownian motion and a European call option, we get (16) where ϕ is the normal probability density function, Φ the normal cumulative distribution function, $\alpha = (1+rh)\widehat{S}_{N-1}(\widehat{W})$ and $\beta = \sigma\sqrt{h}\widehat{S}_{N-1}(\widehat{W})$.

$$\mathbb{E}(P(\widehat{S}_N)|\widehat{W}) = \beta\,\phi\left(\frac{\alpha-K}{\beta}\right) + (\alpha-K)\,\Phi\left(\frac{\alpha-K}{\beta}\right) \tag{16}$$

This expected payoff is infinitely differentiable with respect to the input parameters. We can apply the pathwise sensitivities technique to this smooth function at time $(N-1)\,h$. The multilevel estimator for the Greek is then

$$\widehat{Y}_\ell = \frac{1}{N_\ell}\sum_{1}^{N_\ell}\left[\left(\frac{\partial\widehat{P}_f^{(i)}}{\partial\theta}\right)^{(\ell)} - \left(\frac{\partial\widehat{P}_c^{(i)}}{\partial\theta}\right)^{(\ell-1)}\right] \tag{17}$$

At the fine level we use (16) with $h = h_f$ and $\widehat{W}_f = (\Delta W_0, \Delta W_2, \ldots, \Delta W_{N_f-2})$ to get $\mathbb{E}(P(\widehat{S}_{N_f})|\widehat{W}_f)$. We then use

$$\left(\frac{\partial\widehat{P}_f}{\partial\theta}\right)^{(\ell)} = \frac{\partial\widehat{S}_{N_f-1}}{\partial\theta}\frac{\partial\mathbb{E}(P(\widehat{S}_{N_f})|\widehat{W}_f)}{\partial S_{N_f-1}} + \frac{\partial\mathbb{E}(P(\widehat{S}_{N_f})|\widehat{W}_f)}{\partial\theta} \tag{18}$$

At the coarse level, directly using $\mathbb{E}(P(\widehat{S}_{N_c})|\widehat{W}_c)$ leads to an unsatisfactorily low convergence rate of $\mathbb{V}(\widehat{Y}_\ell)$. As explained in (4) we use a modified estimator. The idea is to include the final fine Brownian increment in the computation of the expectation over the last coarse timestep. This guarantees that the two paths will be close to one another and helps achieve better variance convergence rates.

$\widehat{S}$ still follows a simple Brownian motion with constant drift and volatility on all coarse steps. With $\widehat{W}_c = (\Delta W_0 + \Delta W_1, \ldots, \Delta W_{N_f-4} + \Delta W_{N_f-3})$ and given that the Brownian increment on the first half of the final step is ΔW_{N_f-2} , we get

$$p(\widehat{S}_{N_c}|\widehat{W}_c, \Delta W_{N_f-2}) = \frac{1}{\sigma_{\widehat{W}_c}\sqrt{2\pi}} \exp\left(-\frac{\left(\widehat{S}_{N_c} - \mu_{\widehat{W}_c}\right)^2}{2\sigma^2_{\widehat{W}_c}}\right) \tag{19}$$

with

$$\mu_{\widehat{W}_c} = \widehat{S}_{N_c-1}(\widehat{W}_c) + a\left(\widehat{S}_{N_c-1}, (N_c-1)h_c\right) h_c + b\left(\widehat{S}_{N_c-1}, (N_c-1)h_c\right) \Delta W_{N_f-2}$$

$$\sigma_{\widehat{W}_c} = b\left(\widehat{S}_{N_c-1}, (N_c-1)h_c\right) \sqrt{h_c/2}$$

From this distribution we derive $\mathbb{E}\left[P(\widehat{S}_{N_c}) \mid \widehat{W}_c, \Delta W_{N_f-2}\right]$, which for the particular application being considered, leads to the same payoff formula as before with $\alpha_c = (1 + r\,h_c + \sigma \Delta W_{N_f-2})\,\widehat{S}_{N_c-1}(\widehat{W}_c)$ and $\beta_c = \sigma\sqrt{h_c}\,\widehat{S}_{N_c-1}(\widehat{W}_c)$. Using it as the coarse level's payoff does not introduce any bias. Using the tower property we check that it satisfies condition (5),

$$\mathbb{E}_{\Delta W_{N_f-1}}\left[\mathbb{E}\left[P(\widehat{S}_{N_c})\,\middle|\,\widehat{W}_c, \Delta W_{N_f-2}\right]\,\middle|\,\widehat{W}_c\right] = \mathbb{E}\left[P(\widehat{S}_{N_c})\,\middle|\,\widehat{W}_c\right]$$

Our numerical experiments show the benefits of the conditional expectation technique on the European call:

Estimator	β	MLMC complexity
Value	≈ 2.0	$O(\varepsilon^{-2})$
Delta	≈ 1.5	$O(\varepsilon^{-2})$
Vega	≈ 2.0	$O(\varepsilon^{-2})$

A fraction $O(\sqrt{h_\ell})$ of the paths arrive in the area around the strike where the conditional expectation $\dfrac{\partial \mathbb{E}(P(\widehat{S}_N)|\widehat{W})}{\partial \widehat{S}_{N_f-1}}$ is neither close to 0 nor 1. In this area, its slope is $O(h_\ell^{-1/2})$. The coarse and fine paths differ by $O(h_\ell)$, we thus have $O(\sqrt{h_\ell})$ difference between the coarse and fine Greeks' estimates. Reasoning as in [4] we get $\mathbb{V}_{\widehat{W}}(\mathbb{E}_{\Delta W_{N-1}}(\ldots|\widehat{W})) = O(h_\ell^{3/2})$ for the Greeks' estimators. This is the convergence rate observed for Delta; the higher convergence rate of Vega is not explained yet by this rough analysis and will be investigated in our future research.

The main limitation of this approach is that in many situations it leads to complicated integral computations. Path splitting, to be discussed next, may represent a useful numerical approximation to this technique.

2.3 Split Pathwise Sensitivities

This technique is based on the previous one. The idea is to avoid the tricky computation of $\mathbb{E}\left[P(\widehat{S}_{N_f})|\widehat{W}_f\right]$ and $\mathbb{E}\left[P(\widehat{S}_{N_c})|\widehat{W}_c, \Delta W_{N_f-2}\right]$. Instead, as detailed in Sect. 5.5 of [1], we get numerical estimates of these values by "splitting" every path simulation on the final timestep.

At the fine level: for every simulated path $\widehat{W}_f = (\Delta W_0, \Delta W_2, \ldots, \Delta W_{N_f-2})$, we simulate a set of d final increments $(\Delta W^{(i)}_{N_f-1})_{i\in[1,d]}$ which we average to get

$$\mathbb{E}\left[P(\widehat{S}_{N_f})\,\middle|\,\widehat{W}_f\right] \approx \frac{1}{d}\sum_{i=1}^{d} P(\widehat{S}_{N_f}(\widehat{W}_f, \Delta W^{(i)}_{N_f-1})) \tag{20}$$

At the coarse level we use $\widehat{W}_c = (\Delta W_0 + \Delta W_1, \ldots, \Delta W_{N_f-4} + \Delta W_{N_f-3})$. As before (still assuming a constant drift and volatility on the final coarse step), we improve the convergence rate of $\mathbb{V}(\widehat{Y}_\ell)$ by reusing ΔW_{N_f-2} in our estimation of $\mathbb{E}\left[P(\widehat{S}_{N_c})\mid \widehat{W}_c\right]$. We can do so by constructing the final coarse increments as $(\Delta W^{(i)}_{N_c-1})_{i\in[1,d]} = (\Delta W_{N_f-2} + (\Delta W^{(i)}_{N_f-1}))_{i\in[1,d]}$ and using these to estimate

$$\mathbb{E}(P(\widehat{S}_{N_c})|\widehat{W}_c) = \mathbb{E}\left[P(\widehat{S}_{N_c})\,\middle|\,\widehat{W}_c, \Delta W_{N_f-2}\right] \approx \frac{1}{d}\sum_{i=1}^{d} P\left(\widehat{S}_{N_c}(\widehat{W}_c, \Delta W^{(i)}_{N_c-1})\right)$$

To get the Greeks, we simply compute the corresponding pathwise sensitivities.

We now examine the influence of d the number of splittings on the estimated complexity.

Estimator	d	β	MLMC complexity
Value	10	≈ 2.0	$O(\varepsilon^{-2})$
	500	≈ 2.0	$O(\varepsilon^{-2})$
Delta	10	≈ 1.0	$O(\varepsilon^{-2}(\log\varepsilon)^2)$
	500	≈ 1.5	$O(\varepsilon^{-2})$
Vega	10	≈ 1.6	$O(\varepsilon^{-2})$
	500	≈ 2.0	$O(\varepsilon^{-2})$

As expected this method yields higher values of β than simple pathwise sensitivities: the convergence rates increase and tend to the rates offered by conditional expectations as d increases and the approximation gets more precise.

Taking a constant number of splittings d for all levels is actually not optimal; for Greeks we can write the variance of the estimator as

$$\begin{aligned}\mathbb{V}(\widehat{Y}_\ell) = & \frac{1}{N_\ell}\mathbb{V}_{\widehat{W}_f}\left[\mathbb{E}\left[\left(\frac{\partial \widehat{P}_f}{\partial\theta}\right)^{(\ell)} - \left(\frac{\partial \widehat{P}_c}{\partial\theta}\right)^{(\ell-1)} \middle| \widehat{W}_f\right]\right] \\ & + \frac{1}{N_\ell\, d}\mathbb{E}_{\widehat{W}_f}\left[\mathbb{V}\left[\left(\frac{\partial \widehat{P}_f}{\partial\theta}\right)^{(\ell)} - \left(\frac{\partial \widehat{P}_c}{\partial\theta}\right)^{(\ell-1)} \middle| \widehat{W}_f\right]\right] \end{aligned} \tag{21}$$

As explained in Sect. 2.2 we have $\mathbb{V}_{\widehat{W}_f}(\mathbb{E}(\ldots|\widehat{W}_f)) = O(h_\ell^{3/2})$ for the Greeks. We also have $\mathbb{E}_{\widehat{W}_f}(\mathbb{V}(\ldots|\widehat{W}_f)) = O(h_\ell)$ for similar reasons. We optimise the variance at a fixed computational cost by choosing d such that the two terms of the sum are of similar order. Taking $d = O(h_\ell^{-1/2})$ is therefore optimal.

2.4 Vibrato Monte Carlo

Since the previous method uses pathwise sensitivity analysis, it is not applicable when payoffs are discontinuous. To address this limitation, we use the Vibrato Monte Carlo method introduced by Giles [6]. This hybrid method combines pathwise sensitivities and the Likelihood Ratio Method.

We consider again (15). We now use the Likelihood Ratio Method on the last timestep and with the notations of Sect. 2.2 we get

$$\frac{\partial \widehat{V}}{\partial\theta} = \mathbb{E}_{\widehat{W}}\left[\mathbb{E}_{\Delta W_{N-1}}\left[P\left(\widehat{S}_N\right)\frac{\partial(\log p(\widehat{S}_N|\widehat{W}))}{\partial\theta} \middle| \widehat{W}\right]\right] \tag{22}$$

We can write $p(\widehat{S}_N|\widehat{W}))$ as $p(\mu_{\widehat{W}}, \sigma_{\widehat{W}})$. This leads to the estimator

$$\begin{aligned}\frac{\partial \widehat{V}}{\partial\theta} \approx \frac{1}{N_\ell}\sum_{m=1}^{N_\ell} & \left(\frac{\partial\mu_{\widehat{W}^{(m)}}}{\partial\theta}\mathbb{E}_{\Delta W_{N-1}}\left[P\left(\widehat{S}_N\right)\frac{\partial(\log p)}{\partial\mu_{\widehat{W}}} \middle| \widehat{W}^{(m)}\right]\right. \\ & \left. + \frac{\partial\sigma_{\widehat{W}^{(m)}}}{\partial\theta}\mathbb{E}_{\Delta W_{N-1}}\left[P\left(\widehat{S}_N\right)\frac{\partial(\log p)}{\partial\sigma_{\widehat{W}}} \middle| \widehat{W}^{(m)}\right]\right) \end{aligned} \tag{23}$$

We compute $\dfrac{\partial \mu_{\widehat{W}^{(m)}}}{\partial\theta}$ and $\dfrac{\partial \sigma_{\widehat{W}^{(m)}}}{\partial\theta}$ with pathwise sensitivities.
With $\widehat{S}_N^{(m,i)} = \widehat{S}_N(\widehat{W}^{(m)}, \Delta W_{N-1}^{(i)})$, we substitute the following estimators into (23)

$$\begin{aligned}
\mathbb{E}_{\Delta W_{N-1}}\left[P\left(\widehat{S}_N\right) \frac{\partial(\log p)}{\partial \mu_{\widehat{W}}} \,\middle|\, \widehat{W}^{(m)} \right] &\approx \frac{1}{d}\sum_{i=1}^{d} \left(P\left(\widehat{S}_N^{(m,i)}\right) \frac{\widehat{S}_N^{(m,i)} - \mu_{\widehat{W}^{(m)}}}{\sigma^2_{\widehat{W}^{(m)}}} \right)\\
\mathbb{E}_{\Delta W_{N-1}}\left[P\left(\widehat{S}_N\right) \frac{\partial(\log p)}{\partial \sigma_{\widehat{W}}} \,\middle|\, \widehat{W}^{(m)} \right] &\approx \frac{1}{d}\sum_{i=1}^{d} P\left(\widehat{S}_N^{(m,i)}\right)\\
&\quad \times \left(\frac{\left(\widehat{S}_N^{(m,i)} - \mu_{\widehat{W}^{(m)}}\right)^2}{\sigma^3_{\widehat{W}^{(m)}}} - \frac{1}{\sigma_{\widehat{W}^{(m)}}} \right)
\end{aligned}$$

In a multilevel setting: at the fine level we can use (23) directly. At the coarse level, for the same reasons as in Sect. 2.3, we reuse the fine brownian increments to get efficient estimators. We take

$$\begin{aligned}
\widehat{W}_c &= (\Delta W_0 + \Delta W_1, \ldots, \Delta W_{N_f-4} + \Delta W_{N_f-3})\\
(\Delta W_{N_c-1}^{(i)})_{i\in[1,d]} &= (\Delta W_{N_f-2} + (\Delta W_{N_f-1}^{(i)}))_{i\in[1,d]}
\end{aligned} \tag{24}$$

We use the tower property to verify that condition (5) is verified on the last coarse step. With the notations of (19) we derive the following estimators

$$\begin{aligned}
&\mathbb{E}_{\Delta W_{N_c-1}}\left[P\left(\widehat{S}_{N_c}\right) \frac{\partial(\log p_c)}{\partial \mu_{\widehat{W}_c}} \,\middle|\, \widehat{W}_c^{(m)} \right]\\
&\quad = \mathbb{E}\left[\mathbb{E}\left[P\left(\widehat{S}_{N_c}\right) \frac{\partial(\log p_c)}{\partial \mu_{\widehat{W}_c}} \,\middle|\, \widehat{W}_c^{(m)}, \Delta W_{N_f-2} \right] \,\middle|\, \widehat{W}_c^{(m)} \right]\\
&\quad \approx \frac{1}{d}\sum_{i=1}^{d} \left(P\left(\widehat{S}_{N_c}^{(m,i)}\right) \frac{\widehat{S}_{N_c}^{(m,i)} - \mu_{\widehat{W}_c^{(m)}}}{\sigma^2_{\widehat{W}_c{}^{(m)}}} \right)\\
&\mathbb{E}_{\Delta W_{N_c-1}}\left[P\left(\widehat{S}_{N_c}\right) \frac{\partial(\log p)}{\partial \sigma_{\widehat{W}}} \,\middle|\, \widehat{W}_c^{(m)} \right]\\
&\quad = \mathbb{E}\left[\mathbb{E}\left[P\left(\widehat{S}_{N_c}\right) \frac{\partial(\log p)}{\partial \sigma_{\widehat{W}}} \,\middle|\, \widehat{W}_c^{(m)}, \Delta W_{N_f-2} \right] \,\middle|\, \widehat{W}_c^{(m)} \right]\\
&\quad \approx \frac{1}{d}\sum_{i=1}^{d} P\left(\widehat{S}_{N_c}^{(m,i)}\right) \left(-\frac{1}{\sigma_{\widehat{W}_c^{(m)}}} + \frac{\left(\widehat{S}_{N_c}^{(m,i)} - \mu_{\widehat{W}_c^{(m)}}\right)^2}{\sigma^3_{\widehat{W}_c^{(m)}}} \right)
\end{aligned} \tag{25}$$

Our numerical experiments show the following convergence rates for $d = 10$:

Estimator	β	MLMC complexity
Value	≈ 2.0	$O(\varepsilon^{-2})$
Delta	≈ 1.5	$O(\varepsilon^{-2})$
Vega	≈ 2.0	$O(\varepsilon^{-2})$

As in Sect. 2.3, this is an approximation of the conditional expectation technique, and so the same convergence rates was expected.

3 European Digital Call

The European digital call's payoff is $P = \mathbf{1}_{S_T > K}$. The discontinuity of the payoff makes the computation of Greeks more challenging. We cannot apply pathwise sensitivities, and so we use conditional expectations or Vibrato Monte Carlo.

With the same notation as in Sect. 2.2 we compute the conditional expectations of the digital call's payoff.

$$\mathbb{E}(P(\widehat{S}_{N_f})|\widehat{W}) = \Phi\left(\frac{\alpha - K}{\beta}\right) \qquad \mathbb{E}(P(\widehat{S}_{N_c})|\widehat{W}_c, \Delta W_{N_f-2}) = \Phi\left(\frac{\alpha_c - K}{\beta_c}\right)$$

The simulations give

Estimator	β	MLMC complexity
Value	≈ 1.4	$O(\varepsilon^{-2})$
Delta	≈ 0.5	$O(\varepsilon^{-2.5})$
Vega	≈ 0.6	$O(\varepsilon^{-2.4})$

The Vibrato technique can be applied in the same way as with the European call. We get

Estimator	β	MLMC complexity
Value	≈ 1.3	$O(\varepsilon^{-2})$
Delta	≈ 0.3	$O(\varepsilon^{-2.7})$
Vega	≈ 0.5	$O(\varepsilon^{-2.5})$

The analysis presented in Sect. 2.2 explains why we expected $\beta = 3/2$ for the value's estimator. A fraction $O(\sqrt{h})$ of all paths arrive in the area around the payoff where $(\partial\mathbb{E}(P(\widehat{S}_N)|\widehat{W})/\partial\widehat{S}_{N-1})$ is not close to 0; there its derivative is $O(h_\ell^{-1})$ and we have $|\widehat{S}_{N_f} - \widehat{S}_{N_c}| = O(h_\ell)$. For these paths, we thus have $O(1)$ difference between the fine and coarse Greeks' estimates. This explains the experimental $\beta \approx 1/2$.

4 European Lookback Call

The lookback call's value depends on the values that the asset takes before expiry. Its payoff is $P(T) = (S_T - \min_{t\in[0,T]}(S_t))$.

As explained in [4], the natural discretisation $\widehat{P} = (\widehat{S}_N - \min_n \widehat{S}_n)$ is not satisfactory. To regain good convergence rates, we approximate the behaviour within each fine timestep $[t_n, t_{n+1}]$ of width h_f as a simple Brownian motion with constant drift a_n^f and volatility b_n^f conditional on the simulated values $\widehat{S}_n^f$ and $\widehat{S}_{n+1}^f$. As shown in [8] we can then simulate the local minimum

$$\widehat{S}_{n,min}^f = \frac{1}{2}\left(\widehat{S}_n^f + \widehat{S}_{n+1}^f - \sqrt{\left(\widehat{S}_{n+1}^f - \widehat{S}_n^f\right)^2 - 2(b_n^f)^2 h_f \log U_n}\right) \tag{26}$$

with U_n a uniform random variable on $[0, 1]$. We define the fine level's payoff this way choosing $b_n^f = b(\widehat{S}_n^f, t_n)$ and considering the minimum over all timesteps to get the global minimum of the path.

At the coarse level we still consider a simple Brownian motion on each timestep of width $h_c = 2h_f$. To get high strong convergence rates, we reuse the fine increments by defining a midpoint value for each step

$$\widehat{S}_{n+1/2}^c = \frac{1}{2}\left(\widehat{S}_n^c + \widehat{S}_{n+1}^c - b_n^c(\Delta W_{n+1/2} - \Delta W_n)\right), \tag{27}$$

where $(\Delta W_{n+1/2} - \Delta W_n)$ is the difference of the corresponding fine Brownian increments on $[t_{n+1/2}, t_{n+1}]$ and $[t_n, t_{n+1/2}]$. Conditional on this value, we then define the minimum over the whole step as the minimum of the minimum over each half step, that is

$$\begin{aligned}\widehat{S}_{n,min}^c = \min\Bigg[&\frac{1}{2}\left(\widehat{S}_n^c + \widehat{S}_{n+1/2}^c - \sqrt{\left(\widehat{S}_{n+1/2}^c - \widehat{S}_n^c\right)^2 - (b_n^c)^2 h_c \log U_{1,n}}\right),\\ &\frac{1}{2}\left(\widehat{S}_{n+1/2}^c + \widehat{S}_{n+1}^c - \sqrt{\left(\widehat{S}_{n+1}^c - \widehat{S}_{n+1/2}^c\right)^2 - (b_n^c)^2 h_c \log U_{2,n}}\right)\Bigg]\end{aligned} \tag{28}$$

where $U_{1,n}$ and $U_{2,n}$ are the values we sampled to compute the minima of the corresponding timesteps at the fine level. Once again we use the tower property to check that condition (5) is verified and that this coarse-level estimator is adequate.

Using the treatment described above, we can then apply straighforward pathwise sensitivities to compute the multilevel estimator. This gives the following results:

Estimator	β	MLMC complexity
Value	≈ 1.9	$O(\varepsilon^{-2})$
Delta	≈ 1.9	$O(\varepsilon^{-2})$
Vega	≈ 1.3	$O(\varepsilon^{-2})$

For the value's estimator, Giles et al. [7] have proved that $\mathbb{V}(\widehat{Y}_l) = O(h_\ell^{2-\delta})$ for all $\delta > 0$, thus we expected $\beta \approx 2$. In the Black and Scholes model, we can prove that Delta $= (V/S_0)$. We therefore expected $\beta \approx 2$ for Delta too. The strong convergence speed of Vega's estimator cannot be derived that easily and will be analysed in our future research.

Unlike the regular call option, the payoff of the lookback call is perfectly smooth and so therefore there is no benefit from using conditional expectations and associated methods.

5 European Barrier Call

Barrier options are contracts which are activated or deactivated when the underlying asset S reaches a certain barrier value B. We consider here the down-and-out call for which the payoff can be written as

$$P = (S_T - K)^+ \mathbf{1}_{\min\limits_{t\in[0,T]}(S_t) > K} \tag{29}$$

Both the naive estimators and the approach used with the lookback call are unsatisfactory here: the discontinuity induced by the barrier results in a higher variance than before. Therefore we use the approach developed in [4] where we compute the probability p_n that the minimum of the interpolant crosses the barrier within each timestep. This gives the conditional expectation of the payoff conditional on the Brownian increments of the fine path:

$$\widehat{P}^f = (\widehat{S}^f_{N_f} - K)^+ \prod_{n=0}^{N_f-1} \left(1 - \widehat{p}^f_n\right) \tag{30}$$

with

$$\widehat{p}^f_n = \exp\left(\frac{-2(\widehat{S}^f_n - B)^+(\widehat{S}^f_{n+1} - B)^+}{(b^f_n)^2\, h_f}\right)$$

At the coarse level we define the payoff similarly: we first simulate a midpoint value $\widehat{S}^c_{n+1/2}$ as before and then define $\widehat{p}^c_n$ the probability of not hitting B in $[t_n, t_{n+1}]$, that is the probability of not hitting B in $[t_n, t_{n+1/2}]$ and $[t_{n+1/2}, t_{n+1}]$. Thus

$$\widehat{P}^c = (\widehat{S}^c_{N_c} - K)^+ \prod_{n=0}^{N_c-1} \left(1 - \widehat{p}^c_n\right) = (\widehat{S}^c_{N_c} - K)^+ \prod_{n=0}^{N_c-1} \left((1 - \widehat{p}_{n,1})(1 - \widehat{p}_{n,2})\right) \quad (31)$$

with

$$\widehat{p}_{n,1} = \exp\left(\frac{-2(\widehat{S}^c_n - B)^+(\widehat{S}^c_{n+1/2} - B)^+}{(b^c_n)^2\, h_f}\right)$$
$$\widehat{p}_{n,2} = \exp\left(\frac{-2(\widehat{S}^c_{n+1/2} - B)^+(\widehat{S}^c_{n+1} - B)^+}{(b^c_n)^2\, h_f}\right)$$

5.1 Pathwise Sensitivities

The multilevel estimators $\widehat{Y}_\ell = \left(\widehat{P}^f\right)^{(\ell)} - \left(\widehat{P}^c\right)^{(\ell-1)}$ are Lipschitz with respect to all $(\widehat{S}^f_n)_{n=1\ldots N_f}$ and $(\widehat{S}^c_n)_{n=1\ldots N_c}$, so we can use pathwise sensitivities to compute the Greeks. Our numerical simulations give

Estimator	β	MLMC complexity
Value	≈ 1.6	$O(\varepsilon^{-2})$
Delta	≈ 0.6	$O(\varepsilon^{-2.4})$
Vega	≈ 0.6	$O(\varepsilon^{-2.4})$

Giles proved $\beta = \frac{3}{2} - \delta$ $(\delta > 0)$ for the value's estimator. We are currently working on a numerical analysis supporting the observed convergence rates for the Greeks.

5.2 Conditional Expectations

The low convergence rates observed in the previous section come from both the discontinuity at the barrier and from the lack of smoothness of the call around K. To address the latter, we can use the techniques described in Sect. 1. Since path splitting and Vibrato Monte Carlo offer rates that are at best equal to those of conditional expectations, we have therefore implemented conditional expectations and obtained the following results:

Estimator	β	MLMC complexity
Value	≈ 1.7	$O(\varepsilon^{-2})$
Delta	≈ 0.7	$O(\varepsilon^{-2.3})$
Vega	≈ 0.7	$O(\varepsilon^{-2.3})$

We see that the maximum benefits of these techniques are only marginal. The barrier appears to be responsible for most of the variance of the multilevel estimators.

6 Conclusion and Future Work

In this paper we have shown for a range of cases how multilevel techniques can be used to reduce the computational complexity of Monte Carlo Greeks.

Smoothing a Lipschitz payoff with conditional expectations reduces the complexity to $O(\varepsilon^{-2})$. From this technique we derive the Path splitting and Vibrato methods: they offer the same efficiency and avoid intricate integral computations. Payoff smoothing and Vibrato also enable us to extend the computation of Greeks to discontinuous payoffs where the pathwise sensitivity approach is not applicable. Numerical evidence shows that with well-constructed estimators these techniques provide computational savings even with exotic payoffs.

So far we have mostly relied on numerical estimates of β to estimate the complexity of the algorithms. Our current analysis is somewhat crude; this is why our current research now focuses on a rigorous numerical analysis of the algorithms' complexity.

References

1. S. Asmussen and P. Glynn. *Stochastic Simulation*. Springer, New York, 2007.
2. M. Broadie and P. Glasserman. "Estimating security price derivatives using simulation". *Management Science*, 42, 2 (1996), 269–285.
3. P. L'Ecuyer. "A unified view of the IPA, SF and LR gradient estimation techniques". *Management Science*, 36, 11 (1990), 1364–1383.
4. M.B. Giles. "Improved multilevel Monte Carlo convergence using the Milstein scheme". In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, 343–358. Springer-Verlag, 2007.
5. M.B. Giles. "Multilevel Monte Carlo path simulation". *Operations Research*, 56, 3 (2008), 607–617.
6. M.B. Giles. "Vibrato Monte Carlo sensitivities". In P. L'Ecuyer and A. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, 369–382. Springer, 2009.
7. M.B. Giles, K. Debrabant and A. Rössler. "Numerical analysis of multilevel Monte Carlo path simulation using the Milstein discretisation". In preparation.
8. P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.
9. P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, Berlin, 1992.

Weight Monte Carlo Method Applied to Acceleration Oriented Traffic Flow Model

Aleksandr Burmistrov and Mariya Korotchenko

Abstract We consider acceleration oriented vehicular traffic flow (VTF) model and study evolution of the N-particle systems, which are governed by a homogeneous Boltzmann-like equation. For this model we obtain a linear integral equation of the second kind and suggest to solve it by the weight Monte Carlo algorithms. The numerical results show that the approach to simulation suggested by the authors is reasonable to apply to the vehicular traffic problems. Moreover, this modification enabled us to study parametric dependencies of our functionals of interest.

1 Introduction

This paper is devoted to the study and simulation of the vehicular traffic flow (VTF). This study appears to be significant due to the constant growth of traffic in most parts of the world nowadays. It results in the necessity for improvement of the transportation network, considering the principles of its growth and distribution of load on its sections.

There are two main approaches to the VTF simulation – a deterministic and a stochastic ones. A functional relation between some parameters, such as, for example, velocity and distance between the cars in the flow, underlies the *deterministic* type of models. On the other hand, in the frame of *stochastic* models, VTF is considered as a random process. Moreover, the models describing the VTF can be

A. Burmistrov
Institute of Computational Mathematics and Mathematical Geophysics (Siberian Branch of the Russian Academy of Sciences), prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia

Novosibirsk State University, Pirogova st., 2, Novosibirsk, 630090, Russia
e-mail: burm@osmf.sscc.ru

M. Korotchenko
ICM&MG SB RAS, prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia
e-mail: kmaria@osmf.sscc.ru

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_14, © Springer-Verlag Berlin Heidelberg 2012

further classified into three categories: micro-, macro-, and mesoscopic ones (for more details see [1,6]).

Mesoscopic (or *kinetic*) models, a type of models we use in our paper, consider the VTF as a random process. Moreover, these models regard the VTF as a gas, which consists of interacting particles and every particle in this gas corresponds to a car. By an interaction of two cars we understand an event when their state, determined by a number of parameters, is changed. There are two main types of interactions in the kinetic models between the cars in the system, depending on velocity of the leading car: acceleration and breaking. The possibility of overtaking is usually introduced into the model by means of a probability, depending on the density of cars on the road. The equations describing kinetic models are similar to the gas kinetic equations, in particular, to the Boltzmann equation. However, unlike the latter one, the momentum and energy conservation laws do not hold in case of the VTF.

In this paper we develop our methods in the frame of the kinetic VTF model suggested in [9]. A distinctive feature of this model consists in introducing the acceleration variable into the set of phase coordinates along with the space and velocity coordinates of the car. Such a modification of the phase space allowed in [9] to apply this acceleration oriented model to a wider range of the VTF types. This model adequately describes not only a constrained traffic but also a higher car density regimes.

In order to verify approach to the study and simulation of the VTF suggested in this paper further we will consider a single-lane traffic in a spatially homogeneous case without overtaking. Note that the obtained results will be compared with a known analytical solution in case of stochastic equilibrium (i.e. stationary distribution). We would like to underline that information about the equilibrium velocity can be of a great importance, for example, in planning the road capacity.

In the framework of [9], distribution of a single car with acceleration a and velocity v has the probability density $f(a,v,t)$, which solves the integro-differential equation of Boltzmann type:

$$\frac{\partial f}{\partial t}(a,v,t) + a\frac{\partial f}{\partial v}(a,v,t) = \int\limits_{\bar{a},\bar{v},a'} \big[\Sigma(a|a',v,\bar{a},\bar{v},\mathbf{m}_f(t))f(a',v,t) - \tag{1}$$

$$- \Sigma(a'|a,v,\bar{a},\bar{v},\mathbf{m}_f(t))f(a,v,t)\big]\, f(\bar{a},\bar{v},t)\,\mathrm{d}\bar{a}\,\mathrm{d}\bar{v}\,\mathrm{d}a',$$

with the initial distribution $f(a,v,0) = f_0(a,v)$. Here $\bar{a}$ and $\bar{v}$ are the acceleration and the velocity of the leading car (*leader*), correspondingly. By a *leader* here and further on we understand the car situated straight ahead to the current car, which we will call the *follower*. It is the leader and the follower who interact. The function $\Sigma(a|a',v,\bar{a},\bar{v},\mathbf{m}_f(t)) = \Sigma(a' \to a|v,\bar{a},\bar{v},\mathbf{m}_f(t))$ is a weighted interaction rate function and it has the following form

$$\Sigma(a|a',v,\bar{a},\bar{v},\mathbf{m}_f(t)) = \int\limits_{h_{\min}}^{\infty} \sigma(a|h,a',v,\bar{a},\bar{v})Q(h,a',v,\bar{a},\bar{v})D(h|a',v,\mathbf{m}_f(t))\,\mathrm{d}h. \tag{2}$$

Here we used the notations:

- $h_{\min}$ is the minimal distance between two cars at rest, i.e. the mean length of a car;
- $Q(\cdot)$ is the interaction rate, it depends on a current microscopic state of the interacting car pair and the distance h between them;
- $\sigma(\cdot)$ is the probability density of the follower's acceleration in case the interaction between the cars with states (a', v) and $(\bar{a}, \bar{v})$ takes place at distance h;
- $D(\cdot)$ is a conditioned probability density of the distance h. It depends on the follower's state (a', v) and a vector $\mathbf{m}_f(t)$, which value is determined by some moments of the solution f (such as mean velocity, velocity scattering, mean acceleration etc.). Further on the function $D(\cdot)$ will also depend on the car density $\mathscr{K}$.

We should note that in this model, suggested in [9], the car acceleration a is added to the phase coordinates as an independent variable in contrast to the gas dynamics. As a result of this modification there are only acceleration jumps (no velocity jumps as in other kinetic models) produced by the pairwise interactions in the system. Moreover, after the interaction takes place the leader does not change its acceleration. Therefore the function $\Sigma(\cdot)$ is not symmetric. We suggest to designate the interacting cars as ordered pairs (i, j), where the first number stands for the follower and the second one stands for the leader.

This paper aims at constructing the basic integral equation of the second kind. The latter equation will enable us to use well-developed techniques of the weight statistical modelling (see e.g. [5]) for estimating the functionals of solution to the Eq. 1.

2 Basic Integral Equation of the Second Kind

The simulation process of stochastic kinetics of the N-particle system is a homogeneous Markov chain in which transitions are due to elementary pair interactions. Note that we deliberately do not use a gas dynamic term *collision* because it has evidently a confusing meaning in case of the vehicular traffic flow.

The integral equation, which describes evolution of the particle (car in this case) ensemble, uniquely defines all the transition densities in the Markov chain. It means that the distribution density of time intervals between elementary interactions in the system can also be determined using this integral equation.

In order to construct the required basic integral equation of the second kind we introduce a phase space Λ of velocities and accelerations for the ensemble of N cars:

$$(A, V) = (a_1, v_1, \ldots, a_N, v_N) \in \Lambda.$$

Let us consider the distribution density of the N-particle system $P(A, V, t)$. Further we omit the dependence of the function $\Sigma(\cdot)$ on the vector $\mathbf{m}_f(t)$ without loss of generality. In this case the function $P(A, V, t)$ satisfies a master equation (see [4]) of the form

$$\frac{\partial P}{\partial t}(A,V,t)+A\frac{\partial P}{\partial V}(A,V,t) \;=\; \frac{1}{N-1}\sum_{i\neq j}\int [\ \Sigma(a_i|a'_i,v_i,a_j,v_j)P(A'_i,V,t)- \qquad (3)$$
$$-\Sigma(a'_i|a_i,v_i,a_j,v_j)P(A,V,t)\ ]\ \mathrm{d}a'_i,$$

here $A'_i = (a_1,\dots,a_{i-1},a_i{}',a_{i+1},\dots,a_N)$. To complete the problem statement we add an *initial condition* $P(A,V,0)=P_0(A,V)$ as well as *boundary conditions* to the Eq. 3. The latter conditions should eliminate both negative velocities and ones exceeding some maximum value $V_{\max}$: $P(A,V,t)=0$ if there is such a number i that either condition ($v_i = 0$ and $a_i < 0$) or condition ($v_i = V_{\max}$ and $a_i > 0$) is fulfilled. It is a well-known fact that under "vehicular chaos" assumption (see [10]), saying that the pair state density for two cars decouples into a product of two single car densities, solution to Eq. 3 turns into solution to Eq. 1 when $N\to\infty$ [4].

2.1 From Master Equation to Basic Integral Equation

Let us rewrite the Eq. 3 in the form

$$\frac{\partial P}{\partial t}(A,V,t)+A\frac{\partial P}{\partial V}(A,V,t)+\upsilon(A,V)P(A,V,t)=J_N(A,V,t), \qquad (4)$$

here we used the following notations: $J_N(A,V,t)=\int F(A'\to A|V)P(A',V,t)\,\mathrm{d}A'$,

$$F(A'\to A|V)=\frac{1}{N-1}\sum_{i\neq j}\Sigma(a'_i\to a_i|v_i,a_j,v_j)\Delta_i(A);\quad \Delta_i(A)=\prod_{m\neq i,m=1}^{N}\delta(a'_m-a_m);$$
$$\upsilon(A,V)=\frac{1}{N-1}\sum_{i\neq j}\upsilon_{(i,j)};\quad \upsilon_{(i,j)}=\int\Sigma(a_i\to a''_i|v_i,a_j,v_j)\,\mathrm{d}a''_i.$$

Here $\delta(\cdot)$ is a Dirac delta function. Taking into account the initial conditions and parametric dependence between velocity, acceleration and time $V=V'+A(t-t')$, we can integrate the Eq. 4 with respect to time. As a result, we obtain the integral equation for P:

$$P(A,V,t)=\int_0^t\delta(t')\,\mathrm{d}t'\int P_0(A,V')\delta_V(A,V',t,t')E_\upsilon(A,V',t,t')\,\mathrm{d}V'+$$
$$\int_0^t E_\upsilon(A,V',t,t')\,\mathrm{d}t'\int\delta_V(A,V',t,t')\,\mathrm{d}V'\int F(A'\to A|V')P(A',V',t')\,\mathrm{d}A',$$

here we used the following notations: $\delta_V(A,V',t,t')=\prod_{m=1}^{N}\delta(v_m-v'_m-a_m(t-t'))$,

$$E_{\upsilon}(A, V', t, t') = \exp\left\{ -\int\limits_{t'}^{t} \upsilon(A, V' + A(\tau - t'))\,\mathrm{d}\tau \right\}.$$

Let us consider in our system the interaction density $\Phi(A, V, t) = \upsilon(A, V)$ $P(A, V, t)$ and the function $\Psi(A, V, t)$, for which the following integral relation holds

$$\Phi(A, V, t) = \int\limits_{0}^{t} \int K_t(t' \to t | A, V') K_V(V' \to V | A, t - t') \Psi(A, V', t')\,\mathrm{d}V'\,\mathrm{d}t'. \tag{5}$$

Then $\Psi(A, V, t)$ satisfies the equation $\Psi = \mathbf{K}_1\Psi + \Psi_0$:

$$\Psi(A, V, t) = \int\limits_{\mathcal{A}\times(0,\infty)} K_1(A, V, t | A', V', t') \Psi(A', V', t')\,\mathrm{d}A'\,\mathrm{d}t'\,\mathrm{d}V' + \Psi_0(A, V, t) \tag{6}$$

with a free term $\Psi_0(A, V, t) = \delta(t) P_0(A, V)$ and the kernel

$$K_1(A, V, t | A', V', t') = K_t(t' \to t | A', V') K_V(V' \to V | A', t - t') K_A(A' \to A | V).$$

Note that the kernel K_1 is a product of distribution densities of new values t, V and A correspondingly:

$$K_t(t' \to t | A', V') = \Theta(t - t') \upsilon(A', V' + A'(t - t')) E_{\upsilon}(A', V', t, t'),$$

$$K_V(V' \to V | A', t - t') = \delta_V(A', V', t, t'), \quad K_A(A' \to A | V) = \frac{F(A' \to A | V)}{\upsilon(A', V)},$$

here $\Theta(\cdot)$ is a Heaviside step function.

Thus, the transition in our Markov chain consists of several elementary transitions in the following order: $(A', V', t') \to (A', V', t) \to (A', V, t) \to \{\varpi\} \to (A, V, t)$. Note, the interacting pair number $\varpi = (i, j)$ is chosen according to the probabilities

$$p(\varpi) = p(i, j) = \frac{1}{N - 1} \cdot \frac{\upsilon_{(i,j)}}{\upsilon(A', V)}. \tag{7}$$

2.2 Decomposition of the Distribution Density

Let us denote

$$\Phi(A, V, t) = \upsilon(A, V) P(A, V, t) = \sum_{\varpi} \frac{\upsilon(\varpi)}{N - 1} P(A, V, t) = \sum_{\varpi} F_{\Phi}(\varpi, A, V, t),$$

here the summation is performed over indices $\varpi = (i, j)$ of all possible ordered pairs of cars in the system. Let the function $F_{\Psi}(\varpi, A, V, t)$ is related to the function $\Psi(A, V, t)$ in the same way as the function $F_{\Phi}(\varpi, A, V, t)$ is related to the function $\Phi(A, V, t)$.

Let us now include the variable ϖ to the set of phase coordinates (A, V, t) of our system (see [8] for more details). Further we will consider Markov chain in this modified phase space $\mathbf{Z} \times [0, T] \ni (Z, t) = (\varpi, A, V, t)$.

The initial state $Z_0 = (\varpi_0, A_0, V_0)$ (i.e. the point of the first interaction in the system at $t_0 = 0$) in our modified phase space is simulated according to the distribution density $P_0(A, V) \cdot \delta(\varpi_0)$. Note, that ϖ_0 can be chosen arbitrary since it does not affect the distribution of the next interaction. The density function of the point (Z_0, t_0) is denoted by $F_0(Z, t) = \delta(t) \cdot P_0(A, V) \cdot \delta(\varpi_0)$.

The modification mentioned above results in decomposition of the phase space according to the pair number ϖ and makes it possible to derive a new basic integral equation of the second kind for the function $F(Z, t) = F_{\Psi}(Z, t)$: $F = \mathbf{K}F + F_0$. We can rewrite the latter equation as follows

$$F(Z, t) = \int_0^t \int_{\mathbf{Z}} F(Z', t') K(Z', t' \to Z, t)\, \mathrm{d}Z'\, \mathrm{d}t' + F_0(Z, t). \tag{8}$$

Here $\mathrm{d}Z = \mathrm{d}V\, \mathrm{d}A\, \mathrm{d}\mu(\varpi)$ and integration with respect to μ means the summation over all possible ordered pairs (i, j). The kernel $K(Z', t' \to Z, t)$ of the Eq. 8 is a product of transitional densities

$$K = K_t(t' \to t | A', V') \cdot K_V(V' \to V | A', t - t') \cdot K_{\varpi}(\varpi) \cdot K_a({a_i}' \to a_i | \varpi, V) \cdot \Delta_i(A),$$

i.e. it contains δ-functions as factors only.

Despite the presence of generalized functions, it is possible to treat $\mathbf{K}$ as an operator from $L_1(\mathbf{Z} \times [0, T])$ to $L_1(\mathbf{Z} \times [0, T])$ (see [7]). Moreover, due to the finiteness of the time interval, the norm $\|\mathbf{K}\|_{L_1} < 1$. Therefore, the Neumann series

$$F(Z, t) = \sum_{n=0}^{\infty} \mathbf{K}^n F_0(Z, t) = \sum_{n=0}^{\infty} F_n(Z, t)$$

for the integral Eq. 8 converges with respect to the L_1 norm. Note, that $F_n(Z, t)$ is a distribution density of the nth interaction in the system. This fact makes it possible to construct weight estimates using the integral Eq. 8 rather than the Eq. 6 for the function Ψ.

The transition of the system from the state Z' to the state Z is performed as follows:

1. The instant t of the next interaction in the system is chosen according to the exponential transition density $K_t(t' \to t | A', V')$;
2. The velocities of all cars are calculated at time t according to the transition density $K_V(V' \to V | A', t - t')$;

3. The pair number $\varpi = (i, j)$ of the interacting cars is chosen by the probabilities Eq. 7;
4. New accelerations of all cars are determined according to the distribution density $K_A(A' \to A|\varpi, V)$ as follows:
   - For the car with number i (the follower in the pair $\varpi = (i, j)$) its acceleration a_i is changed according to the transition density $K_a(a_i{}' \to a_i|\varpi, V) = \Sigma(a_i{}' \to a_i|v_i, a_j, v_j)/\upsilon_{(i,j)}$;
   - The accelerations of other cars do not change.

3 Estimation of Functionals

Usually when solving the Eq. 1, following functionals of one-particle distribution function f

$$I_{\mathbf{h}}(T) = \int\int \mathbf{h}(a_1, v_1) f(a_1, v_1, T)\, \mathrm{d}a_1\, \mathrm{d}v_1 = \int\limits_{\Lambda} \mathbf{h}(a_1, v_1) P(A, V, T)\, \mathrm{d}A\, \mathrm{d}V$$

are of interest. Let us denote

$$\mathbf{H}(A, V) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}(a_i, v_i), \;\; \tilde{\mathbf{H}}(A, V, T - t') = \mathbf{H}(A, V + A(T - t'))E_\upsilon(A, V, T, t').$$

Then, by analogy with [8], we use the relation between the functions P, Ψ, F and obtain a formula for the functional $I_{\mathbf{h}}(T)$ of solution to the Eq. 8:

$$I_{\mathbf{h}}(T) = \int\limits_{\mathbf{Z}} \int\limits_{0}^{T} \tilde{\mathbf{H}}(A, V, T - t') F(Z, t')\, \mathrm{d}Z\, \mathrm{d}t'.$$

Since we have at our disposal an integral equation of the second kind and a Markov chain corresponding to it, we can apply well-developed techniques of weight statistical simulation (see [7], e.g.). This enables us to study dependence of our model on various parameters, estimate parametric derivatives and reduce computational costs of statistical methods (e.g. with the help of the value modelling algorithms).

3.1 Majorant Frequency Principle

The majorant frequency principle discussed in this subsection was suggested in the study [3] for simulation of collisional relaxation in rarified gas flows. This principle makes it possible not to compute the value of $\upsilon(A, V)$ on every step of our process. As a result, the computational cost of the majorant frequency principle is linearly dependent on the number of particles in the system. Such a computational cost is similar to the simplest case of simulation of the maxwellian particles.

Suppose there exists such a constant υ_{max} that for all possible pairs $\varpi = (i, j)$ in the system holds $\upsilon_{max} \geq \upsilon(\varpi)$. Denote

$$\upsilon^* = \sum_{i \neq j} \frac{\upsilon_{max}}{(N-1)} = N \cdot \upsilon_{max} \geq \upsilon(A, V).$$

Then we can rewrite the Eq. 4 in an equivalent form

$$\frac{\partial P}{\partial t} + A\frac{\partial P}{\partial V} + \upsilon^* P(A, V, t) = \upsilon^* \int K^*(A' \to A|V) P(A', V, t)\, \mathrm{d}A',$$

here

$$\begin{aligned} K^*(A' \to A|V) \;&=\; \sum_{\varpi=(i,j)} K^*_{\varpi} \cdot \Delta_i(A) \times \\ &\times \left[\left(1 - \frac{\upsilon'(\varpi)}{\upsilon_{max}}\right) \delta({a'}_i - a_i) + \frac{\upsilon'(\varpi)}{\upsilon_{max}} \frac{\Sigma({a'}_i \to a_i|v_i, v_j, {a_j}')}{\upsilon'(\varpi)} \right], \end{aligned}$$

$K^*_{\varpi} = \{N(N-1)\}^{-1}$ is equal probability to choose a pair number,
$\upsilon'(\varpi) = \upsilon({a_i}', v_i, {a_j}', v_j)$,
$p = \upsilon'(\varpi)/\upsilon_{max}$ is the probability that an interaction takes place in the chosen pair of cars,
$(1 - p)$ is the probability of a "fictitious" interaction in the chosen pair of cars (if an interaction of this kind takes place, then no change in acceleration in the chosen pair is made).

The function $F^*(Z, t)$, which is related to $\Psi^*(A, V, t)$ and $\Phi^*(A, V, t) = \upsilon^* P(A, V, t)$ in a similar way as it was described in Sect. 2.2, satisfies the Eq. 8 with kernel

$$K^*(Z', t' \to Z, t) = K^*_t(t' \to t) K_V(V' \to V|A', t - t') K^*_{\varpi} K_a({a_i}' \to a_i|\varpi, V) \Delta_i(A),$$

here $K^*_t(t' \to t) = \Theta(t - t')\upsilon^* \exp\{-\upsilon^*(t - t')\}$. The functionals $I_{\mathbf{h}}(T)$ of our interest can be also expressed using the equation $F^* = \mathbf{K}^* F^* + F_0$ as follows

$$I_{\mathbf{h}}(T) = \int\limits_{\mathbf{Z}} \int\limits_0^T \mathbf{H}(A, V + A(T - t')) \exp\left\{-\upsilon^*(T - t')\right\} F^*(Z, t')\, \mathrm{d}Z\, \mathrm{d}t'.$$

3.2 Weight Algorithms and Parametric Estimators

We introduce a Markov chain $\{Z_n, t_n\}$, $n = 0, 1 \ldots, \kappa$, where κ is the number of interaction preceding the passage of the system beyond the time boundary T, with the normalized transition density

$$P(Z',t' \to Z,t) = P_1(t|A',V',t')P_2(V|A',V',t)P_3(\varpi|A',V,t)P_4(a_i|\varpi,A',V,t)\Delta_i(A),$$

and the normalized distribution density $P^{(0)}(A,V)\delta(t)\delta(\varpi_0)$ of the initial state (Z_0,t_0). We define random weights Q_n by the formulas

$$Q_0 = \frac{P_0(A_0,V_0)}{P^{(0)}(A_0,V_0)}, \quad Q_n = Q_{n-1}Q(Z_{n-1},t_{n-1};Z_n,t_n), \tag{9}$$

$$Q(Z',t';Z,t) = \left\{\frac{K_t(t' \to t|A',V')}{P_1(t|A',V',t')}\right\}\left\{\frac{K_V(V' \to V|A',t-t')}{P_2(V|A',V',t)}\right\} \times$$
$$\times \left\{\frac{K_\varpi(\varpi)}{P_3(\varpi|A',V,t)}\right\}\left\{\frac{K_a({a_i}' \to a_i|\varpi=(i,j),V}{P_4(a_i|\varpi,A',V,t)}\right\}.$$

For numerical estimation of the functional $I_{\mathbf{h}}(T)$ we can use the collision estimator ξ or absorption estimator η, which are functionals of the Markov chain trajectory. These estimators are well known in the theory of the Monte Carlo methods (see, e.g., [7, 8]):

$$\xi = \sum_{n=0}^{\kappa} Q_n\tilde{\mathbf{H}}(A_n,V_n,T-t_n), \quad \eta = \frac{Q_\kappa\tilde{\mathbf{H}}(A_\kappa,V_\kappa,T-t_\kappa)}{q(A_\kappa,V_\kappa,t_\kappa)}.$$

here $q(A,V,t') = 1 - \int\limits_0^{T-t'} P_1(\tau|A,V,t')\,\mathrm{d}\tau$. Using the inequalities $K \geq 0$, $\|\mathbf{K}\|_{L_1} < 1$ and theoretical results of [7] we can obtain the following theorem.

Theorem 1. *If $P^{(0)}(A,V) \neq 0$ whenever $P_0(A,V) \neq 0$; and $Q(Z',t';Z,t) < +\infty$ for $Z',Z \in \mathbf{Z}$, $t',t < T$, then $\mathbf{E}\xi = I_{\mathbf{h}}(T)$. If, additionally, $q(A,V,t') > 0$ for $(A,V) \in \Lambda$ and $t' < T$, then $\mathbf{E}\eta = I_{\mathbf{h}}(T)$. Moreover, if the weights Eq. 9 are uniformly bounded and $\mathbf{H} \in L_\infty$, then there exists such T^* that $\mathbf{V}\xi < +\infty$ and $\mathbf{V}\eta < +\infty$ for $T < T^*$.* □

It was noted in [8] that the estimators' variances remain finite if a direct simulation of $P(Z',t' \to Z,t) \equiv K(Z',t' \to Z,t)$ is performed for $T > T^*$.

The weight method can be effectively used to analyze the dependence of results on the problem parameters. By using standard techniques from the theory of weight methods (see, e.g., [7]), we can construct estimators for the corresponding parametric derivatives. Namely, if $\rho(\mathbf{K}) < 1$, $\rho(\mathbf{K}_p) < 1$, $P_0/P^{(0)} \in L_1$ and norms $||{K'}_c||$, $||{\mathbf{H}'}_c||_{L_\infty}$ are uniformly bounded in some range of c: $c_k - \varepsilon \leq c \leq c_k + \varepsilon$, then, under the assumptions of Theorem 1, we have (according to [7])

$$\mathbf{E}\left(\frac{\partial\eta}{\partial c}\right) = \frac{\partial I_{\mathbf{h}}(T,c)}{\partial c}, \quad \mathbf{V}\left(\frac{\partial\eta}{\partial c}\right) < +\infty.$$

Analogous expressions hold in case of the collision estimator ξ.

4 Numerical Results

As a test for the algorithm described at the end of the Sect. 2 we took three problems, which have a known analytical solution in case of stochastic equilibrium (see [9, 10]). In the first two examples we consider a spatially homogeneous nearly free stationary VTF. In this VTF all cars have velocities around the mean velocity $V \gg 0$.

4.1 Estimation of Velocity Distribution

In this section we choose functions $\mathbf{h}(a, v)$ equal to indicators of some partitioning of the velocity interval $0 \leq v_i \leq V_{max} = 40\,\text{m/s}$.

4.1.1 Maxwellian Interaction

The type of interaction in this test problem is called *maxwellian* because the rate Q is a constant value: $Q = 1/\mathscr{T}$, and $\mathscr{T}$ is a constant time threshold [9]. The function σ here is a probability density of a new acceleration with the values $\pm a_0$:

$$\sigma(a|h, a', v, \bar{a}, \bar{v}) = \sigma(a|\bar{v} - v) = \Theta(\bar{v} - v)\delta(a - a_0) + \Theta(v - \bar{v})\delta(a + a_0).$$

For such coefficients we have $\upsilon(\varpi) = 1/\mathscr{T}$ and $\upsilon^* = N/\mathscr{T}$. As an initial velocity distribution we use a normal density with the mean $\text{V} = 20\,\text{m/s}$ and the variance $\sigma_0^2 = 0.1\,\text{m}^2/\text{s}^2$. Initial accelerations are equal to 0.

The solution to the Eq. 1 in stochastic equilibrium is given by (see [9]):

$$f(v, a) = \frac{\pi}{4\sqrt{3}\sigma_v} \cosh^{-2}\left\{\frac{\pi}{2\sqrt{3}} \frac{(v - \text{V})}{\sigma_v}\right\} \frac{\delta(a - a_0) + \delta(a + a_0)}{2}, \tag{10}$$

with the mean V and the variance $\sigma_v^2 = (\pi \mathscr{T} a_0)^2/3$. The numerical estimate of the velocity distribution is shown in Fig. 1a ($\mathscr{T} = 2\,\text{s}$, $\sigma_v = 1.088\,\text{m/s}$). We simulated $M = 10^3$ trajectories of our system consisting of $N = 10^3$ cars.

4.1.2 Hard Sphere Interaction

The only difference from the previous paragraph is in the rate $Q(v, \bar{v}) = r_0|\bar{v} - v|$. In this case we apply the majorant frequency principle, described in Sect. 3.1, with the following parameters: $\upsilon(\varpi) = r_0|v_i - v_j|$, $\upsilon_{max} = r_0 V_{max}$, $\upsilon^* = N V_{max} r_0$. The solution to the Eq. 1 is given by (see [9]):

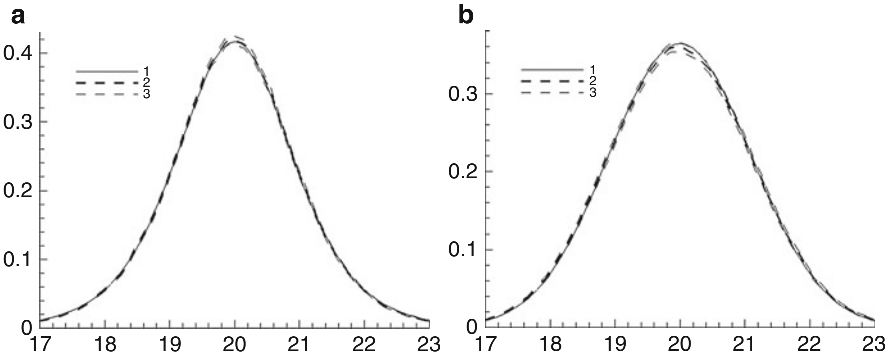


Fig. 1 Numerical estimate of the velocity distribution $f(v)$: *1* – exact solution Eq. 10 for **a**, Eq. 11 for **b**; *2* – numerical estimate; *3* – confidence interval $\pm 3\sigma_\eta$.

$$f(v,a) = \frac{1}{\sqrt{2\pi}\sigma_v} \exp\left\{-\frac{(v-\mathrm{V})^2}{2\sigma_v^2}\right\} \frac{\delta(a-a_0)+\delta(a+a_0)}{2}, \qquad (11)$$

with the mean V and the variance $\sigma_v^2 = a_0/r_0$. The numerical estimate of the velocity distribution is shown in Fig. 1b ($r_0 = 0.25\,\mathrm{m}^{-1}$, $\sigma_v = 1.095\,\mathrm{m/s}$).

4.1.3 Distance Threshold Interaction

Here a distance oriented interaction model [10] is considered with the following parameters: $Q = 1/\mathscr{T}$,

$$D(h) = \frac{1}{\bar{H}-h_{\min}} \exp\left\{-\frac{h-h_{\min}}{\bar{H}-h_{\min}}\right\} \Theta(h-h_{\min}), \quad \bar{H} = 1/\mathscr{K},$$
$$\sigma(a|h,v) = \Theta(h-H(v))\cdot\delta(a-a_0) + \Theta(H(v)-h)\cdot\delta(a+a_0)$$

and a simple distance interaction threshold $H(v) = \alpha\cdot v + h_{\min}$. Taking these functions into account we find the form of the weighted interaction density Eq. 2

$$\Sigma(a_i{}' \to a_i|v_i,a_j,v_j) = \frac{p}{\mathscr{T}}\delta(a+a_0) + \frac{(1-p)}{\mathscr{T}}\delta(a-a_0), \quad p = \int\limits_{h_{\min}}^{H(v_i)} D(h)\,\mathrm{d}h.$$

For such coefficient $\Sigma(\cdot)$ the solution to the Eq. 1 is given by (see [10]):

$$f(v) = \frac{\exp\left\{-\frac{v}{a_0\mathscr{T}} - 2\beta\mathrm{e}^{-\frac{v}{a_0\mathscr{T}\beta}}\right\}}{a_0\mathscr{T}\left(\mathrm{e}^{-2\beta}+\beta(2\beta)^{-\beta}\gamma(\beta,2\beta)\right)}, \quad \beta = \frac{\bar{H}-h_{\min}}{\alpha a_0\mathscr{T}}.$$

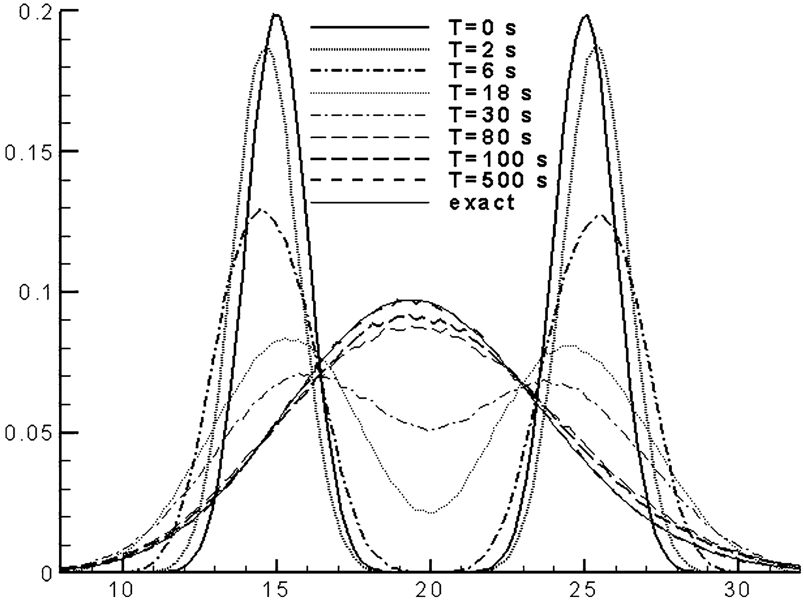


Fig. 2 Numerical estimate of the velocity distribution evolution $f(v)$.

here $\gamma(\beta, 2\beta)$ is an incomplete gamma function. The values of $\upsilon(\varpi)$ and υ^* are equal to those for the case of maxwellian interaction. As an initial velocity distribution we use a mixture of two normal distributions with the means $V_1 = 15\,\text{m/s}$, $V_2 = 25\,\text{m/s}$ and the variance $\sigma_0 = 1\,\text{m/s}$. Initial accelerations are equal to 0. The numerical estimate of the velocity distribution evolution is shown in Fig. 2 ($\mathscr{T} = 2.5\,\text{s}$, $h_{\min} = 6.5\,\text{m}$, $\mathscr{K} = 0.025\,\text{m}^{-1}$, $a_0 = 0.3\,\text{m/s}^2$, $\alpha = 1.2\,\text{s}$).

4.2 Fundamental Diagram

In this section we consider a numerical estimation of the traffic density $\mathscr{K}$V dependence (here V is the mean velocity which is estimated with the help of corresponding function $\mathbf{h}(a, v) = v$) on the car density $\mathscr{K}$ which is called a *fundamental diagram*. Figure 3 shows a typical shape of this curve for the following parameters: $\mathscr{T} = 2.5\,\text{s}$, $h_{\min} = 6.5\,\text{m}$, $a_0 = 0.1\,\text{m/s}^2$, $\alpha_1 = 1.2\,\text{s}$, $\alpha_2 = 1.5\,\text{s}$, $\alpha_3 = 1.8\,\text{s}$. For some value of $\mathscr{K}$ there is a change from a free flow (with no dependence on α) to an interaction oriented flow (with strong dependence on α). For the latter flow cars can not drive in their own way, but they should agree their velocity with the flow velocity.

In the general case each driver in the flow has its own threshold parameter value α. Note that low values of α correspond to a more aggressive driver, while high values of this parameter stand for a more conservative driving manner. Taking into account the numerical results, we can summarize the following remark.

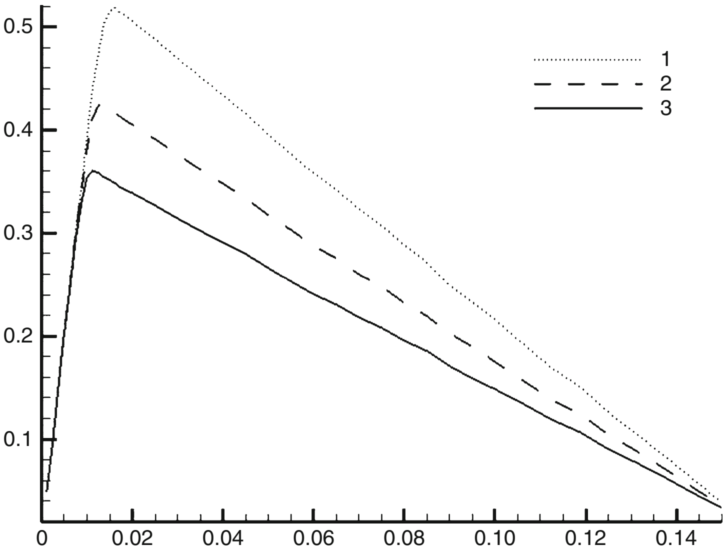


Fig. 3 Fundamental diagram ($M = 10^2, N = 10^2$): *1* – α_1, *2* – α_2, *3* – α_3.

Remark 1. Let the ith driver has its own parameter $\alpha_i \in [\alpha_{\min}, \alpha_{\max}]$, $i = 1, \ldots, N$. Then the fundamental diagram of this VTF will be in the area between two curves corresponding to $\alpha_{\min}$ and $\alpha_{\max}$.

4.3 Parametric Analysis

As an example of parametric dependence study here we consider the case of the maxwellian interaction described above. Therefore in this section we consider the coefficient $\Sigma_c = c \cdot \Sigma_1 = c/\mathscr{T}$. The value of the constant c influences the simulation of the time interval τ between interactions since $\upsilon_c^* = c \cdot \upsilon^*$ is included into the distribution density K_t. As a result, the functionals $I_{\mathbf{h}}(T, c)$ depend on the value of c. In order to study this parametric dependence we will use the weight method with respect to c.

Define the random weights according to the algorithm described in the Sect. 3.2. As a distribution density K_t for simulation we will consider the density with parameter υ^* corresponding to the value of $c = 1$. In this case the random weights have the following form:

$$Q_0 = 1, \quad Q_n = Q_{n-1} \frac{\Sigma_c}{\Sigma_1} \frac{\exp\{-N\Sigma_c \tau_n\}}{\exp\{-N\Sigma_1 \tau_n\}}, \quad n = 1, \ldots, \kappa,$$

$$Q^{(T)} = Q_\kappa \frac{\exp\{-N\Sigma_c\,(T - t_\kappa)\}}{\exp\{-N\Sigma_1\,(T - t_\kappa)\}} = \exp\{\kappa \ln c - NT\Sigma_1(c-1)\},$$

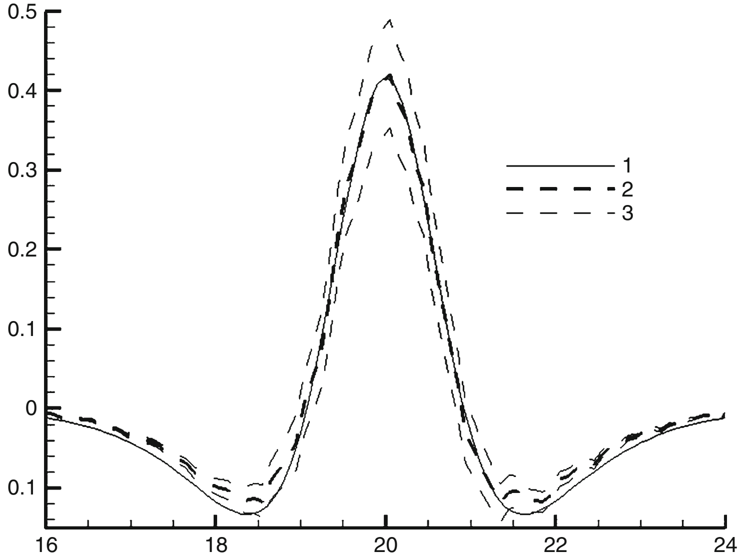


Fig. 4 The estimation of $\partial f(v,c)/\partial c$ when $c = 1$: *1* – exact derivative Eq. 13; *2* – derivative estimation Eq. 12; *3* – confidence interval $\pm 3\sigma_\eta$.

here τ_n is the time interval between $(n-1)$th and nth interactions in the system of N cars. In order to estimate the functionals $I_{\mathbf{h}}(T,c)$ we can use an absorption weight estimator, which has in this particular case the following form $\eta(c) = Q^{(T)}(c)\eta(1)$. As a result, when simulating a Markov chain corresponding to the value of $c = 1$, we can simultaneously estimate the functionals $I_{\mathbf{h}}(T,c)$ for other values of parameter c (analogous algorithm for Boltzmann equation is described in [2]).

Moreover, for the purpose of studying the dependence of the functionals $I_{\mathbf{h}}(T,c)$ on the parameter c, we can estimate the value of $\partial I_{\mathbf{h}}(T,c)/\partial c$ using a corresponding derivative of the estimator $\eta(c)$:

$$\frac{\partial \eta}{\partial c}(c) = Q^{(T)}(c)\left[\frac{\kappa}{c} - NT\Sigma_1\right] H(A_\kappa, V_\kappa + A_\kappa(T - t_\kappa)) = \tilde{Q}(c)\eta(1). \tag{12}$$

With the help of analytical solution Eq. 10, we obtain the derivative $\partial f/\partial c$:

$$\frac{\partial f}{\partial c}(v,a,c) = f(v,a,c)\cdot\left[\frac{1}{c} - \tanh\left(\frac{c\Sigma_1(v-\mathrm{V})}{2a_0}\right)\cdot\frac{\Sigma_1(v-\mathrm{V})}{a_0}\right]. \tag{13}$$

The estimation of $\partial f(v,c)/\partial c$ when $c = 1$ is shown in Fig. 4. We simulated $M = 10^5$ trajectories of our system consisting of $N = 500$ cars.

5 Conclusion

The results of Sect. 4 show the efficiency of transition to the basic integral equation and Markov chain simulation in VTF problems. Moreover, this transition enables us to study parametric dependencies of our functionals of interest and apply various techniques to reduce computational costs. Note also, that we do not use in the simulation procedure an external (to the initial model) discrete time parameter Δt which was used in [9] for splitting the movement and the interaction process. (It resulted in a simpler simulation process.)

Possible directions for future studies should take into account various aspects such as more realistic interaction profiles (including random parameters); mixture of both driver behaviors and vehicle classes; multilane traffic with overtaking; cluster formation on the road (according to kinetic Smoluchowski equation); spatial inhomogeneity (off- and on-ramps).

Acknowledgements The authors acknowledge the kind hospitality of the Warsaw University and the MCQMC'2010 conference organizers. This work was partly supported by Russian Foundation for Basic Research (grants 09–01–00035, 09–01–00639, 11–01–00252) and SB RAS (Integration Grant No. 22). The authors would also like to thank Dr. S. V. Rogasinsky, Professor M. S. Ivanov and Corresponding member of RAS G. A. Mikhailov for valuable discussions.

References

1. Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. Phys. Rep. **329** (4–6) 199–329 (2000)
2. Ivanov, M.S., Korotchenko, M.A., Mikhailov, G.A., Rogasinsky, S.V.: New Monte Carlo global weight method for the approximate solution of the nonlinear Boltzmann equation. Russ. J. Numer. Anal. Math. Modelling, **19** (3), 223–238 (2004)
3. Ivanov, M.S., Rogasinsky, S.V.: The Direct Statistical Simulation Method in Dilute Gas Dynamics. Publ. Comput. Center, Sib. Branch, USSR Acad. Sci., Novosibirsk (1988) [in Russian]
4. Kac, M.: Probability and Related Topics in Physical Sciences. Interscience, New York (1959)
5. Korotchenko, M.A., Mikhailov, G.A., Rogasinsky, S.V.: Value modifications of weighted statistical modeling for solving nonlinear kinetic equations. Russ. J. Numer. Anal. Math. Modelling, **22** (5), 471–486 (2007)
6. Mahnke, R., Kaupužs, J., Lubashevsky, I.: Probabilistic description of traffic flow. Phys. Rep. **408** (1–2), 1–130 (2005)
7. Mikhailov, G.A.: Parametric Estimates by the Monte Carlo Method. VSP, Utrecht (1999)
8. Mikhailov, G.A., Rogasinsky, S.V.: Weighted Monte Carlo methods for approximate solution of the nonlinear Boltzmann equation. Sib. Math. J. **43** (3), 496–503 (2002)
9. Waldeer, K.T.: The direct simulation Monte Carlo method applied to a Boltzmann-like vehicular traffic flow model. Comput. Phys. Commun. **156** (1), 1–12 (2003)
10. Waldeer, K.T.: A vehicular traffic flow model based on a stochastic acceleration process. Transp. Theory Stat. Phys. **33** (1), 7–30 (2004)

New Inputs and Methods for Markov Chain Quasi-Monte Carlo

Su Chen, Makoto Matsumoto, Takuji Nishimura, and Art B. Owen

Abstract We present some new results on incorporating quasi-Monte Carlo rules into Markov chain Monte Carlo. First, we present some new constructions of points, fully equidistributed LFSRs, which are small enough that the entire point set can be used in a Monte Carlo calculation. Second, we introduce some antithetic and round trip sampling constructions and show that they preserve the completely uniformly distributed property necessary for QMC in MCMC. Finally, we also give some new empirical results. We see large improvements in sampling some GARCH and stochastic volatility models.

1 Introduction

Simple Monte Carlo sampling has two limitations when used in practice. First, it converges only at a slow rate, with root mean squared error $O(n^{-1/2})$. Second, on many challenging problems there is no known way to generate independent samples from the desired target distribution. Quasi-Monte Carlo (QMC) methods have been developed to address the first problem, yielding greater accuracy, while Markov

S. Chen
Stanford University, Stanford, CA, USA
e-mail: suchenpku@gmail.com

M. Matsumoto
University of Tokyo, Tokyo, Japan
e-mail: matumoto@ms.u-tokyo.ac.jp

T. Nishimura
Yamagata University, Yamagata, Japan
e-mail: nisimura@sci.kj.yamagata-u.ac.jp

A.B. Owen
Stanford University, Stanford, CA, USA
e-mail: owen@stanford.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_15, © Springer-Verlag Berlin Heidelberg 2012

chain Monte Carlo (MCMC) methods have been developed for the second problem yielding wider applicability.

It is natural then to seek to combine these two approaches. There were some early attempts by Chentsov [2] and Sobol' [15] around 1970. The problem has been revisited more recently. See for example [11] and [13]. For a survey of recent combinations of QMC and MCMC see [1].

QMC uses n points in $[0,1)^d$, where typically $n \gg d$. MCMC uses one long stream of IID $U[0,1)$ inputs, which we call the 'driving sequence'. It has effectively $n = 1$ with $d \to \infty$, quite unlike QMC. Chentsov's key insight was to use completely uniformly distributed points to drive the MCMC. That is the approach taken in [13].

The contributions of this paper are as follows. First, we present some new point sets, small fully equidistributed LFSRs, to use as driving sequences for MCMC. Second, we show how some antithetic sampling strategies within the driving sequence still give rise to valid driving sequences. Third, we present some new empirical findings.

The outline of the paper is as follows. Section 2 defines some key notions that we need. Section 3 describes the LFSRs that we use. Section 4 presents our antithetic extensions of the driving sequence. We give new empirical results in Section 5. Our conclusions are in Section 6.

2 Background

In this section we describe completely uniformly distributed points and some generalizations that we need. We also give a sketch of MCMC. For more details on the latter, the reader may consult [12, 14].

2.1 Completely Uniformly Distributed Sequences

Here we define some notions of completely uniformly distributed sequences. We assume that the reader is familiar with the star discrepancy D_n^{*d}.

Let $u_i \in [0,1]$ for $i \geq 1$. For integer $d \geq 1$, define

$$\bar{u}_i^{(d)} = (u_i, u_{i+1}, \dots, u_{i+d-1}), \quad \text{and,} \tag{1}$$

$$u_i^{(d)} = (u_{i(d-1)+1}, u_{i(d-1)+2}, \dots, u_{id}). \tag{2}$$

Both $\bar{u}_i^{(d)}$ and $u_i^{(d)}$ are made up of consecutive d-tuples from u_i, but the former are overlapping while the latter are non-overlapping.

Definition 1. The infinite sequence u_i is *completely uniformly distributed* (CUD), if

$$\lim_{n\to\infty} D_n^{*d}(\bar{u}_1^{(d)}, \dots, \bar{u}_n^{(d)}) = 0 \tag{3}$$

for all integer $d \ge 1$.

If u_i are CUD, then

$$\lim_{n\to\infty} D_n^{*d}(u_1^{(d)}, \dots, u_n^{(d)}) = 0 \tag{4}$$

holds for all $d \ge 1$. Conversely (see [2]), if (4) holds for all $d \ge 1$ then u_i are CUD.

For randomized points u_i it is useful to have the following definition.

Definition 2. The infinite sequence u_i is *weakly completely uniformly distributed* (WCUD), if

$$\lim_{n\to\infty} \Pr\big(D_n^{*d}(\bar{u}_1^{(d)}, \dots, \bar{u}_n^{(d)}) > \epsilon\big) = 0 \tag{5}$$

for all $\epsilon > 0$ and integer $d \ge 1$.

To better model driving sequences of finite length, there are also triangular array versions of these definitions. A triangular array has elements $u_{n,i} \in [0,1]$ for $i = 1, \dots, n$ and $n \in \mathscr{N}$ where $\mathscr{N}$ is an infinite set of nonnegative integers. This triangular array is CUD if $\lim_{n\to\infty} D_n^{*d}(\bar{u}_{n,1}^{(d)}, \dots, \bar{u}_{n,n-d+1}^{(d)}) = 0$ for all integer $d \ge 1$ as $n \to \infty$ through values in $\mathscr{N}$. There is a similar definition for weakly CUD triangular arrays.

For further background on CUD sequences see [10]. For triangular arrays and sufficient conditions for weak CUD see [18]. The usual construction for WCUD sequences applies Cranley-Patterson [4] rotation to a CUD sequence [18].

2.2 *Markov Chain Monte Carlo*

A typical MCMC run begins with a starting point X_0. Then, for $i \ge 1$

$$X_i = \phi(X_{i-1}, u_i^{(m)}) \tag{6}$$

where $u_i^{(m)}$ is defined at (2) in terms of an IID driving sequence $u_i \sim U[0,1]$. This version of MCMC assumes that each update consumes exactly m elements of the driving sequence. MCMC sometimes uses more general schemes, and its QMC version can too. See [18]. In this paper we will suppose that (6) holds. The CUD property for a driving sequence has to apply to all integer values $d \ge 1$, not just $d = m$.

The update function $\phi(\cdot, \cdot)$ is chosen so that as $n \to \infty$, the distribution of X_n approaches a desired distribution π. If we are interested in the quantity

$$\mu = \int f(x)\pi(x)\,\mathrm{d}x$$

we estimate it by

$$\hat{\mu} = \frac{1}{n} \sum_{i=b+1}^{b+n} f(X_i)$$

where $b \geq 0$ is a burn-in parameter. For simplicity, we take $b = 0$.

The typical behavior of MCMC is that $f(X_i)$ and $f(X_{i+k})$ have a correlation that decreases as ρ^k, where $|\rho| < 1$. As a result $\hat{\mu}$ ordinarily approaches μ with an RMSE of $O(1/\sqrt{n})$. There are however pathologies in which the chain can get stuck. Such failure to mix can result in lack of convergence. Considerable creativity goes into constructing the update function ϕ, to obtain a rapidly mixing Markov chain. The details are beyond the scope of this article. See [12, 14]. Our focus is on replacing IID driving sequences by CUD ones in chains that do mix well. CUD driving sequences do not repair faulty choices of $\phi()$.

2.3 QMC in MCMC Results

Much of the literature combining QMC with MCMC is empirical. Here we provide a short summary of the theoretical results that underpin the work described in this paper.

Running an MCMC algorithm with deterministic inputs gives output that is not Markovian. As a result, there is potential for error. There is however a safe harbor in replacing IID points by (W)CUD points.

Suppose first that $X_i \in \Omega = \{\omega_1, \ldots, \omega_M\}$. Such finite state spaces are technically simpler. If X_i is sampled by inversion and $\min_{1 \leq j,k \leq M} \Pr(X_i = \omega_j \mid X_{i-1} = \omega_k) > 0$ then Chentsov [2] shows that a CUD driving sequence gives consistency, i.e.,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} 1_{X_i=\omega_j} = \pi(\omega_j) \tag{7}$$

for $j = 1, \ldots, M$. Chentsov [2] also gives a converse. Given a non-CUD sequence, he constructs a Markov chain for which (7) will fail to hold. For random driving sequences, the consistency condition is

$$\lim_{n\to\infty} \Pr\Bigl(|\frac{1}{n} \sum_{i=1}^{n} 1_{X_i=\omega_j} - \pi(\omega_j)| > \epsilon\Bigr) = 0, \quad \forall \epsilon > 0. \tag{8}$$

It is seldom possible to sample the transitions by inversion. The Metropolis-Hastings update [8] is usually used instead. For the Metropolis-Hastings update, consistency (7) still holds (see [13]) under three conditions. First, the driving sequence must be CUD. Second, the function ϕ must be one for which an IID $U(0,1)$ driving sequence achieves weak consistency (8). (It could include some zero transition probabilities.) Finally, there is a technical condition that pre-images in $[0,1]^m$ for transitions from one state to another must all give Jordan measurable sets of $u_i^{(m)}$. To summarize, if Metropolis-Hastings sampling on a finite state space is weakly consistent with IID sampling, then it is consistent with CUD sampling. It is then also weakly consistent with weakly CUD sampling.

The case of continuous state spaces was taken up by Chen et al. [1]. The same conclusion holds. If an MCMC algorithm, either Metropolis-Hastings (their Theorem 2) or Gibbs sampling (Theorem 3), is weakly consistent when driven by IID $U(0,1)$ inputs, then it is consistent when driven by CUD inputs and is weakly consistent when driven by WCUD inputs. In the continuous state space setting consistency means having the empirical probability of hyperrectangles match their probability under π. The dimension of these hyperrectangles equals that of the point X_i, which is not necessarily m. The technical conditions for Metropolis-Hastings involve Jordan measurability of pre-images for multistage transitions, while those for Gibbs sampling require a kind of contraction mapping.

3 New Small LFSR Constructions

Constructions for CUD points are surveyed in [10], but the ones there are not convenient to implement. Tribble [17] used small versions of multiple congruential generators and linear feedback shift registers (LFSRs). His best results were for LFSRs but he had only a limited number of them.

In this section we present some new LFSR type sequences with lengths $2^d - 1$ for all integers $10 \le d \le 32$. Their consecutive blocks of various lengths obey an equidistribution property. That makes them suitable for applications which require low discrepancy for vectors formed by taking overlapping consecutive points.

Let P be an integer and u_i for $i = 0, 1, 2, \ldots$ be a sequence of real numbers in the half-open interval $[0, 1)$ with period P. Let

$$u_i = \sum_{j=1}^{\infty} b_{i,j} 2^{-j} \tag{9}$$

be 2-adic expansion of u_i.

We associate to the sequence (u_i) a multi-set (namely, a set with multiplicity of each element counted) Ψ_k as follows:

$$\Psi_k := \{\bar{u}_i^{(k)} \mid 0 \le i \le P - 1\}.$$

The multi-set Ψ_k consists of k-dimensional points obtained as overlapping k-tuples in the sequence for one period. For some positive integer v, we divide the interval $[0, 1)$ into 2^v equal pieces. This yields a partition of the unit hypercube $[0, 1)^k$ into 2^{kv} cubic cells of equal size. Following [16] (cf. [9]), we say that the sequence (x_i) is *k-dimensionally equidistributed* with v-bit accuracy if each cell contains exactly same number of points of Ψ_k, except for the cell at the origin that contains one less. The largest value of such k is called the dimension of equidistribution with v-bit accuracy and denoted by $k(v)$.

Let $M(k, v)$ denote the set of $k \times v$ binary matrices. The above condition is equivalent to that the multiset of $k \times v$ matrices

$$\Phi_{k,v} := \{(b_{i+r,j})_{r=0,\dots,k-1;j=1,\dots,v} \mid 0 \leq i \leq P-1\} \tag{10}$$

contains every element of $M(k, v)$ with the same multiplicity, except the 0 matrix with one less multiplicity. Since there are $2^{kv} - 1$ nonzero such matrices, we have an inequality $2^{kv} - 1 \leq P$. In the following examples, $P = 2^d - 1$, and hence $k(v) \leq \lfloor d/v \rfloor$. A sequence (x_i) of period $2^d - 1$ is said to be *fully equidistributed* (FE) if the equality holds for all $1 \leq v \leq d$. When d is equal to the number of binary digits of the elements of the sequence, this property is equivalent to the maximal equidistribution property [9, 16].

Definition 3. ($GF(2)$-linear sequence generators)

Let $S := GF(2)^d$ be the state space, $F : S \to S$ be a $d \times d$ $GF(2)$-matrix F (multiplication from left) representing the state transition, and $o : S \to GF(2)^d$ be another $d \times d$-matrix for the output function. Choose an initial state $s_0 \neq 0$. The state transition is given by $s_i = F(s_{i-1})$ for $i \geq 1$. The i-th output $o(s_i) = (b_{i,1}, \dots, b_{i,d})$ is regarded as a real number u_i by

$$u_i = \sum_{j=1}^{d} b_{i,j} 2^{-j}. \tag{11}$$

This generator of the real number sequence u_i ($i \geq 0$) is called *GF(2)-linear generator.*

We discuss below a method to search for such generators with the FE property. Note that we could add random digits beyond the d'th in (11), but they would not affect the FE property.

Assume that F has the maximal period $P = 2^d - 1$. Then, every nonzero element of S is on one orbit; namely, $S = \{s_i = F^i(s_0) \mid 1 \leq i \leq P-1\} \cup \{0\}$ for any nonzero s_0. Now we define a mapping

$$o_{k,v} : S \to M(k, v); \quad s_i \mapsto (b_{i+r,j})_{0 \leq r \leq k-1, 1 \leq j \leq v}, \;\; 0 \mapsto 0,$$

where $b_{i.j}$ is the j-th bit in $o(s_i)$ as in Definition 3. This mapping maps s_i to the $k \times v$-matrix consisting of the most significant v-bits of the k consecutive

Table 1 Parameters s_d for LFSRs of length $P = 2^d - 1$.

d	s_d	d	s_d	d	s_d	d	s_d
10	115	16	283	22	1,336	28	2,573
11	291	17	514	23	1,236	29	2,633
12	172	18	698	24	1,511	30	2,423
13	267	19	706	25	1,445	31	3,573
14	332	20	1,304	26	1,906	32	3,632
15	388	21	920	27	1875		

Table 2 Primitive polynomials f_d for LFSRs of length $P = 2^d - 1$. The lead monomials are t^d.

$t^{10} + t^3 + 1$	$t^{16} + t^5 + t^3 + t^2 + 1$	$t^{22} + t + 1$	$t^{28} + t^3 + 1$
$t^{11} + t^2 + 1$	$t^{17} + t^3 + 1$	$t^{23} + t^5 + 1$	$t^{29} + t^2 + 1$
$t^{12} + t^6 + t^4 + t + 1$	$t^{18} + t^7 + 1$	$t^{24} + t^4 + t^3 + t + 1$	$t^{30} + t^6 + t^4 + t + 1$
$t^{13} + t^4 + t^3 + t + 1$	$t^{19} + t^5 + t^2 + t + 1$	$t^{25} + t^3 + 1$	$t^{31} + t^3 + 1$
$t^{14} + t^5 + t^3 + t + 1$	$t^{20} + t^3 + 1$	$t^{26} + t^6 + t^2 + t + 1$	$t^{32} + t^7 + t^6 + t^2 + 1$
$t^{15} + t + 1$	$t^{21} + t^2 + 1$	$t^{27} + t^5 + t^2 + t + 1$	

outputs from the state s_i. Thus, the multiset $\Phi_{k,v} \cup \{0\}$ defined by (10) is the image of S by the mapping $o_{k,v}$. The mapping inherits GF(2)-linearity from F and o. Consequently, k-dimensional equidistribution with v-bit accuracy is equivalent to the surjectivity of $o_{k,v}$ (since the inverse image of any element is an affine space of the same dimension), and hence is easy to check for small d such as $d < 100$.

A linear feedback shift register (LFSR) is an example of a GF(2)-linear generator as follows: Let $(a_{d-1}, a_{d-2}, \ldots, a_0) \in GF(2)^d$. Choose the state transition matrix $f : S \to S$ to be $f : (b_0, b_1, \ldots, b_{d-1}) \mapsto (b_1, b_2, \ldots, b_{d-1}, \sum_{i=0}^{d-1} a_i b_i)$. Take o as an identity matrix. Thus, $s_i = (b_i, \ldots, b_{i+d-1})$ and b_i satisfies the linear recurrence

$$b_{i+d} = a_{d-1} b_{i+d-1} + \cdots + a_0 b_i.$$

The characteristic polynomial of f is $t^d + a_{d-1} t^{d-1} + \cdots + a_1 t + a_0$, and f attains the maximal period $2^d - 1$ if and only if the polynomial is primitive.

By modifying such LFSRs, we obtain FE generators as follows. For each $d = 10, 11, \ldots, 32$, we take a primitive polynomial of degree d from a list in [7] and let f_d be the associated transition function as above. Let $F := f_d^s$ for some integer s. Then F has the maximal period $2^d - 1$ if and only if s and $2^d - 1$ are coprime. We have a GF(2)-linear generator with transition matrix F and the identity output function o. We search for s in ascending order among the integers coprime to $2^d - 1$ such that the corresponding generator satisfies the FE condition. For each d, we found such s in the range $1 < s < 4{,}000$. We select one s for each d, and call it s_d. See Table 1 for the values we used. We compute $F_d = f_d^{s_d}$ as a $d \times d$ matrix, and then implement the FE $GF(2)$-linear generator with transition function F_d and identity output function. The corresponding polynomials themselves are in Table 2. Although we found a suitable s_d for $10 \le d \le 32$, we have no proof of the existence of s_d for general d.

The FE condition gives stratification over congruent subcubes. Because any rectangle in $[0, 1]^d$ can be closely approximated by subcubes, the d dimensional discrepancy tends to 0 for points formed from an LFSR satisfying the FE condition. Thus an infinite sequence of FE-LFSRs provides a triangular array that is CUD.

4 Antithetic and Round Trip Sampling

Some Markov chains are closely connected to random walks. For example, Metropolis samplers accept or reject proposals made by a random walk process. For a random walk with increments of mean zero, the expected value of X_n is X_0. Similarly, for an autoregressive process such as $X_i = \rho X_{i-1} + \sqrt{1-\rho^2} Z_i$ for Gaussian Z_i, we have $\mathbb{E}(X_n \mid X_0) = X_0$.

We can sample an autoregression by taking

$$X_i = \rho X_{i-1} + \sqrt{1-\rho^2}\Phi^{-1}(u_i) \tag{12}$$

where the driving sequence u_i are IID $U(0, 1)$.

In an *antithetic driving sequence*, we take

$$u_1, u_2, \dots, u_n, 1-u_1, 1-u_2, \dots, 1-u_n.$$

That is, the second half of the sequence simply replays the ones complement of the first half. In a *round trip driving sequence*, we take

$$u_1, u_2, \dots, u_n, 1-u_n, 1-u_{n-1}, \dots, 1-u_1.$$

The sequence steps backwards the way it came.

With either of these driving sequences, an autoregression (12) would satisfy $X_{2n} = X_0 \equiv \mathbb{E}(X_{2n} \mid X_0)$. A random walk would also end where it started. A Markov chain driven by symmetric random walk proposals would be expected to end up close to where it started if most of its proposals were accepted.

Inducing the chain to end up at or near to its expected value should bring a variance reduction. To ensure that the points asymptotically cover the space properly, we require the driving sequence to be (W)CUD. The sampling methods we use are similar to antithetic sampling. The antithetic sampling here differs from that of [6] who sample two chains. A related method in [3] also runs two chains, the second time-reversed one driven by $u_n, \dots, u_1$. The second half of the round trip sequence is time reversed and antithetic to the first half.

When the updates to the Markov chain consume $m > 1$ uniform numbers each, we may write the input to the i'th step as the tuple $u_i^{(m)} = (u_{(m-1)i+1}, \dots, u_{mi}) \in (0, 1)^m$ for $i = 1, \dots, \lfloor n/m \rfloor$. Then a reasonable variant of antithetic and round-trip sampling methods is to use the $2\lfloor n/m \rfloor$ tuples

$$u_1^{(m)}, u_2^{(m)}, \dots, u_{\lfloor n/m \rfloor}^{(m)}, 1 - u_1^{(m)}, 1 - u_2^{(m)}, \dots, 1 - u_{\lfloor n/m \rfloor}^{(m)}, \quad \text{or,}$$
$$u_1^{(m)}, u_2^{(m)}, \dots, u_{\lfloor n/m \rfloor}^{(m)}, 1 - u_{\lfloor n/m \rfloor}^{(m)}, 1 - u_{\lfloor n/m \rfloor - 1}^{(m)}, \dots, 1 - u_1^{(m)}$$

in the simulation, in the orders given above. The corresponding driving sequences in [0, 1] are of length $2m\lfloor n/m \rfloor$, formed by concatenating these m-tuples. We call them m-fold antithetic and m-fold round trip driving sequences, respectively. The subtraction in $1 - u_i^{(m)}$ is interpreted componentwise. When an m-fold method is used, we update the Markov chain $2\lfloor n/m \rfloor$ times using

$$\widetilde{u}_i^{(m)} = \begin{cases} u_i^{(m)} & 1 \le i \le n \\ 1 - u_{i-n}^{(m)} & n < i \le 2n \end{cases}$$

for m-fold antithetic sampling or

$$\widehat{u}_i^{(m)} = \begin{cases} u_i^{(m)} & 1 \le i \le n \\ 1 - u_{2n-i+1}^{(m)} & n < i \le 2n \end{cases}$$

for round trip sampling.

For round trip and antithetic sequences, we will use some results about discrepancies. If $v_1, \dots, v_n$ and $w_1, \dots, w_n$ are points in $[0, 1]^d$ then

$$D_{2n}^{*d}(v_1, \dots, v_n, w_1, \dots, w_n) \le \frac{1}{2}\big(D_n^{*d}(v_1, \dots, v_n) + D_n^{*d}(w_1, \dots, w_n)\big), \tag{13}$$

$$D_n^{*d}(1 - v_1, \dots, 1 - v_n) \le 2^d D_n^{*d}(v_1, \dots, v_n), \quad \text{and} \tag{14}$$

$$\Big| D_{n+k}^{*d}(v_1, \dots, v_{n+k}) - D_n^{*d}(v_1, \dots, v_n) \Big| \le \frac{k}{n+k}. \tag{15}$$

Equation (13) is simple to prove, (14) follows from the well known bound relating discrepancy to star discrepancy and (15) is Lemma 4.2.2 of [17].

For m-fold versions we need another result. In the case $m = 3$ the second half of the driving sequence has entries

$$1-u_3, 1-u_2, 1-u_1,\ \ 1-u_6, 1-u_5, 1-u_4,\ \cdots,\ \ 1-u_{\lfloor n/m \rfloor}, 1-u_{\lfloor n/m \rfloor - 1}, 1-u_{\lfloor n/m \rfloor - 2}.$$

In addition to the one's complement operation, we have reversed the sequence order in blocks of m but preserved order within each block. The $\lfloor n/m \rfloor$ entries are grouped into blocks of size m and a fixed permutation (here a simple reversal) is applied Lemma within each such block. If u_i are CUD then so are the block permuted points. The reasoning is as follows. Consider integers d that are multiples of m. The discrepancy of (nonoverlapping) points $u_i^{(d)}$ is preserved by the permutation. Therefore it vanishes for all such d. Because there are infinitely many such d, the permuted points are CUD by Theorem 3 of [13].

Theorem 1. *Suppose that $u_{n,1},\dots,u_{n,n}$ are from a triangular array that is CUD or weakly CUD. Then the points of an antithetic sequence or a round trip sequence in either original or m-fold versions are CUD (respectively, weakly CUD).*

Proof. First consider the antithetic construction. Pick any integer $d \geq 1$ and let $u_{n,n+j} = 1 - u_{n,j}$ for $n \geq d$ and $j = 1,\dots,n$. Then using u_j for $u_{n,j}$,

$$\begin{aligned}
&D^{*d}_{2n-d+1}(\bar{u}^{(d)}_1,\dots,\bar{u}^{(d)}_{2n-d+1})\\
&\leq D^{*d}_{2n-2d+2}(\bar{u}^{(d)}_1,\dots,\bar{u}^{(d)}_{n-d+1},\bar{u}^{(d)}_{n+1},\dots,\bar{u}^{(d)}_{2n-d+1}) + \frac{d-1}{2n-d+1}\\
&= D^{*d}_{2n-2d+2}(\bar{u}^{(d)}_1,\dots,\bar{u}^{(d)}_{n-d+1},1-\bar{u}^{(d)}_1,\dots,1-\bar{u}^{(d)}_{n-d+1}) + \frac{d-1}{2n-d+1}\\
&\leq \frac{2^d+1}{2} D^{*d}_{n-d+1}(\bar{u}^{(d)}_1,\dots,\bar{u}^{(d)}_{n-d+1}) + \frac{d-1}{2n-d+1}\\
&\to 0,
\end{aligned}$$

using (15) at the first inequality and (13) and (14) at the second. The proof for the round trip construction is similar. For the m-fold versions, we apply Theorem 3 of [13] as described above, to show that the second half of the sequence is CUD. □

5 Empirical Results

We tried four methods on each of four problems. The methods used are IID, CUD, ANT and RND. In these, the driving sequences are IID, CUD based on the construction from Section 3, CUD with antithetics, and CUD with round trip sampling, respectively.

The four problems we tried were: bivariate Gaussian Gibbs sampling using various correlations and tracking the estimated mean, the same but tracking the estimated correlation, a Garch model, and a stochastic volatility model. We label these GMU, GRHO, GARCH and SV respectively.

What we report are root mean square errors based on 100 independent replications generated by Cranley-Patterson rotations. In the Gaussian-Gibbs problem we used twofold versions of ANT and RND. For GARCH and SV we used ordinary (onefold) ANT and RND.

The bivariate Gaussian Gibbs sampler is a simple test case for algorithms. It has $X_i \in \mathbb{R}^2$. The sampling proceeds via

$$X_{i,1} = \rho X_{i-1,2} + \sqrt{1-\rho^2}\,\Phi^{-1}(u_{2i-1}), \quad \text{and} \tag{16}$$

$$X_{i,2} = \rho X_{i,1} + \sqrt{1-\rho^2}\,\Phi^{-1}(u_{2i}), \tag{17}$$

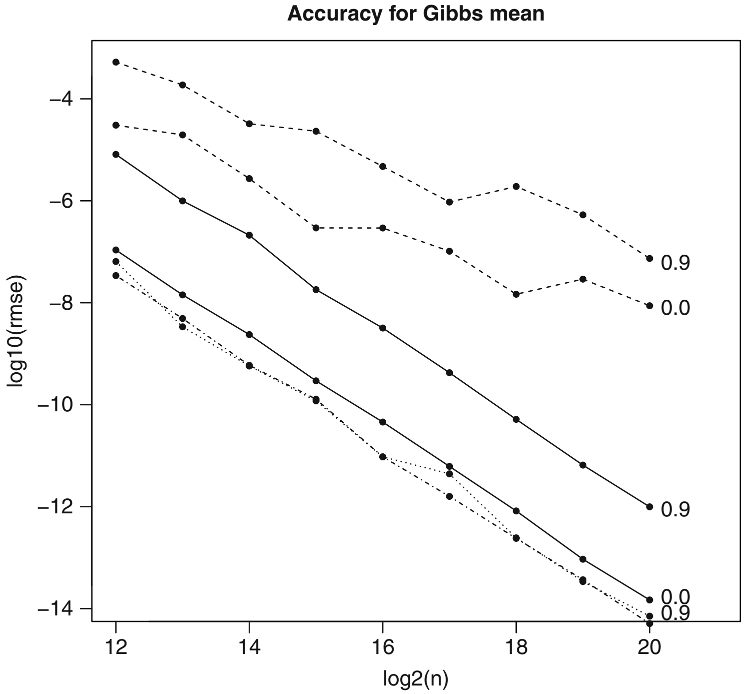


Fig. 1 Numerical results for bivariate Gaussian Gibbs sampling. CUD = *solid* and IID = *dashed*. The goal is to estimate the mean. The correlation is marked at the *right*. For $\rho = 0$ the ANT and RND methods had no error due to symmetry. For $\rho = 0.9$ they were essentially equal and much better than CUD, lying below even the CUD $\rho = 0$ curve. For $\rho = 0.9$, ANT is shown in *dotted lines* and RND in *dot-dash lines*.

starting with $X_0 = (0,0)^T$. We then use $2n$ driving variables to generate $X_1, \dots, X_n$. We varied the true correlation ρ over the range from -0.9 to 0.9.

For problem GMU, we studied estimation of $\mathbb{E}(X_{1,1})$. This is somewhat of a toy problem. In the case $\rho = 0$, the round trip and antithetic sampling algorithms got the answer exactly. The CUD method seemed to attain a better rate than did IID sampling. For $\rho = 0.9$, we also saw an apparently better rate for CUD than IID, while the ANT and RND methods seem to have a better constant than the CUD method. See Fig. 1.

The mean under Gibbs sampling is much easier than most problems we will face. To make it a bit more difficult we considered estimating the correlation itself from the data. This GRHO problem is artificial because we have to know that correlation in order to do the sampling. But a badly mixing chain would not allow us to properly estimate the correlation and so this is a reasonable test. In IID sampling the closer $|\rho|$ is to 1, the easier ρ is to estimate. In Gibbs sampling large $|\rho|$ makes the data values more dependent, but we will see $\rho = 0.9$ is still easier than $\rho = 0$.

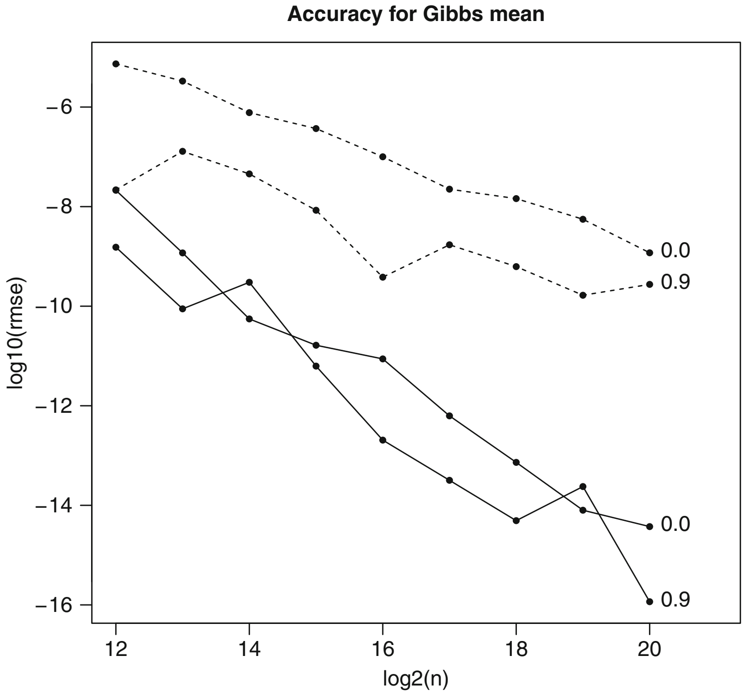


Fig. 2 Numerical results for bivariate Gaussian Gibbs sampling. CUD = *solid* and IID = *dashed*. The goal is to estimate the correlation, which is marked at the *right*. There was little difference between CUD and its balanced alternatives ANT and RND (not shown).

We found that CUD outperformed IID on this case. The ANT and RND methods did about the same as CUD for most correlations but seemed to be worse than CUD for the most extreme values ±0.9. The results comparing CUD to IID are shown in Fig. 2.

The next two models are more challenging. They are stochastic volatility and Garch models. We apply them to a European call option. Under geometric Brownian motion that problem requires one dimensional quadrature and has a closed form solution due to Black and Scholes. For these models the value is a higher dimensional integral.

The SV model we used, from Zhu [19], is generated as follows:

$$\mathrm{d}S = rs\,\mathrm{d}t + \sqrt{V}S\,\mathrm{d}W_1, \quad 0 < t < T \tag{18}$$

$$\mathrm{d}V = \kappa(\theta - V)\,\mathrm{d}t + \sigma\sqrt{V}\,\mathrm{d}W_2, \tag{19}$$

for parameters $T = 6$ (years), $r = 0.04$, $\theta = 0.04$, $\kappa = 2$ and $\sigma = 0.3$. The initial conditions were $S(0) = 100$ and $V(0) = 0.025$. The processes W_1 and W_2 to the price and volatility were correlated Brownian motions with $\rho(\mathrm{d}W_1, \mathrm{d}W_2) = -0.5$.

Table 3 Log (base 10) of root mean squared error in the Heston stochastic volatility model for the four sampling methods and sample sizes 2^{11} to 2^{17}.

$\log_2(n)$	IID	CUD	ANT	RND
11	0.287	−0.089	−0.511	−0.545
12	−0.137	−0.534	−0.311	−0.327
13	0.112	−0.697	−1.017	−0.973
14	−0.594	−0.954	−1.013	−1.085
15	−0.611	−1.245	−1.099	−1.118
16	−1.150	−1.704	−1.770	−1.749
17	−0.643	−1.760	−1.892	−1.927

We priced a European call option, the discounted value of $\mathbb{E}((S(T) - K)_+)$ where the strike price K was 100. That is, the option starts at the money. Each sample path was generated by discretizing time into 2^8 equispaced intervals. It required requiring 2^9 elements u_i to generate both of the required Brownian motions. The results are in Table 3.

The GARCH(1, 1) model we used had

$$\log\Big(\frac{X_t}{X_{t-1}}\Big) = r + \lambda\sqrt{h_t} - \frac{1}{2}h_t + \varepsilon_t, \quad 1 \le t \le T, \quad \text{where} \tag{20}$$

$$\varepsilon_t \sim N(0, h_t), \quad \text{and} \tag{21}$$

$$h_t = \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \beta_1 h_{t-1}. \tag{22}$$

The parameter values, from Duan [5] were $r = 0$, $\lambda = 7.452 \times 10^{-3}$, $T = 30$, $\alpha_0 = 1.525 \times 10^{-5}$, $\alpha_1 = 0.1883$ and $\beta_1 = 0.7162$. The process starts with $h = 0.64\sigma^2$ where $\sigma^2 = 0.2413$ is the stationary variance of X_t.

Once again, the quantity we simulated was the value of a European call option. The strike price was $K = 1$. We started the process at values of $X_0 \in \{0.8, 0.9, 1.0, 1.2\}$.

In this example there was little difference between CUD sampling and either ANT or RND. Plain CUD sampling did better at sample sizes $2^{11} \le n \le 2^{18}$. It seemed to do slightly worse at sample sizes 2^{19} and 2^{20}. The CUD points outperformed IID sampling by a large margin and because the Garch model is interesting and important we show that result in Fig. 3.

6 Conclusions

We have presented some new LFSRs and seen that they yield improved Markov chain quasi-Monte Carlo algorithms on some problems. Other problems do not show much improvement with the introduction of QMC ideas. This pattern is already familiar in finite dimensional applications.

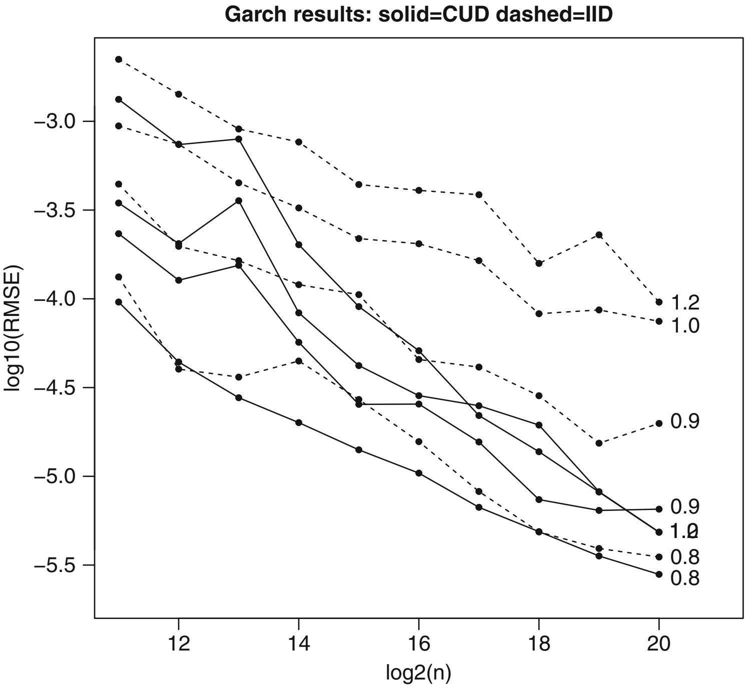


Fig. 3 Numerical results for the Garch(1, 1) model described in the text. The initial price is marked on each trajectory, with the CUD trajectories for $X_0 = 0.9$ and 1.0 getting overlapping labels.

We have also developed some ways to construct new (W)CUD sequences from old ones. The new sequences have a reflection property that we find is sometimes helpful and sometimes not, just as antithetic sampling is sometimes helpful and sometimes not in IID sampling.

The (W)CUD constructions sometimes appear to be achieving a better convergence rate than the IID ones do. There is therefore a need for a theoretical understanding of these rates of convergence.

Acknowledgements This work was supported by grant DMS-0906056 from the U.S. National Science Foundation and by JSPS Grant-in-Aid for Scientific Research No.19204002, No.21654017, No.23244002 and JSPS Core-to-Core Program No.18005. We thank the organizers of MCQMC 2010, Leszek Pleskota and Henryk Woźniakowski, for providing an excellent scientific venue.

References

1. Chen, S., Dick, J., Owen, A.B.: Consistency of Markov chain quasi-Monte Carlo on continuous state spaces. **39**, 673–701 (2011).
2. Chentsov, N.N.: Pseudorandom numbers for modelling Markov chains. Computational Mathematics and Mathematical Physics **7**, 218–233 (1967)
3. Craiu, R.V., Meng, X.L.: Multi-process parallel antithetic coupling for backward and forward Markov chain Monte Carlo. Annals of Statistics **33**, 661–697 (2005)
4. Cranley, R., Patterson, T.: Randomization of number theoretic methods for multiple integration. SIAM Journal of Numerical Analysis **13**, 904–914 (1976)
5. Duan, J.C.: The garch option pricing model. Mathematical Finance **5**(1), 13–32 (1995)
6. Frigessi, A., Gäsemyr, J., Rue, H.H.: Antithetic coupling of two Gibbs sampler chains. Annals of Statistics **28**, 1128–1149 (2000)
7. Hansen, T., Mullen, G.L.: Primitive polynomials over finite fields. Mathematics of Computation **59**, 639–643, S47–S50 (1992)
8. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97–109 (1970)
9. L'Ecuyer, P.: Tables of maximallychenmatsumotonishimuraowen-equidistributed combined lfsr generators. Mathematics of Computation **68**, 261–269 (1999)
10. Levin, M.B.: Discrepancy estimates of completely uniformly distributed and pseudo-random number sequences. International Mathematics Research Notices pp. 1231–1251 (1999)
11. Liao, L.G.: Variance reduction in Gibbs sampler using quasi random numbers. Journal of Computational and Graphical Statistics **7**, 253–266 (1998)
12. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer, New York (2001)
13. Owen, A.B., Tribble, S.D.: A quasi-Monte Carlo Metropolis algorithm. Proceedings of the National Academy of Sciences **102**(25), 8844–8849 (2005)
14. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 2nd edn. Springer, New York (2004)
15. Sobol', I.M.: Pseudo-random numbers for constructing discrete Markov chains by the Monte Carlo method. USSR Computational Mathematics and Mathematical Physics **14**(1), 36–45 (1974)
16. Tootill, J.P.R., Robinson, W.D., Eagle, D.J.: An asymptotically random Tausworthe sequence. Journal of the ACM **20**(3), 469–481 (1973)
17. Tribble, S.D.: Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences. Ph.D. thesis, Stanford University (2007)
18. Tribble, S.D., Owen, A.B.: Construction of weakly CUD sequences for MCMC sampling. Electronic Journal of Statistics **2**, 634–660 (2008)
19. Zhu, J.: A simple and exact simulation approach to Heston model. Tech. rep., Lucht Probst Associates (2008)

Average Case Approximation: Convergence and Tractability of Gaussian Kernels

G.E. Fasshauer, F.J. Hickernell, and H. Woźniakowski

Abstract We study the problem of approximating functions of d variables in the average case setting for a separable Banach space $\mathscr{F}_d$ equipped with a zero-mean Gaussian measure. The covariance kernel of this Gaussian measure takes the form of a Gaussian that depends on shape parameters γ_ℓ. We stress that d can be arbitrarily large. Our approximation error is defined in the $\mathscr{L}_2$ norm, and we study the minimal average case error $e_d^{\mathrm{avg}}(n)$ of algorithms that use at most n linear functionals or function values. For $\gamma_\ell = \ell^{-\alpha}$ with $\alpha \geq 0$, we prove that $e_d^{\mathrm{avg}}(n)$ has a polynomial bound of roughly order $n^{-(\alpha-1/2)}$ independent of d iff $\alpha > 1/2$. This property is equivalent to strong polynomial tractability and says that the minimal number of linear functionals or function values needed to achieve an average case error ε has a bound independent of d proportional roughly to $\varepsilon^{-1/(\alpha-1/2)}$. In the case of algorithms that use only function values the proof is non-constructive. In order to compare the average case with the worst case studied in our earlier paper we specialize the function space $\mathscr{F}_d$ to a reproducing kernel Hilbert space whose kernel is a Gaussian kernel with shape parameters $\gamma_\ell^{\mathrm{rep}}$. To allow for a fair comparison we further equip this space with a zero-mean Gaussian measure whose covariance operator has eigenvalues that depend on a positive parameter q. We prove that the average cases for the whole space and for the unit ball of $\mathscr{F}_d$ are roughly the same provided the $\gamma_\ell^{\mathrm{rep}}$ decay quickly enough. Furthermore, for a particular choice of q

G.E. Fasshauer · F.J. Hickernell
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA
e-mail: fasshauer@iit.edu; hickernell@iit.edu

H. Woźniakowski
Department of Computer Science, Columbia University, New York, NY, USA

Institute of Applied Mathematics, University of Warsaw, ul. Banacha 2, 02-097 Warszawa, Poland
e-mail: henryk@cs.columbia.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_16, © Springer-Verlag Berlin Heidelberg 2012

the dimension-independent convergence for the worst and average case settings are essentially the same.

1 Introduction

Function approximation based on a symmetric, positive definite kernel is popular in practice. One may encounter these methods under various names, including (smoothing) splines [19], kriging [17], radial basis function methods [1], scattered data approximation [24], Gaussian process modeling [12], meshfree methods [2], and surrogate modeling [5]. Because of their popularity and practical success, it is important to understand the accuracy of such methods.

Wendland [24] provides error bounds, but without careful attention to their dependence on the number of input or independent variables, d. Taking that point of view is acceptable as long as one is only concerned with solving problems formulated in our low-dimensional ($d \le 3$) physical world. However, many applications of kernel methods such as problems in finance, statistical learning or computer experiments take place in much higher-dimensional spaces. In the past two decades there has been a broad and deep investigation of the *tractability* of multivariate numerical problems, i.e., determining whether the errors for the best algorithms increase slower than exponentially in d. The volumes of [9, 10] summarize this extensive effort.

The kernels employed in practice, $K_d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, are often assumed to depend on the difference of their arguments or even on the norm of the difference of their arguments:

$$K_d(\boldsymbol{x},\boldsymbol{t}) = \widetilde{K}_d(\boldsymbol{x}-\boldsymbol{t}), \quad \text{stationary or translation invariant,}$$

$$K_d(\boldsymbol{x},\boldsymbol{t}) = \kappa(\|\boldsymbol{x}-\boldsymbol{t}\|_2), \quad \text{isotropic or radially symmetric.}$$

There are few tractability results for these kinds of kernels, which motivates us to study a popular choice of kernel that is amenable to analysis:

$$K_d(\boldsymbol{x},\boldsymbol{t}) = \exp\left(-\gamma_1^2(x_1-t_1)^2 - \cdots - \gamma_d^2(x_d-t_d)^2\right) \quad \text{for all} \quad \boldsymbol{x},\boldsymbol{t} \in \mathbb{R}^d. \tag{1}$$

This Gaussian kernel is used, for example, in the JMP software Gaussian Process Modeling module [15]. Here, $\boldsymbol{\gamma} = \{\gamma_\ell\}_{\ell\in\mathbb{N}}$ is a sequence of positive weights that are called the *shape parameters*. The isotropic (or radial) case corresponds to constant shape parameters, $\gamma_\ell = \gamma > 0$ for all ℓ, whereas the anisotropic case corresponds to varying shape parameters γ_ℓ. We study general γ_ℓ, however, it will be easier to explain the results for specific shape parameters given by $\gamma_\ell = \ell^{-\alpha}$ for $\alpha \ge 0$.

The functions to be approximated are assumed to lie in a Banach or Hilbert space, $\mathscr{F}_d$, which is assumed to be continuously embedded in the space $\mathscr{L}_2 = \mathscr{L}_2(\mathbb{R}^d, \rho_d)$ of square Lebesgue integrable functions, where ρ_d is the probability density function

$$\rho_d(\boldsymbol{t}) = \frac{1}{\pi^{d/2}} \exp\left(-(t_1^2 + t_2^2 + \cdots + t_d^2)\right) \quad \text{for all} \quad \boldsymbol{t} \in \mathbb{R}^d, \quad \int_{\mathbb{R}^d} \rho_d(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t} = 1. \tag{2}$$

The weighted $\mathscr{L}_2$ inner product is defined by

$$\langle f, g \rangle_{\mathscr{L}_2} = \int_{\mathbb{R}^d} f(\boldsymbol{t}) g(\boldsymbol{t})\, \rho_d(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t}.$$

The approximation error for a function, f, approximated by an algorithm, $A : \mathscr{F}_d \to \mathbb{R}$, is defined as $\| I_d f - A f \|_{\mathscr{L}_2}$, where $I_d : \mathscr{F}_d \to \mathscr{L}_2$ is the embedding operator defined by $I_d f = f$. This choice of the weight ρ_d reflects a requirement for greater approximation accuracy near the origin.

The *worst case error* of an algorithm is defined by its worst behavior over the unit ball in $\mathscr{F}_d$. Worst case convergence and tractability of approximation for functions in the Hilbert space with reproducing kernel (1) has recently been studied in [3]. We will later summarize the results obtained in that paper.

An alternative to the worst case setting is the *average case setting*, where functions to be approximated are assumed to be realizations of a stochastic (often Gaussian) process. In addition to the aforementioned monographs, [9, 10], the work of [6, 8, 13, 14, 21], and [22, 23] addresses average case convergence and tractability. The purpose of this article is to investigate the average case error convergence and tractability for function approximation (or recovery) defined over a separable Banach space, $\mathscr{F}_d$, of real-valued functions equipped with a zero-mean Gaussian measure, μ_d, with a covariance kernel K_d, i.e.,

$$\int_{\mathscr{F}_d} L(f)\, \mu_d(\mathrm{d}f) = 0 \quad \text{for all} \quad L \in \mathscr{F}_d^*, \quad K_d(\boldsymbol{x}, \boldsymbol{t}) = \int_{\mathscr{F}_d} f(\boldsymbol{x}) f(\boldsymbol{t})\, \mu_d(\mathrm{d}f). \tag{3}$$

Although we need to present known results for general kernels K_d, our main focus and new results will be presented for the Gaussian covariance kernel (1).

Two types of algorithms, A, are investigated: those that may use function data based on arbitrary continuous linear functionals, the class $\Lambda^{\text{all}} = \mathscr{F}_d^*$, and those that use only function values, the class Λ^{std}. In the average case setting it is seen that the convergence and tractability for the two classes are the same, whereas in the worst case they may be different.

The average case $\mathscr{L}_2$ function approximation error for a measurable algorithm A is defined by

$$e_d^{\text{avg}}(A) := \left(\int_{\mathscr{F}_d} \| I_d f - A f \|_{\mathscr{L}_2}^2\, \mu_d(\mathrm{d}f) \right)^{1/2},$$

It is important to know how small the average case error can be when $A(f)$ is based on n pieces of function information generated by $L_i \in \Lambda$. Let the minimal average case error of all algorithms that use at most n linear functionals be denoted by

$$e_d^{\text{avg}}(n;\Lambda) = \inf_{\substack{\{L_i\}_{i=1}^n \\ L_i \in \Lambda}} \inf_{\substack{A \text{ based} \\ \text{on } \{L_i\}_{i=1}^n}} e_d^{\text{avg}}(A), \qquad \Lambda \in \{\Lambda^{\text{all}}, \Lambda^{\text{std}}\}. \tag{4}$$

We define a covariance kernel such that the initial error is $e_d^{\text{avg}}(0,\Lambda) = 1$. The aim is to find out how quickly $e_d^{\text{avg}}(n;\Lambda)$ decays with n.

In the case of the Gaussian covariance kernel, (1), and the Gaussian weight, (2), we will show that for an arbitrary (large) positive p there exists $C_{d,p}$ depending on d and p such that

$$e_d^{\text{avg}}(n;\Lambda) \le C_{d,p} n^{-p} \quad \text{for all} \quad n \in \mathbb{N}. \tag{5}$$

This means that the convergence of the $\mathscr{L}_2$ average case approximation error is as fast as *any* polynomial in n^{-1}. Unfortunately, the dimension dependence of the leading factor $C_{d,p}$ might prove to be disastrous. We define a *dimension-independent* convergence exponent as

$$p_{\text{cnv}}(\Lambda) = \sup\big\{p > 0 : \sup_{d,n \in \mathbb{N}} n^p e_d^{\text{avg}}(n;\Lambda) < \infty\big\}. \tag{6}$$

The supremum of the empty set is taken to be zero. This means that $e_d^{\text{avg}}(n;\Lambda) \le C_p n^{-p}$ for all $p < p_{\text{cnv}}$, but perhaps not for $p = p_{\text{cnv}}$. We say that *dimension-independent convergence* holds iff $p_{\text{cnv}}(\Lambda) > 0$. We want to find conditions on the shape parameters γ_ℓ that bound p_{cnv} above and below.

This notion is equivalent to *strong polynomial tractability*, which says that the minimal number, $n^{\text{avg}}(\varepsilon; d, \Lambda)$, of linear functionals or function values needed to achieve an average case error ε can be bounded by $M_\tau\, \varepsilon^{-\tau}$, for some positive M_τ and τ, and this holds for all d. The *exponent of strong polynomial tractability* is defined as

$$p_{\text{str}}(\Lambda) = \inf\big\{p \ge 0 : \sup_{d \in \mathbb{N},\ \varepsilon \in (0,1)} \varepsilon^p n^{\text{avg}}(\varepsilon; d, \Lambda) < \infty\big\} = \frac{1}{p_{\text{cnv}}(\Lambda)}. \tag{7}$$

The infimum of the empty set is taken to be infinity.

The main result of this paper for $\gamma_\ell = \ell^{-\alpha}$ is that dimension-independent convergence and strong tractability hold iff $\alpha > 1/2$, and then the exponents for the two classes Λ^{all} and Λ^{std} are the same and equal to

$$p_{\text{cnv}} = p_{\text{str}}^{-1} = \alpha - 1/2.$$

It is worth noting that for the isotropic case, $\gamma_\ell = \gamma$, we have $\alpha = 0$. Therefore there is *no* dimension-independent convergence and *no* strong polynomial tractability.

Even a modest decay of $\gamma_\ell = \ell^{-\alpha}$, $0 < \alpha \le 1/2$, is insufficient to yield dimension-independent convergence and strong polynomial tractability.

In fact, for constant shape parameters, $\gamma_\ell = \gamma > 0$, the number of data based on linear functionals, $n^{\mathrm{avg}}(\varepsilon; d, \Lambda^{\mathrm{all}})$, required to guarantee an error no larger than ε in the average case setting is bounded below as follows:

$$n^{\mathrm{avg}}(\varepsilon; d, \Lambda^{\mathrm{all}}) \ge (1-\varepsilon^2)\left(1 + \frac{2\gamma^2}{1+\sqrt{1+4\gamma^2}}\right)^d \quad \text{for all} \quad \varepsilon \in (0,1),\ d \in \natural. \qquad (8)$$

Hence, the minimal number of linear functionals is *exponential* in d, and this is called the *curse of dimensionality*. For $\gamma = 1$, the lower bound above is $(1 - \varepsilon^2)(1.618033\ldots)^d$, whereas for $\gamma = 0.1$, it is $(1-\varepsilon^2)(1.009901\ldots)^d$. This means that for small positive γ, the curse of dimensionality is delayed. However, in the Λ^{std} case, small values of γ give rise to ill-conditioned Gram matrices K given in (12). The recent work of [4] uses the same eigen-decomposition (13) of the Gaussian kernel employed here to avoid forming the matrix K and to compute Gaussian kernel approximants in a numerically stable manner with small γ. Furthermore, it is known that in the "flat" limit, $\gamma \to 0$, isotropic Gaussian kernel interpolants converge to a polynomial interpolant, and thus isotropic Gaussian interpolation generalizes multivariate polynomial interpolation [16].

We now comment on the constructiveness of our convergence results. For the class Λ^{all}, optimal error algorithms in the average case are known. Hence, for $\alpha > 1/2$ we know how to construct algorithms for which we can achieve dimension-independent convergence and strong polynomial tractability with the exponents p_{cnv} and p_{str}. For the class Λ^{std}, we use the result from [7] that states the equality of the exponents for the classes Λ^{all} and Λ^{std} for a much more general case. Unfortunately, this result is *non-constructive* and we only know the existence of such algorithms. It would be of a practical interest to find an explicit construction of such algorithms.

In the final section of this paper we compare the worst case and average case results for the function space $\mathscr{F}_d$ studied in [3]. This is the reproducing kernel Hilbert space whose kernel is a Gaussian kernel with γ_ℓ replaced by a possibly different sequence of shape parameters $\gamma_\ell^{\mathrm{rep}}$. For simplicity of presentation, we assume that $\gamma_\ell^{\mathrm{rep}} = \ell^{-\beta}$ for some $\beta \ge 0$. It turns out that dimension-independent convergence and strong polynomial tractability, both in the worst case setting, hold for the class Λ^{all} iff $\beta > 0$, and for the class Λ^{std} if $\beta > 1/2$. For the class Λ^{all}, the exponents are

$$p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} = \beta,$$

whereas for the class Λ^{std}, we only have estimates of p_{cnv}, namely,

$$\frac{\beta}{1 + \frac{1}{2\beta}} \le p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} \le \beta.$$

To allow for a fair comparison between the worst case and average case results we equip the function space $\mathscr{F}_d$ with a zero-mean Gaussian measure with a covariance kernel K_d that depends on a positive parameter q, but is *not of the form* (1). For $\beta > 1/2$ and $q > 1/(2\beta)$, we show that dimension-independent convergence and strong polynomial tractability, both in the average case setting, hold and the exponents for the classes Λ^{all} and Λ^{std} are the same and equal to

$$p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} = (q+1)\beta - 1/2 > \beta.$$

For $q \approx 1/(2\beta)$, the exponents in the worst and average case settings for the class Λ^{all} are almost the same. In fact, we present conditions when they are the same. For instance, this holds for $\gamma_\ell^{\mathrm{rep}} = (\ell \ln^2(\ell+1))^{-\beta}$.

It is interesting that for some cases we have the same exponents in the worst case and average case settings. We stress that this holds for the space of functions that are analytic and for the exponents that are independent of d. If one wants to find the exponents for a fixed d and is ready to accept factors in the error bounds that may arbitrarily depend on d, our analysis does not apply.

We finish the introduction by indicating a number of open problems. We have already mentioned one problem concerning a construction of an algorithm that uses only function values and achieves the dimension-independent convergence exponent. Another problem would be to address different types of tractability. We restrict ourselves in this paper to strong polynomial tractability. It would be of interest to extend the analysis to polynomial, quasi-polynomial, T-tractability as well as to weak tractability, see [9, 10] for the definition of these tractability notions and survey of the results. Finally, it would be of interest to study the approximation for the function spaces with error measured not necessarily in the $\mathscr{L}_2$ norm as done here. The case of the $\mathscr{L}_\infty$ norm seems especially challenging.

2 Assumptions and Background

The problem formulation and results that are outlined in the introduction require some assumptions that are stated here for clarity and completeness. Moreover, some known results on the convergence and tractability of function approximation are reviewed.

It is assumed that function evaluation at any point is a continuous linear functional on $\mathscr{F}_d$, the Banach space of functions to be approximated. This implies the existence of the covariance kernel, $K_d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, defined above in (3). The kernel K_d is symmetric and positive semi-definite, i.e.,

$$K_d(\boldsymbol{x},\boldsymbol{t}) = K_d(\boldsymbol{t},\boldsymbol{x}) \quad \forall \boldsymbol{x},\boldsymbol{t} \in \mathbb{R}^d, \tag{9a}$$

$$\sum_{i,j=1}^{n} K_d(\boldsymbol{x}_i,\boldsymbol{x}_j)c_i c_j \geq 0 \quad \forall n \in \mathbb{N},\ \boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \in \mathbb{R}^d,\ c_1,\ldots,c_n \in \mathbb{R}. \tag{9b}$$

We assume that $K_d(\boldsymbol{x},\cdot)$, $K_d(\cdot,\boldsymbol{t})$ as well as $K_d(\cdot,\cdot)$ are in $\mathscr{L}_2$ for all $\boldsymbol{x},\boldsymbol{t}\in\mathbb{R}^d$. In addition, we assume for convenience that $\mathscr{F}_d$ and μ_d in (3) are chosen such that

$$\int_{\mathbb{R}^d} K_d(\boldsymbol{x},\boldsymbol{x})\,\rho_d(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1. \tag{10}$$

This guarantees that the minimal average case error using no function information is unity. Assumption (10) of course holds for the Gaussian covariance kernel defined in (1), as well as for any stationary kernel where $\widetilde{K}(\boldsymbol{x},\boldsymbol{x}) = 1$. Since $K_d(\boldsymbol{x},\boldsymbol{t}) \le \sqrt{K_d(\boldsymbol{x},\boldsymbol{x})}\,\sqrt{K_d(\boldsymbol{t},\boldsymbol{t})}$ for all $\boldsymbol{x},\boldsymbol{t}\in\mathbb{R}^d$, we have

$$\int_{\mathbb{R}^d} K_d^2(\boldsymbol{x},\boldsymbol{t})\,\rho_d(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \le K_d(\boldsymbol{t},\boldsymbol{t}) \quad\text{and}\quad \int_{\mathbb{R}^{2d}} K_d^2(\boldsymbol{x},\boldsymbol{t})\,\rho_d(\boldsymbol{x})\,\rho_d(\boldsymbol{t})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \le 1.$$

The approximations, Af, considered here use partial information about f, namely, n continuous linear functional evaluations denoted $L_1(f), L_2(f), \ldots, L_n\ (f)$, where the L_i belong to Λ^{all} or Λ^{std}. It is known that nonlinear algorithms and adaptive choice of L_i do not essentially help for the $\mathscr{L}_2$ approximation problem, see [18, 20]. That is why we can restrict our attention to *linear algorithms*, i.e., algorithms of the form

$$Af = \sum_{i=1}^{n} L_i(f)\,g_i, \tag{11}$$

where $g_i \in \mathscr{L}_2$. The number n is called the *cardinality* of A and characterizes the cost of the algorithm A. The case of $n = 0$, i.e., no information about the function is used, leads to the zero algorithm, $Af = 0$.

For a fixed design, $L_1, \ldots, L_n$, one may choose $\boldsymbol{g} = (g_i)_{i=1}^n$ to minimize $e_d^{\mathrm{avg}}(A)$ as follows: $\boldsymbol{g}(\boldsymbol{x}) = \mathsf{K}^{-1}\boldsymbol{z}(\boldsymbol{x})$, where

$$\mathsf{K} := \left(\int_{\mathscr{F}_d} L_i(f)L_j(f)\,\mu_d(\mathrm{d}f)\right)_{i,j=1}^{n}, \quad \boldsymbol{z}(\boldsymbol{x}) := \left(\int_{\mathscr{F}_d} L_i(f)\,f(\boldsymbol{x})\,\mu_d(\mathrm{d}f))\right)_{i=1}^{n}. \tag{12}$$

This is the *spline algorithm*, which was mentioned in the introduction. Note that depending on the choice of L_i the matrix K may be singular. In this case, the solution $\boldsymbol{g}(\boldsymbol{x}) = \mathsf{K}^{-1}\boldsymbol{z}(\boldsymbol{x})$ is well defined as the vector with minimal Euclidean norm that satisfies the equation $\mathsf{K}\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})$, which always has at least one solution. The average case error of the spline algorithm is

$$e_d^{\mathrm{avg}}\left(\{L_i\}_{i=1}^n\right) := \inf_{\substack{A \text{ based}\\ \text{on } \{L_i\}_{i=1}^n}} e_d^{\mathrm{avg}}(A) = \left(1 - \int_{\mathbb{R}^d} \boldsymbol{z}^T(\boldsymbol{x})\mathsf{K}^{-1}\boldsymbol{z}(\boldsymbol{x})\,\rho_d(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right)^{1/2}.$$

We stress that (10) has been exploited to simplify the expression for $e_d^{\mathrm{avg}}(A)$.

When one is able to draw data from the class of all continuous linear functionals, the optimal design or sampling scheme, $\{L_i\}_{i=1}^n$, is known, see [11, 18, 20]. More precisely, consider the probability measure $\nu_d = \mu_d I_d^{-1}$ on $\mathscr{L}_2$. Due to the assumptions on K_d, the measure ν_d is also a zero-mean Gaussian with the covariance operator $C_{\nu_d} : \mathscr{L}_2^* = \mathscr{L}_2 \to \mathscr{L}_2$ given by

$$C_{\nu_d} g = \int_{\mathbb{R}^d} K_d(\boldsymbol{x}, \cdot) g(\boldsymbol{x})\, \rho_d(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \quad \text{for all} \quad g \in \mathscr{L}_2,$$

where K_d is the covariance kernel of μ_d. The operator C_{ν_d} is compact and self-adjoint. The eigenpairs $(\lambda_{d,j}, \varphi_{d,j})$ of C_{ν_d} satisfy the integral equation

$$\int_{\mathbb{R}^d} K_d(\boldsymbol{x}, \boldsymbol{t})\, \varphi_{d,j}(\boldsymbol{t})\, \rho_d(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t} = \lambda_{d,j}\, \varphi_{d,j}(\boldsymbol{x}). \tag{13a}$$

The eigenfunctions $\varphi_{d,j}$ can be chosen to be $\mathscr{L}_2$ orthonormal, and the eigenvalues $\lambda_{d,j}$ are ordered, $\lambda_{d,1} \geq \lambda_{d,2} \geq \cdots$. If only k eigenvalues are positive, then to simplify the notation we formally set $\lambda_{d,j} = 0$ for all $j > k$. Then

$$K_d(\boldsymbol{x}, \boldsymbol{t}) = \sum_{j=1}^{\infty} \lambda_{d,j}\, \varphi_{d,j}(\boldsymbol{x})\, \varphi_{d,j}(\boldsymbol{t}) \quad \text{for all} \quad \boldsymbol{x}, \boldsymbol{t} \in \mathbb{R}^d. \tag{13b}$$

Note that due to (10) we have

$$1 = \int_{\mathbb{R}^d} K_d(\boldsymbol{x}, \boldsymbol{x})\, \rho_d(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \sum_{j=1}^{\infty} \lambda_{d,j}. \tag{13c}$$

The optimal sampling scheme for the class Λ^{all} is to choose $L_i(f) = \langle f, \varphi_{d,i} \rangle_{\mathscr{L}_2}$, for which the linear algorithm which minimizes the average case error corresponds to projecting the function f into the vector space spanned by the eigenfunctions corresponding to the n largest eigenvalues. This algorithm is of the form

$$A_{\text{all}} f = \sum_{i=1}^{n} \langle f, \varphi_{d,i} \rangle_{\mathscr{L}_2}\, \varphi_{d,i}. \tag{14}$$

The square of the average case error of A_{all} is the tail sum of the eigenvalues so that

$$e_d^{\text{avg}}(n; \Lambda^{\text{all}}) = \left(\sum_{j=n+1}^{\infty} \lambda_{d,j} \right)^{1/2}. \tag{15}$$

Before delving into the detailed derivations of $p_{\text{cnv}}(\Lambda)$ and $p_{\text{str}}(\Lambda)$ defined in (6) and (7), respectively, it is important to be clear about what they depend on. They do not depend on n, d, or ε since they are defined in terms of a supremum/infimum over these quantities. The exponents $p_{\mathrm{cnv/str}}$ may depend on the class of available function information, Λ. They also depend on the spaces $\mathscr{F}_d$, the measures μ_d through the covariance kernels K_d and the weights ρ_d. If we choose K_d to be a Gaussian kernel then, as we shall see, $p_{\mathrm{cnv/str}}$ strongly depend on the sequence of shape parameters $\boldsymbol{\gamma} = \{\gamma_\ell\}_{\ell\in\natural}$ appearing in (1). More precisely, they depend on how quickly γ_ℓ decays as $\ell \to \infty$. The decay of $\boldsymbol{\gamma} = \{\gamma_\ell\}_{\ell\in\natural}$ is measured by the rate $r(\boldsymbol{\gamma})$ defined as

$$r(\boldsymbol{\gamma}) = \sup\left\{\beta > 0 \;:\; \sum_{\ell=1}^{\infty} \gamma_\ell^{1/\beta} < \infty\right\} \tag{16}$$

with the convention that the supremum of the empty set is taken to be zero. For example, if $\gamma_\ell = \ell^{-r}$ for $r \geq 0$ then $r(\boldsymbol{\gamma}) = r$.

3 Convergence and Tractability for Λ^{all} and Λ^{std}

We now specify the average case results for the Gaussian kernel given by (1) and the Gaussian function ρ_d given by (2). The eigenpairs of C_{ν_d} are known in this case. More precisely, for the univariate case, $d = 1$, and the covariance kernel

$$K_1(x,t) = e^{-\gamma^2(x-t)^2} \quad \text{for all} \quad x, t \in \mathbb{R},$$

the eigenpairs $\left(\lambda_{1,j}, \varphi_{1,j}\right) = \left(\tilde{\lambda}_{\gamma,j}, \tilde{\varphi}_{\gamma,j}\right)$ are given by

$$\tilde{\lambda}_{\gamma,j} = (1-\omega_\gamma)\,\omega_\gamma^{j-1}, \qquad \text{where} \quad \omega_\gamma = \frac{\gamma^2}{\frac{1}{2}(1+\sqrt{1+4\gamma^2})+\gamma^2}, \tag{17}$$

$$\tilde{\varphi}_{\gamma,j}(x) = \sqrt{\frac{(1+4\gamma^2)^{1/4}}{2^{j-1}(j-1)!}}\exp\left(-\frac{\gamma^2x^2}{\frac{1}{2}(1+\sqrt{1+4\gamma^2})}\right)H_{j-1}\left((1+4\gamma^2)^{1/4}x\right),$$

where H_{j-1} is the Hermite polynomial of degree $j-1$,

$$H_{j-1}(x) = (-1)^{j-1}e^{x^2}\frac{\mathrm{d}^{j-1}}{\mathrm{d}x^{j-1}}e^{-x^2} \quad \text{for all} \quad x \in \mathbb{R},$$

see e.g., [3, 12].

Since the multivariate ($d > 1$) anisotropic Gaussian kernel, (1), is a product of univariate Gaussian kernels, the eigenpairs for the multivariate case are products of those for the univariate case. Specifically, for $d > 1$, let $\boldsymbol{\gamma} = \{\gamma_\ell\}_{\ell\in\natural}$, $\mathbf{j} = (j_1, j_2, \ldots, j_d) \in \natural^d$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$. Then the eigenpairs $\left(\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}, \tilde{\varphi}_{d,\boldsymbol{\gamma},\mathbf{j}}\right)$ are given by the products

$$\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}} = \prod_{\ell=1}^{d} \tilde{\lambda}_{\gamma_\ell,j_\ell} = \prod_{\ell=1}^{d} (1-\omega_{\gamma_\ell})\,\omega_{\gamma_\ell}^{j_\ell-1}, \qquad \tilde{\varphi}_{d,\boldsymbol{\gamma},\mathbf{j}}(\boldsymbol{x}) = \prod_{\ell=1}^{d} \tilde{\varphi}_{\gamma_\ell,j_\ell}(x_\ell). \tag{18}$$

The notations $\left(\tilde{\lambda}_{\gamma,j}, \tilde{\varphi}_{\gamma,j}\right)$ for $d = 1$ and $\left(\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}, \tilde{\varphi}_{d,\boldsymbol{\gamma},\mathbf{j}}\right)$ for $d > 1$ have been introduced to emphasize the dependence of the eigenpairs on $\boldsymbol{\gamma}$. Note that while the eigenvalues $\tilde{\lambda}_{\gamma,j}$ are ordered in decreasing magnitude, the $\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}$ are not. The $\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}$, for which we have an explicit expression in (18), are, however, the same as the *ordered* $\lambda_{d,j}$ referred to above in (13), where the $\boldsymbol{\gamma}$ dependence is hidden.

Since the tail sum of the ordered eigenvalues in (15) is often not directly accessible, we show that it can be related to the sums of the powers of all eigenvalues. Let

$$M_{d,\tau} := \left(\sum_{j=1}^{\infty} \lambda_{d,j}^{1/\tau}\right)^{\tau} \qquad \text{for all} \quad \tau \geq 1. \tag{19}$$

Note that $M_{d,1} = 1$ and by Jensen's inequality $M_{d,\tau} \geq M_{d,1} = 1$. Furthermore,

$$M_{d,\tau} = 1 \ \text{ for } \ \tau > 1 \quad \text{iff} \quad \lambda_{d,1} = 1 \ \text{ and } \ \lambda_{d,j} = 0 \ \text{ for all } \ j \geq 2.$$

This sum of powers of all eigenvalues bounds the tail sum as follows:

$$\sum_{j=n+1}^{\infty} \lambda_{d,j} \leq \sum_{j=n+1}^{\infty} \left(\lambda_{d,j}^{1/\tau}\right)^{\tau} \leq \sum_{j=n+1}^{\infty} \left(\frac{1}{j}\sum_{k=1}^{j} \lambda_{d,k}^{1/\tau}\right)^{\tau}$$
$$\leq M_{d,\tau} \sum_{j=n+1}^{\infty} j^{-\tau} \leq \frac{M_{d,\tau}}{(1-\tau)n^{1-\tau}}$$

Another argument gives bounds in the opposite direction, see [6, Corollary 1] and [7, Lemmas 1 and 2]. Thus, the convergence rate for average case approximation depends on the finiteness of $M_{d,\tau}$, and strong tractability or dimension-independent convergence rates depends on the boundedness of $M_{d,\tau}$ over all d. This is embodied in the following lemma, which is a special case of Theorems 6.1 and 6.2 in [9] for $\Lambda = \Lambda^{\text{all}}$ and utilizes [7] for the case $\Lambda = \Lambda^{\text{std}}$.

Lemma 1. *Consider $\mathscr{L}_2$ function approximation for the class Λ^{all} or Λ^{std} in the average case setting with any symmetric, positive definite covariance kernel, $K_d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfying (9) and having an eigen-decomposition (13). Then $\mathscr{L}_2$ function approximation has a* dimension-dependent *convergence rate of $\mathscr{O}(n^{-p})$ provided that $M_{d,2p+1}$ is finite. There is* dimension-independent *convergence and strong polynomially tractability iff there exists a positive τ such that*

$$\sup_{d\in\mathbb{N}} M_{d,2\tau+1} < \infty.$$

In this case the exponents are

$$p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} = \sup\left\{\tau \in (0,\infty) \ : \ \sup_{d\in\natural} M_{d,2\tau+1} < \infty\right\}. \qquad \square$$

It is easy to see that dimension-independent convergence and strong polynomial tractability do *not* hold for non-trivial product covariance kernels of the form

$$K_d(\boldsymbol{x},\boldsymbol{t}) = \prod_{\ell=1}^{d} K_1(x_\ell,t_\ell) \quad \text{for all} \quad \boldsymbol{x},\boldsymbol{t} \in \mathbb{R}^d,$$

In this case, the eigenvalues $\lambda_{d,j}$ are the products of the eigenvalues, λ_j, of K_1, and

$$M_{d,2\tau+1} = \left(\sum_{j=1}^{\infty} \lambda_j^{1/(2\tau+1)}\right)^{d(2\tau+1)}.$$

Unless, $\lambda_1 = 1$ and $0 = \lambda_2 = \lambda_3 = \cdots$, it follows that $\sum_{j=1}^{\infty} \lambda_j^{1/(2\tau+1)} > 1$ and $M_{d,2\tau+1}$ goes to infinity for all $\tau > 0$. In fact, we can say more and show that the minimal number $n^{\mathrm{avg}}(\varepsilon; d, \Lambda^{\mathrm{all}})$ depends exponentially on d. Indeed, from (15), note that

$$\left(e_d^{\mathrm{avg}}(n;\Lambda^{\mathrm{all}})\right)^2 = \sum_{j=n+1}^{\infty} \lambda_{d,j} = \sum_{j=1}^{\infty}\lambda_{d,j} - \sum_{j=1}^{n}\lambda_{d,j} \geq 1 - n\lambda_{d,1} = 1 - \lambda_1^d.$$

If $\lambda_2 > 0$ then $\sum_{j=1}^{\infty} \lambda_j = 1$ implies that $\lambda_1 < 1$, which then yields

$$n^{\mathrm{avg}}(\varepsilon; d, \Lambda^{\mathrm{all}}) \geq (1-\varepsilon^2)\left(\frac{1}{\lambda_1}\right)^d \quad \text{for all} \quad \varepsilon \in (0,1),\ d \in \natural. \tag{20}$$

This is called the *curse of dimensionality.*

We now specify Lemma 1 for the Gaussian anisotropic kernel, (1). The analysis in [3, Lemma 1] shows that

$$M_{d,\tau} = \left(\sum_{\mathbf{j}\in\natural^d} \tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}^{1/\tau}\right)^{\tau} = \prod_{\ell=1}^{d} \frac{1-\omega_{\gamma_\ell}}{\left(1-\omega_{\gamma_\ell}^{1/\tau}\right)^{\tau}} = \begin{cases} = 1, & \tau = 1, \\ > 1, & 1 < \tau < \infty, \end{cases}$$

where ω_γ was defined above in (17). Noting that $\omega_\gamma = \gamma^2(1+o(1))$ as $\gamma \to 0$, it is further shown that for all $\tau > 1$,

$$\tau < 2r(\boldsymbol{\gamma}) \implies \sup_{d\in\natural} M_{d,\tau} = \prod_{\ell=1}^{\infty} \frac{1-\omega_{\gamma_\ell}}{\left(1-\omega_{\gamma_\ell}^{1/\tau}\right)^{\tau}} < \infty \implies \tau \leq 2r(\boldsymbol{\gamma}), \tag{21}$$

where $r(\boldsymbol{\gamma})$ is defined above in (16). Combining this equation with Lemma 1 determines dimension-independent convergence and strong polynomial tractability as well as their exponents.

Theorem 1. *Consider $\mathscr{L}_2$ approximation for the class Λ^{all} or Λ^{std} in the average case setting with the Gaussian kernel,* (1)*, and shape parameters $\boldsymbol{\gamma} = \{\gamma_\ell\}_{\ell\in\mathbb{N}}$. Then $\mathscr{L}_2$ function approximation has a* dimension-dependent *convergence rate of $\mathscr{O}(n^{-p})$ for all $p > 0$. Moreover, $\mathscr{L}_2$ approximation has dimension-independent convergence exponent and is strongly polynomially tractable iff $r(\boldsymbol{\gamma}) > 1/2$. In this case the exponents are*

$$p_{\text{cnv}} = p_{\text{str}}^{-1} = r(\boldsymbol{\gamma}) - 1/2.$$

For the class Λ^{all}, the algorithm, (14)*, which attains these exponents is given by projecting the function into the first n eigenfunctions of the kernel. For the class Λ^{std}, the algorithm that attains these exponents is not known explicitly.*

The isotropic Gaussian kernel, i.e., constant shape parameters, $\gamma_\ell = \gamma > 0$, has $r(\boldsymbol{\gamma}) = 0$, which implies no dimension-independent convergence rate in the average case setting. Furthermore, we can apply (20), with $1/\lambda_1 = 1/(1-\omega_\gamma)$ to obtain (8).

4 Comparison of the Worst and Average Case Settings

We compare the results in the worst and average case settings for the function space $\mathscr{F}_d = H(K_d^{\text{rep}})$. This is the Hilbert space whose reproducing kernel is

$$K_d^{\text{rep}}(\boldsymbol{x},\boldsymbol{t}) = \exp\left(-[\gamma_1^{\text{rep}}]^2(x_1-t_1)^2 - \cdots - [\gamma_d^{\text{rep}}]^2(x_d-t_d)^2\right) \text{ for all } \boldsymbol{x},\boldsymbol{t}\in\mathbb{R}^d. \tag{22}$$

That is, K_d^{rep} has the same form as the covariance Gaussian kernel, (1), for possibly different shape parameters or coordinate weights $\boldsymbol{\gamma}^{\text{rep}} = \{\gamma_\ell^{\text{rep}}\}_{\ell\in\mathbb{N}}$.

The $\mathscr{L}_2$ approximation problem for this space and the Gaussian function ρ_d given by (2) was studied in [3] for the worst case setting. We now briefly recall some of the results from this paper. For simplicity, we only consider the normalized error criterion for which we want to decrease the initial error by ε. For the class Λ^{all}, we have dimension-independent convergence and strong polynomial tractability iff $r(\boldsymbol{\gamma}^{\text{rep}}) > 0$. If so, then the exponents are

$$p_{\text{cnv}} = p_{\text{str}}^{-1} = r(\boldsymbol{\gamma}^{\text{rep}}).$$

For the class Λ^{std}, the results are less satisfactory because we only have upper and lower bounds on the exponents. We have dimension-independent convergence and strong polynomial tractability if $r(\boldsymbol{\gamma}^{\text{rep}}) > 1/2$ and then

$$\frac{r(\boldsymbol{\gamma}^{\text{rep}})}{1+\frac{1}{2r(\boldsymbol{\gamma}^{\text{rep}})}} \le p_{\text{cnv}} = p_{\text{str}}^{-1} \le r(\boldsymbol{\gamma}^{\text{rep}}).$$

We turn to the average case setting. As already mentioned, to get a fair comparison between the worst and average settings, we must guarantee that the average case setting over the unit ball is roughly the same as for the whole space. We will do this by constructing a covariance kernel with the same eigenfunctions as those for K_d^{rep}, but with different eigenvalues. Since the dimension-independent convergence and tractability depend only on the eigenvalues, the arguments used in the previous sections may then be applied. However, the resulting covariance kernel will no longer be of Gaussian form (1).

Let $(\tilde{\lambda}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}, \tilde{\varphi}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}})$ be the eigenpairs for the reproducing kernel (22) as defined in (18). Then

$$\eta_{d,\mathbf{j}} := \eta_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}} = \sqrt{\tilde{\lambda}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}}\;\tilde{\varphi}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}, \quad \mathbf{j} \in \mathbb{N}^d$$

is the complete orthonormal basis of $\mathscr{F}_d$. We equip the space $\mathscr{F}_d$ with a zero-mean Gaussian measure μ_d whose covariance is defined in such a way that

$$\int_{\mathscr{F}_d} \langle \eta_{d,\mathbf{i}}, f\rangle_{\mathscr{F}_d} \left\langle \eta_{d,\mathbf{j}}, f\right\rangle_{\mathscr{F}_d} \mu_d(\mathrm{d}f) = \beta_{d,\mathbf{i}}\delta_{\mathbf{i},\mathbf{j}} \quad \text{for all} \quad \mathbf{i},\mathbf{j} \in \mathbb{N}^d.$$

Here the $\beta_{d,\mathbf{j}}$ are positive, and for convenience of calculation are chosen to be of product form:

$$\beta_{d,\mathbf{j}} = \prod_{\ell=1}^{d} \beta_{\ell,j_\ell}, \qquad \beta_{\ell,j_\ell} = \frac{1-\left[\omega_{\gamma_\ell^{\text{rep}}}\right]^{q+1}}{1-\omega_{\gamma_\ell^{\text{rep}}}} \left[\omega_{\gamma_\ell^{\text{rep}}}\right]^{q(j-1)} \quad \text{for } j = 1,2,\ldots, \tag{23}$$

where $q > 0$. For Gaussian measures we must assume that

$$\infty > \sum_{\mathbf{j}\in\mathbb{N}^d} \beta_{d,\mathbf{j}} = \prod_{\ell=1}^{d}\sum_{j=1}^{\infty} \beta_{\ell,j} = \prod_{\ell=1}^{d} \frac{1-[\omega_{\gamma_\ell^{\text{rep}}}]^{q+1}}{(1-\omega_{\gamma_\ell^{\text{rep}}})(1-[\omega_{\gamma_\ell^{\text{rep}}}]^{q})}, \tag{24}$$

which holds automatically.

Next, we find the covariance kernel K_d. Since $f = \sum_{\mathbf{j}\in\mathbb{N}^d} \left\langle f, \eta_{d,\mathbf{j}}\right\rangle_{\mathscr{F}_d} \eta_{d,\mathbf{j}}$ it follows that

$$\begin{aligned} K_d(\boldsymbol{x},\boldsymbol{t}) &= \int_{\mathscr{F}_d} f(\boldsymbol{x})\, f(\boldsymbol{t})\, \mu_d(\mathrm{d}f) \\ &= \sum_{\mathbf{j}\in\mathbb{N}^d} \beta_{d,\mathbf{j}}\, \eta_{d,\mathbf{j}}(\boldsymbol{x})\, \eta_{d,\mathbf{j}}(\boldsymbol{t}) = \sum_{\mathbf{j}\in\mathbb{N}^d} \beta_{d,\mathbf{j}}\, \tilde{\lambda}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}\, \tilde{\varphi}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}(\boldsymbol{x})\, \tilde{\varphi}_{d,\boldsymbol{\gamma}^{\text{rep}},\mathbf{j}}(\boldsymbol{t}). \end{aligned} \tag{25}$$

Thus, the eigenvalues for the covariance kernel are

$$\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}} = \beta_{d,\mathbf{j}}\,\tilde{\lambda}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}} = \prod_{\ell=1}^{d} \beta_{\ell,j_\ell}\tilde{\lambda}_{\gamma_\ell^{\mathrm{rep}},j_\ell} = \prod_{\ell=1}^{d} \left(1-\left[\omega_{\gamma_\ell^{\mathrm{rep}}}\right]^{q+1}\right)\left[\omega_{\gamma_\ell^{\mathrm{rep}}}\right]^{(q+1)(j_\ell-1)}.$$

The covariance kernel, K_d, can be compared with the formula for the reproducing kernel,

$$K_d^{\mathrm{rep}}(\boldsymbol{x},\boldsymbol{t}) = \sum_{\mathbf{j}\in\mathbb{N}^d} \eta_{d,\mathbf{j}}(\boldsymbol{x})\,\eta_{d,\mathbf{j}}(\boldsymbol{t}) = \sum_{\mathbf{j}\in\mathbb{N}^d} \tilde{\lambda}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}}\,\tilde{\varphi}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}}(\boldsymbol{x})\,\tilde{\varphi}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}}(\boldsymbol{t}).$$

It would be tempting to have $K_d = K_d^{\mathrm{rep}}$, which holds for $\beta_{d,\mathbf{j}} = 1$, for all $\mathbf{j}\in\mathbb{N}^d$, i.e., $q = 0$ in (23). This is, however, *not* allowed since the sum of $\beta_{d,\mathbf{j}}$ must be finite. Note that for this choice of $\beta_{d,\mathbf{j}}$ the covariance kernel, K_d, is of product form, but no longer a Gaussian. This is because while the eigenfunctions in the expansion (25) are $\tilde{\varphi}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}}$, the eigenvalues are not those corresponding to the Gaussian kernel. Note that (10) is naturally satisfied because

$$\sum_{\mathbf{j}\in\mathbb{N}^d} \tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}} = \sum_{\mathbf{j}\in\mathbb{N}^d} \beta_{d,\mathbf{j}}\,\tilde{\lambda}_{d,\boldsymbol{\gamma}^{\mathrm{rep}},\mathbf{j}} = \prod_{\ell=1}^{d}\sum_{j=1}^{\infty} \beta_{\ell,j}\,\tilde{\lambda}_{\gamma_\ell^{\mathrm{rep}},j} = \prod_{\ell=1}^{d} 1 = 1. \tag{26}$$

We stress that the worst case setting is studied for the unit ball of $\mathscr{F}_d$ whereas the average case setting is defined for the whole space $\mathscr{F}_d$. However, it is known that the average case setting for the unit ball is roughly the same as the average case setting for the whole space if the sum of the eigenvalues is of order one, see Theorem 5.8.1 of Chap. 6 and Lemma 2.9.3 of the Appendix in [18]. For our purpose we need to assume that this holds uniformly in dimension, namely, that the $\beta_{d,\mathbf{j}}$ are chosen to satisfy

$$\sup_{d\in\mathbb{N}} \sum_{\mathbf{j}\in\mathbb{N}^d} \beta_{d,\mathbf{j}} < \infty. \tag{27}$$

From (24) we easily conclude that (27) holds iff the sum $\sum_{\ell=1}^{\infty}\left[\omega_{\gamma_\ell^{\mathrm{rep}}}\right]^{\min(q,1)}$ converges. Since $\omega_\gamma \asymp \gamma^2$, this implies that $\sum_{\ell=1}^{\infty}\left[\gamma_\ell^{\mathrm{rep}}\right]^{2\min(q,1)} < \infty$ is needed to guarantee that the average case for the whole function space $\mathscr{F}_d$ is roughly the same as the average case setting for the unit ball of $\mathscr{F}_d$, and this makes the comparison between the worst and average case settings fair. Note that the convergence of the last series implies that $r(\boldsymbol{\gamma}^{\mathrm{rep}}) \geq 1/(2\min(q,1)) \geq 1/2$. That is why we need to assume that $\sum_{\ell=1}^{\infty}[\gamma_\ell^{\mathrm{rep}}]^2 < \infty$, and $q \geq 1/(2r(\boldsymbol{\gamma}^{\mathrm{rep}}))$.

Inspecting the formula for $\tilde{\lambda}_{d,\boldsymbol{\gamma},\mathbf{j}}$ above and following the arguments leading to Theorem 1, we obtain the corresponding exponents for dimension-independent convergence and strong tractability for the average case setting over the unit ball in $\mathscr{F}_d = H(K_d^{\mathrm{rep}})$. These results are summarized in the theorem below.

Theorem 2. *Consider $\mathscr{L}_2$ approximation for the function space $H(K_d^{\mathrm{rep}})$.*

- *Consider the worst case setting for the normalized error criterion.*
  - *For the class Λ^{all}, dimension-independent convergence and strong polynomial tractability hold iff $r(\boldsymbol{\gamma}^{\mathrm{rep}}) > 0$. If so, their exponents are*

    $$p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} = r(\boldsymbol{\gamma}^{\mathrm{rep}}).$$

  - *For the class Λ^{std}, assume that $r(\boldsymbol{\gamma}^{\mathrm{rep}}) > 1/2$. Then dimension-independent convergence and strong polynomial tractability hold and their exponents satisfy*

    $$\frac{r(\boldsymbol{\gamma}^{\mathrm{rep}})}{1 + \frac{1}{2r(\boldsymbol{\gamma}^{\mathrm{rep}})}} \le p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} \le r(\boldsymbol{\gamma}^{\mathrm{rep}}).$$

- *Consider the average case setting defined as in this section for weights satisfying (23), and for $\sum_{\ell=1}^{\infty} [\gamma_\ell^{\mathrm{rep}}]^2 < \infty$ so that $r(\boldsymbol{\gamma}^{\mathrm{rep}}) \ge 1/2$.*
  - *The average case setting over the whole space and the unit ball of the function space $H(K_d^{\mathrm{rep}})$ are roughly the same.*
  - *For both classes Λ^{all} and Λ^{std}, dimension-independent convergence and strong polynomial tractability hold and their exponents are*

    $$p_{\mathrm{cnv}} = p_{\mathrm{str}}^{-1} = (q+1) r(\gamma^{\mathrm{rep}}) - 1/2 \qquad \textit{for all} \quad q \ge 1/(2r(\boldsymbol{\gamma}^{\mathrm{rep}})).$$

  - *If $q = 1/(2r(\boldsymbol{\gamma}^{\mathrm{rep}}))$ then dimension-independent convergence and strong polynomial tractability exponents are the same in the worst and average case setting for the class Λ^{all}.*

Acknowledgements This article is dedicated to Stefan Heinrich on the occasion of his 60th birthday. We are grateful for many fruitful discussions with several colleagues. The authors were partially supported by the National Science Foundation, the first and second author under DMS-0713848 and DMS-1115392, and the third author under DMS-0914345. The second author was also partially supported by the Department of Energy grant SC0002100.

References

1. Buhmann, M. D. (2003) Radial Basis Functions. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge.
2. Fasshauer G. E. (2007) Meshfree Approximation Methods with MATLAB, Interdisciplinary Mathematical Sciences, vol 6. World Scientific Publishing Co., Singapore.
3. Fasshauer G. E., Hickernell F.J., Woźniakowski H. (2012) On dimension-independent rates of convergence for function approximation with Gaussian kernels. SIAM J. Numer. Anal. 50(1): 247–271.

4. Fasshauer G. E., McCourt M. J. (2012) Stable evaluation of Gaussian radial basis function interpolants, SIAM J. Sci. Comput. 34(2): A737–A762.
5. Forrester A. I. J., Sóbester A., Keane A. J. (2008) Engineering Design via Surrogate Modelling. Wiley, Chichester.
6. Hickernell F. J., Woźniakowski H. (2000) Integration and approximation in arbitrary dimensions. Adv Comput Math 12:25–58.
7. Hickernell F. J., Wasilkowski G. W., Woźniakowski H. (2008) Tractability of linear multivariate problems in the average case setting. In: Keller A, Heinrich S, Niederreiter H (eds) Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer-Verlag, Berlin, pp 423–452.
8. Kuo F. Y., Sloan I. H., Woźniakowski H. (2008) Lattice rule algorithms for multivariate approximation in the average case setting. J Complexity 24:283–323.
9. Novak E., Woźniakowski H. (2008) Tractability of Multivariate Problems Volume 1: Linear Information. No. 6 in EMS Tracts in Mathematics, European Mathematical Society.
10. Novak E., Woźniakowski H. (2010) Tractability of Multivariate Problems Volume 2: Standard Information for Functionals. No. 12 in EMS Tracts in Mathematics, European Mathematical Society.
11. Papageorgiou A., Wasilkowski G. W. (1990) On the average complexity of multivariate problems. J Complexity 6:1–23.
12. Rasmussen C. E., Williams C. (2006) Gaussian Processes for Machine Learning. MIT Press, Cambridge, Massachusetts, (online version at http://www.gaussianprocess.org/gpml/).
13. Ritter K., Wasilkowski G. W. (1996) On the average case complexity of solving Poisson equations. In: Renegar J., Shub M., Smale S. (eds) The mathematics of numerical analysis, Lectures in Appl. Math., vol 32, American Mathematical Society, Providence, Rhode Island, pp 677–687.
14. Ritter K., Wasilkowski G. W. (1997) Integration and L_2 approximation: Average case setting with isotropic Wiener measure for smooth functions. Rocky Mountain J Math 26:1541–1557.
15. SAS Institute, *JMP 9.0*, 2010.
16. Schaback, R. (2008) Limit problems for interpolation by analytic radial basis functions. J Comp Appl Math 212:127–149.
17. Stein M. L. (1999) Interpolation of Spatial Data: Some Theory for Kriging. Springer-Verlag, New York.
18. Traub J. F., Wasilkowski G. W., Woźniakowski H. (1988) Information-Based Complexity. Academic Press, Boston.
19. Wahba G. (1990) Spline Models for Observational Data, CBMS-NSF Regional Conference Series in Applied Mathematics, vol 59. SIAM, Philadelphia.
20. Wasilkowski G. W. (1986) Information of varying cardinality. J Complexity 2:204–228.
21. Wasilkowski G. W. (1993) Integration and approximation of multivariate functions: Average case complexity with isotropic Wiener measure. Bull Amer Math Soc 28:308–314.
22. Wasilkowski G. W., Woźniakowski H. (1995) Explicit cost bounds for multivariate tensor product problems. J Complexity 11:1–56.
23. Wasilkowski G. W., Woźniakowski H. (2001) On the power of standard information for weighted approximation. Found Comput Math 1:417–434.
24. Wendland H. (2005) Scattered Data Approximation. No. 17 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge.

Extensions of Atanassov's Methods for Halton Sequences

Henri Faure, Christiane Lemieux, and Xiaoheng Wang

Abstract We extend Atanassov's methods for Halton sequences in two different directions: (1) in the direction of Niederreiter $(t,s)-$sequences, (2) in the direction of generating matrices for Halton sequences. It is quite remarkable that Atanassov's method for *classical* Halton sequences applies almost "word for word" to $(t,s)-$sequences and gives an upper bound quite comparable to those of Sobol', Faure, and Niederreiter. But Atanassov also found a way to improve further his bound for classical Halton sequences by means of a clever scrambling producing sequences which he named *modified* Halton sequences. We generalize his method to nonsingular lower triangular matrices in the last part of this article.

1 Introduction

Halton sequences and their generalizations are a popular class of low-discrepancy sequences. Their relevance in practical settings has been enhanced by various improvements that have been proposed over the years (see [8] for a survey). But it is the remarkable result published by E. Atanassov in 2004 [1] that has increased their appeal from a theoretical point of view. In Theorem 2.1 of this paper, Atanassov reduced by a factor of $s!$ the value of the hidden constant c_s in the discrepancy bound of these sequences. His proof relies on a result from diophantine geometry, and as

H. Faure
Institut de Mathématiques de Luminy, Marseille, France
e-mail: faure@iml.univ-mrs.fr

C. Lemieux
University of Waterloo, Waterloo, ON, Canada
e-mail: clemieux@uwaterloo.ca

X. Wang
Harvard University, Cambridge, MA, USA
e-mail: xwang@math.harvard.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_17, © Springer-Verlag Berlin Heidelberg 2012

such, provides a new approach to study the behavior of low-discrepancy sequences. The purpose of this paper is to explore how this approach can be extended to other constructions.

Our contribution is to first extend Atanassov's methods to $(t,s)-$sequences, including Sobol' and Faure sequences, and then to a more general class of Halton sequences which makes use of generating matrices.

It is quite remarkable that Atanassov's method for the original Halton sequences applies almost "word for word" to $(t,s)-$sequences in the narrow sense (as defined in [16]) and gives an upper bound that is comparable to those of Sobol', Faure, and Niederreiter, with the same leading term. The details are provided Sect. 3, after first reviewing Halton and (t,s)-sequences in Sect. 2. This method also applies to extensions of these sequences introduced by Tezuka [17, 18]) and Niederreiter–Xing [16] as shown in our recently submitted work [9].

In [1], Atanassov also introduces a family of sequences called *modified Halton sequences*, and proves that an even better behavior for the constant c_s holds in that case. So far, this approach has no equivalent for $(t,s)-$sequences. In fact, this method works for Halton sequences and gives asymptotic improvements thanks to the structure of these sequences, which is completely different from the structure of $(t,s)-$sequences.

However, what we propose to do here is to extend these modified Halton sequences, which rely on so-called *admissible integers*, by using what we call *admissible matrices*. As shown later in Sect. 4, the same improved behavior holds for this more general construction.

Another direction for generalizations would be to consider a larger family including both Halton and $(t,s)-$sequences. Until now, attempts in this direction have been disappointing, except in the almost trivial case of $(0,s)-$sequences in variable base which, in fact, are very close to original Halton sequences (see [7] and [11] more recently, where many other references are given).

We end the introduction with a review of the notion of discrepancy, which will be used throughout the paper. Various types exist but here, for short, we only consider the so-called *extreme discrepancy*, which corresponds to the worst case error in the domain of complexity of multivariate problems. Assume we have a point set $\mathscr{P}_N = \{X_1, \ldots, X_N\} \subseteq I^s = [0,1]^s$ and denote $\mathscr{J}$ (resp $\mathscr{J}^*$) the set of intervals J of I^s of the form $J = \prod_{j=1}^{s} [y_j, z_j)$, where $0 \leq y_j < z_j \leq 1$ (resp. $J = \prod_{j=1}^{s} [0, z_j)$). Then the *discrepancy function* of $\mathscr{P}_N$ on J is the difference

$$E(J;N) = A(J;\mathscr{P}_N) - NV(J),$$

where $A(J;\mathscr{P}_N) = \#\{n; 1 \leq n \leq N, X_n \in J\}$ is the number of points in $\mathscr{P}_N$ that fall in the subinterval J, and $V(J) = \prod_{j=1}^{s}(z_j - y_j)$ is the volume of J.

Then, the *star (extreme) discrepancy* D^* and the *(extreme) discrepancy* D of $\mathscr{P}_N$ are defined by

$$D^*(\mathscr{P}_N) = \sup_{J \in \mathscr{J}^*} |E(J;N)| \quad \text{and} \quad D(\mathscr{P}_N) = \sup_{J \in \mathscr{J}} |E(J;N)|.$$

It is well known that $D^*(\mathscr{P}_N) \leq D(\mathscr{P}_N) \leq 2^s D^*(\mathscr{P}_N)$. For an infinite sequence X, we denote by $D(N, X)$ and $D^*(N, X)$ the discrepancies of its first N points. Note that several authors have a $1/N$ factor when defining the above quantities.

A sequence satisfying $D^*(N, X) \in O((\log N)^s)$ is typically considered to be a *low-discrepancy sequence*. But the constant hidden in the O notation needs to be made explicit to make comparisons possible across sequences. This is achieved in many papers with an inequality of the form

$$D^*(N, X) \leq c_s(\log N)^s + O((\log N)^{s-1}). \tag{1}$$

As mentioned before, the constant c_s in this inequality is the main object of study in [1], as well as in the present paper.

2 Review of Halton and (t, s)−Sequences

2.1 Generalized Halton Sequences

Halton sequences are s-dimensional sequences, with values in the hypercube I^s. They are obtained using one-dimensional van der Corput sequences S_b in base b for each coordinate, defined as follows: For any integer $n \geq 1$

$$S_b(n) = \sum_{r=0}^{\infty} \frac{a_r(n)}{b^{r+1}}, \text{ where } n-1 = \sum_{r=0}^{\infty} a_r(n)\, b^r \text{ (b-adic expansion of } n-1).$$

An s-dimensional *Halton sequence* [10] $X_1, X_2, \ldots$ in I^s is defined as

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

where the b_j's, for $j = 1, \ldots, s$, are pairwise coprime.

A *generalized van der Corput sequence* [4] is obtained by scrambling the digits with a sequence $\Sigma = (\sigma_r)_{r\geq 0}$ of permutations of $\mathbb{Z}_b = \{0, 1, \ldots, b-1\}$:

$$S_b^{\Sigma}(n) = \sum_{r=0}^{\infty} \frac{\sigma_r\big(a_r(n)\big)}{b^{r+1}}. \tag{3}$$

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 S_b is obtained by taking $\sigma_r = id$ for all $r \geq 0$, where id stands for the identity permutation over $\mathbb{Z}_b$.

A *generalized Halton sequence* [6] $X_1, X_2, \ldots$ in I^s is defined by choosing s generalized van der Corput sequences:

$$X_n = (S_{b_1}^{\Sigma_1}(n), \dots, S_{b_s}^{\Sigma_s}(n)),\ n \geq 1, \tag{4}$$

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

Throughout the paper, we denote respectively by H and GH the Halton and generalized Halton sequence defined by (2) and (4), in which case, to avoid some difficulties, for $1 \leq j \leq s$, the sequence $\Sigma_j = (\sigma_{j,r})_{r\geq 0}$ satisfies $\sigma_{j,r}(0) \neq b_j - 1$ for infinitely many r. Various bounds for the discrepancy of Halton sequences have been obtained since their introduction by Halton—by Meijer, Faure, Niederreiter—all of them by refinements of the same idea. But the major theoretical improvement goes back to Atanassov [1, Theorem 2.1], with a completely different proof using an argument of diophantine geometry:

$$D^*(N, GH) \leq \frac{1}{s!}\prod_{j=1}^{s}\left(\frac{(b_j-1)\log N}{2\log b_j} + s\right) + \sum_{k=0}^{s-1}\frac{b_{k+1}}{k!}\prod_{j=1}^{k}\left(\left\lfloor\frac{b_j}{2}\right\rfloor\frac{\log N}{\log b_j} + k\right) + u, \tag{5}$$

where $u = 0$ when all bases b_j are odd, and

$$u = \frac{b_j}{2(s-1)!}\prod_{1\leq i\leq s, i\neq j}\left(\frac{(b_i-1)\log N}{2\log b_i} + s - 1\right)$$

if b_j is the even number among them. Therefore estimate (1) holds with constant

$$c_s = \frac{1}{s!}\prod_{j=1}^{s}\frac{b_j - 1}{2\log b_j}. \tag{6}$$

By making the constant c_s smaller by a factor $s!$ compared to previously established bounds, it is going to 0, instead of infinity, as s goes to infinity!

2.2 (t, s)–*Sequences*

The concept of (t, s)–sequences has been introduced by Niederreiter to give a general framework for various constructions including Sobol' and Faure sequences.

Definition 1. Given an integer $b \geq 2$, an *elementary interval* in I^s is an interval of the form $\prod_{i=1}^{s}[a_i b^{-d_i}, (a_i + 1)b^{-d_i})$ where a_i, d_i are nonnegative integers with $0 \leq a_i < b^{d_i}$ for $1 \leq i \leq s$.
Given integers t, m with $0 \leq t \leq m$, a (t, m, s)–*net in base* b is an s-dimensional set with b^m points such that any elementary interval in base b with volume b^{t-m} contains exactly b^t points of the set.

An s-dimensional sequence $X_1, X_2, \ldots$ in I^s is a $(t,s)-$*sequence* if the subset $\{X_n : kb^m < n \leq (k+1)b^m\}$ is a $(t,m,s)-$net in base b for all integers $k \geq 0$ and $m \geq t$.

Further generalizations by Niederreiter and Xing would require an extension of that definition with the so-called truncation operator. To avoid the additional developments required to explain these, we leave them out. Issues related to the construction of these sequences and the optimization of the quality parameter t are not relevant for our purpose in Sect. 3. But since we will use the digital method with generating matrices for Halton sequences in Sect. 4, we now briefly recall that method for constructing $(t,s)-$sequences in base b.

A *linearly scrambled* van der Corput sequence is obtained by choosing an $\infty \times \infty$ matrix $C = (C_{r,l})_{r \geq 0, l \geq 0}$ with elements in $\mathbb{Z}_b$, and then defining the nth term of this one-dimensional sequence as

$$S_b^C(n) = \sum_{r=0}^{\infty} y_{n,r} b^{-(r+1)} \quad \text{with} \quad y_{n,r} = \sum_{l=0}^{\infty} C_{r,l} a_l(n) \bmod b, \tag{7}$$

where $a_r(n)$ is the r-th digit of the b-adic expansion of $n - 1 = \sum_{r=0}^{\infty} a_r(n)\, b^r$.

Then, in arbitrary dimension s, one has to choose s linearly scrambled van der Corput sequences with generating matrices $C_1, \ldots, C_s$ to define the so-called *digital sequence* $(S_b^{C_1}, \ldots, S_b^{C_s})$ as proposed by Niederreiter in [14]. Of course the generating matrices must satisfy strong properties to produce low-discrepancy sequences. Special cases are the Sobol' sequences—defined in base $b = 2$ and making use of primitive polynomials to construct the non-singular upper triangular (NUT) C_i recursively—and the Faure sequences—defined in a prime base $b \geq s$ and taking C_i as the NUT Pascal matrix in $\mathbb{Z}_b$ raised to the power $i - 1$.

As to bounds for the star discrepancy, $(t,s)-$sequences satisfy estimate (1) with constant c_s (see for instance [3, 14])

$$c_s = \frac{b^t}{s!} \frac{b-1}{2\lfloor \frac{b}{2} \rfloor} \left(\frac{\lfloor \frac{b}{2} \rfloor}{\log b} \right)^s. \tag{8}$$

Note that Kritzer [12] recently improved constants c_s in (8) by a factor $1/2$ for odd $b \geq 3$ and $s \geq 2$, and by a factor $1/3$ for $b = 2$ and $s \geq 5$ (a similar result holds for even b).

3 Atanassov's Method Applied to $(t,s)-$Sequences

In this section, we apply Atanassov's method to $(t,s)-$sequences and obtain a new proof for estimate (1) and constant (8). To do so, we need to recall an important property of $(t,s)-$sequences and lemmas used in [1], reformulated here for convenience with base b instead of bases p_i (in brackets we recall the corresponding lemmas in [1] with label A). In what follows, $\mathscr{P}_N$ denotes the set containing the first N points of a sequence X.

Property 1. (**Lemma A.3.1.**) Let X be a $(t,s)-$sequence. Let $J = \prod_{i=1}^{s}[b_i b^{-d_i}, c_i b^{-d_i})$ where b_i, c_i are integers satisfying $0 \le b_i < c_i \le b^{d_i}$. Then

$$A(J;\mathscr{P}_N) = kb^t(c_1-b_1)\cdots(c_s-b_s) \text{ where } N = kb^t b^{d_1}\cdots b^{d_s}\ (k \ge 0) \quad \text{and}$$

$$|A(J;\mathscr{P}_N) - NV(J)| \le b^t \prod_{i=1}^{s}(c_i - b_i), \text{ for any integer } N \ge 1.$$

This property directly follows from the definition of $(t,s)-$sequences and is left for the reader to verify.

Lemma 1. *(**Lemma A.3.3.**) Let $N \ge 1, k \ge 1$ and $b \ge 2$ be integers. For integers $j \ge 0, 1 \le i \le k$, let some numbers $c_j^{(i)} \ge 0$ be given, satisfying $c_0^{(i)} \le 1$ and $c_j^{(i)} \le c$ for $j \ge 1$, for some fixed number c. Then*

$$\sum_{(j_1,\ldots,j_k)|b^{j_1}\cdots b^{j_k}\le N} \prod_{i=1}^{k} c_{j_i}^{(i)} \le \frac{1}{k!}\left(c\frac{\log N}{\log b} + k\right)^k. \tag{9}$$

For convenience, all the j_i's are nonnegative unless otherwise stated.

Proof. The proof proceeds very closely to the one given for Lemma 3.3 in [1], except that here we work with a single base b rather than with s different bases. For each $m \in \{0,1,\ldots,k\}$, fix a subset $L = \{i_1,\ldots,i_m\}$ of $\{1,\ldots,k\}$ and consider the contributions of all the k-tuples $\mathbf{j}$ with $j_r > 0$ for $r \in L$, and $j_r = 0$ for $r \notin L$, with $\prod_{i=1}^{k} b^{j_i} = \prod_{i\in L} b^{j_i} \le N$. One can verify as in [1, Lemma A.3.2] that there are $\frac{1}{m!}\left(\frac{\log N}{\log b}\right)^m$ such k-tuples, each having a contribution of

$$\prod_{i=1}^{k} c_{j_i}^{(i)} = \prod_{i\in L} c_{j_i}^{(i)} \prod_{i\notin L} c_{j_i}^{(i)} \le \prod_{i\in L} c \prod_{i\notin L} 1 = c^m.$$

Expanding both sides of (9), the result now follows since $\frac{1}{m!} \le \frac{1}{k!}k^{k-m}$. □

Definition 2. (**Definition A.3.2.**) Consider an interval $J \subseteq I^s$. We call a *signed splitting* of J any collection of intervals $J_1,\ldots,J_n$ and respective signs $\epsilon_1,\ldots,\epsilon_n$ equal to ± 1, such that for any (finitely) additive function ν on the intervals in I^s, we have $\nu(J) = \sum_{i=1}^{n} \epsilon_i \nu(J_i)$.

The following lemma is taken from [1], in a slightly modified form.

Lemma 2. *(**Lemma A.3.5.**) Let $J = \prod_{i=1}^{s}[0, z^{(i)})$ be an s-dimensional interval and, for each $1 \le i \le s$, let $n_i \ge 0$ be given integers. Set $z_0^{(i)} = 0$, $z_{n_i+1}^{(i)} = z^{(i)}$ and, if $n_i \ge 1$, let $z_j^{(i)} \in [0,1]$ be arbitrary given numbers for $1 \le j \le n_i$. Then the collection of intervals $\prod_{i=1}^{s}[\min(z_{j_i}^{(i)}, z_{j_i+1}^{(i)}), \max(z_{j_i}^{(i)}, z_{j_i+1}^{(i)}))$, with signs $\epsilon(j_1,\dots,j_s) = \prod_{i=1}^{s} sgn(z_{j_i+1}^{(i)} - z_{j_i}^{(i)})$, for $0 \le j_i \le n_i$, is a signed splitting of J.*

Now we have all the ingredients to prove the following theorem:

Theorem 1. *The discrepancy bound for a (t,s)−sequence X in base b satisfies*

$$D^*(N,X) \le \frac{b^t}{s!}\left(\left\lfloor\frac{b}{2}\right\rfloor\frac{\log N}{\log b} + s\right)^s + b^t\sum_{k=0}^{s-1}\frac{b}{k!}\left(\left\lfloor\frac{b}{2}\right\rfloor\frac{\log N}{\log b} + k\right)^k. \qquad (10)$$

Proof. As in [5] and [1], we will use special numeration systems in base b—using signed digits a_j bounded by $\lfloor\frac{b}{2}\rfloor$—to expand reals in $[0,1)$. That is, we write $z \in [0,1)$ as

$$z = \sum_{j=0}^{\infty} a_j b^{-j} \begin{cases} \text{with } |a_j| \le \frac{b-1}{2} \text{ if } b \text{ is odd} \\ \text{with } |a_j| \le \frac{b}{2} \text{ and } |a_j| + |a_{j+1}| \le b-1 \text{ if } b \text{ is even.} \end{cases} \qquad (11)$$

The existence and unicity of such expansions are obtained by induction, see [1, p. 21–22] or [19, p. 12–13] where more details are given. For later use, it is worth pointing out that the expansion starts at b^0 and as a result, it is easy to see that a_0 is either 0 or 1.

Now we can begin the proof: Pick any $\mathbf{z} = (z^{(1)},\dots,z^{(s)}) \in [0,1)^s$. Expand each $z^{(i)}$ as $\sum_{j=0}^{\infty} a_j^{(i)} b^{-j}$ according to our numeration systems (11) above.

Let $n := \left\lfloor\frac{\log N}{\log b}\right\rfloor$ and define $z_0^{(i)} = 0$ and $z_{n+1}^{(i)} = z^{(i)}$. Consider the numbers $z_k^{(i)} = \sum_{j=0}^{k-1} a_j^{(i)} b^{-j}$ for $k = 1,\dots,n$. Applying Lemma 2 with $n_i = n$, we expand $J = \prod_{i=1}^{s}[0, z^{(i)})$ using $(z_j^{(i)})_{j=1}^{n+1}$, obtaining a signed splitting

$$I(\mathbf{j}) = \prod_{i=1}^{s}[\min(z_{j_i}^{(i)}, z_{j_i+1}^{(i)}), \max(z_{j_i}^{(i)}, z_{j_i+1}^{(i)})), \qquad 0 \le j_i \le n, \qquad (12)$$

and signs $\epsilon(j_1,\dots,j_s) = \prod_{i=1}^{s}\operatorname{sgn}(z_{j_i+1}^{(i)} - z_{j_i}^{(i)})$, where $\mathbf{j} = (j_1,\dots,j_s)$.

Since V and $A(.;\mathscr{P}_N)$ are both additive, so is any scalar linear combination of them, and hence $A(J;\mathscr{P}_N) - NV(J)$ may be expanded as

$$A(J;\mathscr{P}_N) - NV(J) = \sum_{j_1=0}^{n}\cdots\sum_{j_s=0}^{n}\epsilon(\mathbf{j})\,(A(I(\mathbf{j});\mathscr{P}_N) - NV(I(\mathbf{j}))) =: \Sigma_1 + \Sigma_2 \qquad (13)$$

where we rearrange the terms so that in Σ_1 we put the terms $\mathbf{j}$ such that $b^{j_1}\cdots b^{j_s} \leq N$ (that is $j_1 + \cdots + j_s \leq n$) and in Σ_2 the rest. Notice that in Σ_1, the j_i's are small, so the corresponding $I(\mathbf{j})$ is bigger. Hence, Σ_1 deals with the coarser part whereas Σ_2 deals with the finer part.

It is easy to deal with Σ_1: from Property 1 and since $z_{k+1}^{(i)} - z_k^{(i)} = a_k^{(i)} b^{-k}$, we have that

$$|\,A(I(\mathbf{j});\mathscr{P}_N) - NV(I(\mathbf{j}))| \leq b^t \prod_{i=1}^{s} |\,z_{j_i+1}^{(i)} - z_{j_i}^{(i)}|\,b^{j_i} = b^t \prod_{i=1}^{s} |\,a_{j_i}^{(i)}|. \tag{14}$$

Hence, applying Lemma 1 with $k = s$, $c_j^{(i)} = |a_j^{(i)}|$ and $c = \left\lfloor \frac{b}{2} \right\rfloor$, we obtain

$$|\Sigma_1| \leq \sum_{\mathbf{j}|b^{j_1}\cdots b^{j_s} \leq N} |A(I(\mathbf{j});\mathscr{P}_N) - NV(I(\mathbf{j}))| \leq \frac{b^t}{s!}\left(\left\lfloor \frac{b}{2} \right\rfloor \frac{\log N}{\log b} + s\right)^s$$

which is the first part of the bound of Theorem 1.

The terms gathered in Σ_2 give the second part of the bound of Theorem 1, i.e., the part in $O((\log N)^{s-1})$. The idea of Atanassov for his proof of Theorem 2.1 for Halton sequences is to divide the set of s-tuples $\mathbf{j}$ in Σ_2 into s disjoint sets included in larger ones for which Lemma 1 applies and gives the desired upper bound. His proof is very terse. It has been rewritten in detail in [19] and we refer the reader to this note for further information. Following the same approach, we can adapt the proof to (t,s)−sequences and get the second part of the bound of Theorem 1. □

From Theorem 1 we can derive the constant c_s, which for the case where b is odd is the same as in the known bound (8), and for b even is larger than (8) by a factor $b/(b-1)$ (this has recently been improved, together with the extension to Niederreiter–Xing sequences suggested in Sect. 1, in our submitted work [9]).

Corollary 1. *The discrepancy of a (t,s)−sequence X in base b satisfies (1) with*

$$c_s = \begin{cases} \frac{b^t}{s!}\left(\frac{b-1}{2\log b}\right)^s & \text{if } b \text{ is odd} \\ \frac{b^t}{s!}\left(\frac{b}{2\log b}\right)^s & \text{if } b \text{ is even.} \end{cases}$$

4 Scrambling Halton Sequences with Matrices

In this section, we generalize Atanassov's methods from [1] to Halton sequences scrambled with matrices, especially the method where he uses *admissible integers* to get a smaller constant c_s. We start by the simplest case of Theorem 2.1 from [1] extended with matrices.

4.1 Halton Sequences Scrambled with Lower Triangular Matrices

Our idea of scrambling Halton sequences with matrices goes back to the scrambling of Faure $(0, s)$-sequences in [18]: to improve the initial portions of these sequences that tend to not spread uniformly over $[0, 1)^s$, Tezuka suggested to apply linear transformations to the generating matrices of the original sequences by mean of non-singular lower triangular (NLT) matrices $A_1, \dots, A_s$. That is, he introduced the idea of *generalized Faure sequences*, which are based on generating matrices of the form $C_i = A_i P_i$, where P_i is the NUT Pascal matrix in $\mathbb{Z}_b$ raised to the power $i - 1$. Now, going back to Halton sequences, it seems natural to use similar ideas to scramble Halton sequences, as described in the following definition (see also [13, App. B]).

Definition 3. The *linearly scrambled* Halton (LSH) sequence $(X_n)_{n \geq 1}$, based on NLT matrices $A_1, \dots, A_s$, where A_i has entries in $\mathbb{Z}_{p_i}$, is obtained as

$$X_n = (S_{p_1}^{A_1}(n), \dots, S_{p_s}^{A_s}(n)), \; n \geq 1,$$

where $S_b^C(n)$ was defined in (7).

Theorem 2. *An LSH sequence satisfies the discrepancy bound (1) with c_s given by (6) (the same constant as for GH sequences).*

This theorem results from an analog of [1, Lemma 3.1]. But here, the use of NLT matrices A_i implies that there might be infinitely many $y_{n,r} = b - 1$ in (7). This introduces disruptions in the proof (when using elementary intervals), as it does for $(t, s)-$sequences generalized with linear scramblings [18] or with global function fields [16]. Hence, as in [16, 18], we must introduce the *truncation operator* to overcome this difficulty.

Truncation: Let $x = \sum_{r=0}^{\infty} x_r b^{-(r+1)}$ be a b–adic expansion of $x \in [0, 1]$, with the possibility that $x_r = b - 1$ for all but finitely many r. For every integer $m \geq 1$, we define the m-truncation of x by $[x]_{b,m} = \sum_{r=0}^{m} x_r b^{-(r+1)}$ (depending on x via its expansion). In the multi-dimensional case, the truncation is defined coordinate-wise. Next, we define an *elementary interval in bases $p_1, \dots, p_s$*, i. e., an interval of the form

$$\prod_{i=1}^{s} [l_i p_i^{-d_i}, (l_i + 1) p_i^{-d_i}), \text{ where } d_i \geq 0 \text{ and } 0 \leq l_i < p_i^{d_i} \text{ are given integers.} \quad (15)$$

In order to establish our discrepancy bound for an LSH sequence, we first need to work with the truncated version of the sequence, and to do so the following definition is useful.

Definition 4. Let $(S_{p_1}^{A_1}, \ldots, S_{p_s}^{A_s})$ be an LSH sequence. We define

$$[\mathscr{P}_N] = \{([S_{p_1}^{A_1}(n)]_{p_1,D_1}, \ldots, [S_{p_s}^{A_s}(n)]_{p_s,D_s}), 1 \le n \le N\}, \text{ where } D_i = \lceil \log N / \log p_i \rceil.$$

We refer to $[\mathscr{P}_N]$ as the first N points of a truncated version of the sequence.

The next result, about $A(J;[\mathscr{P}_N])$ viewed as a function of N, would be trivial without the truncation operator.

Lemma 3. *Let $(S_{p_1}^{A_1}, \ldots, S_{p_s}^{A_s})$ be an LSH sequence and J be an interval of the form $\prod_{i=1}^s [b_i p_i^{-d_i}, c_i p_i^{-d_i})$ with integers b_i, c_i satisfying $0 \le b_i < c_i \le p_i^{d_i}$. Then for $N \ge p_1^{d_1} \cdots p_s^{d_s}$, $A(J;[\mathscr{P}_N])$ is an increasing function of N.*

Proof. Let $D_i = \lceil \log N / \log p_i \rceil$. If $N \ge p_1^{d_1} \cdots p_s^{d_s}$, then $D_i \ge d_i$ for all i. Therefore as N increases, there can only be more points (from the truncated sequence) inside a particular interval J. The reason why we have to make sure $D_i \ge d_i$ for all i is that otherwise, as N increases some points could leave the interval J as more precision is added on their digital expansion, but once the precision D_i is greater than the precision d_i used to define the interval, then this can no longer happen. □

We then establish the following lemma, analog of [1, Lemma 3.1] and Property 1.

Lemma 4. *Let $(S_{p_1}^{A_1}, \ldots, S_{p_s}^{A_s})$ be an LSH sequence. Then for any integer $k \ge 0$, any elementary interval as in (15) contains exactly one point of the point set*

$$\left\{\left([S_{p_1}^{A_1}(n)]_{p_1,d_1}, \ldots, [S_{p_s}^{A_s}(n)]_{p_s,d_s}\right) \; ; \; kp_1^{d_1} \cdots p_s^{d_s} + 1 \le n \le (k+1)p_1^{d_1} \cdots p_s^{d_s}\right\}.$$

Moreover, for all intervals of the form $J = \prod_{i=1}^s [b_i p_i^{-d_i}, c_i p_i^{-d_i})$ with integers b_i, c_i satisfying $0 \le b_i < c_i \le p_i^{d_i}$, we have for all $k \ge 0$

$$A(J;[\mathscr{P}_N]) = k(c_1 - b_1) \cdots (c_s - b_s), \text{ where } N = kp_1^{d_1} \cdots p_s^{d_s}.$$

Proof. For short, write $X_n^{(i)} := S_{p_i}^{A_i}(n)$ for all $1 \le i \le s$. First, the condition on n implies that the digits $a_r(n)$ from the expansion of $n-1$ are uniquely determined for $r \ge p_1^{d_1} \cdots p_s^{d_s}$.

Then, it is easy to see that the digits $y_{n,r}^{(i)}$ $(0 \le r < d_i)$ defining $[X_n^{(i)}]_{p_i,d_i}$ are uniquely determined by the integers d_i, l_i describing a given elementary interval.

Now, since A_i is an NLT matrix, the $d_i \times d_i$ linear system in the unknowns $a_r(n)$ $(0 \le r < d_i)$ given by

$$A_i \; (a_0(n), \ldots, a_{d_i-1}(n))^T = (y_{n,0}^{(i)}, \ldots, y_{n,d_i-1}^{(i)})^T,$$

also has a unique solution and hence the digits $a_r(n)$ $(0 \le r < d_i)$ are uniquely determined, which means that n is unique modulo $p_i^{d_i}$ for all $1 \le i \le s$.
Finally, applying the Chinese remainder theorem, we obtain that n is unique modulo $p_1^{d_1} \cdots p_s^{d_s}$. Together with the condition $kp_1^{d_1} \cdots p_s^{d_s} + 1 \le n \le (k+1)p_1^{d_1} \cdots p_s^{d_s}$, all digits $a_r(n)$ $(r \ge 0)$ are unique and so is n, which ends the proof of the first part of Lemma 4. The second part simply results from the fact that J splits into $(c_1 - b_1) \cdots (c_s - b_s)$ disjoint elementary intervals. □

We also need the following lemma, another result that would be trivial without the truncation.

Lemma 5. *Let $(S_{p_1}^{A_1}, \ldots, S_{p_s}^{A_s})$ be an LSH sequence and J be an interval of the form $J = \prod_{i=1}^{s} [b_i p_i^{-d_i}, c_i p_i^{-d_i})$ with integers b_i, c_i satisfying $0 \le b_i < c_i \le p_i^{d_i}$. If $N < p_1^{d_1} \cdots p_s^{d_s}$ then $A(J; [\mathscr{P}_N]) \le (c_1 - b_1) \cdots (c_s - b_s)$.*

Proof. Define $\tilde{d}_i = \min(D_i, d_i)$. Let $[J]$ be defined as the smallest interval of the form $\prod_{i=1}^{s} [\tilde{b}_i p_i^{-\tilde{d}_i}, \tilde{c}_i p_i^{-\tilde{d}_i})$ with $0 \le \tilde{b}_i < \tilde{c}_i \le p_i^{\tilde{d}_i}$ and such that $J \subseteq [J]$. We can see that $[J]$ is obtained by using $\tilde{c}_i = \lceil c_i / p_i^{\tilde{d}_i - d_i} \rceil$ and $\tilde{b}_i = \lfloor b_i / p_i^{\tilde{d}_i - d_i} \rfloor$. Using the same arguments as in the proof of the previous lemma, we have that each interval of the form $\prod_{i=1}^{s} [l_i p_i^{-\tilde{d}_i}, (l_i + 1) p_i^{-\tilde{d}_i})$ has at most one point from $[\mathscr{P}_N]$. Hence

$$A(J; [\mathscr{P}_N]) \le A([J]; [\mathscr{P}_N]) \le \prod_{i=1}^{s} (\tilde{c}_i - \tilde{b}_i) \le \prod_{i=1}^{s} (c_i - b_i),$$

where the last inequality follows from the definition of $\tilde{b}_i$ and $\tilde{c}_i$. □

Now, we can give the proof of Theorem 2.

Proof. From Lemma 3 and the second part of Lemma 4, we obtain that for every $N \ge p_1^{d_1} \cdots p_s^{d_s}$ and $J = \prod_{i=1}^{s} [b_i p_i^{-d_i}, c_i p_i^{-d_i})$

$$|A(J; [\mathscr{P}_N]) - NV(J)| \le (c_1 - b_1) \cdots (c_s - b_s). \tag{16}$$

Further, Lemma 5 proves that (16) also holds when $N < p_1^{d_1} \cdots p_s^{d_s}$.

The inequality (16) is similar to the result stated in Lemma A.3.1 from [1], but note that here it applies to the truncated sequence. From that point, we can proceed as in Atanassov's proof of his Theorem 2.1, which consists in breaking down $A(J; [\mathscr{P}_N]) - NV(J)$ into a sum $\Sigma_1 + \Sigma_2$ as done in (13), and then bound each term separately. Note however that in our case, the obtained bound applies to the truncated version of the sequence. But as discussed in [15, 16], it is easy to show that if a bound of the form (1) applies to the truncated version of a sequence, it applies to the untruncated version as well (with the same constant c_s). □

4.2 Scrambling Halton Sequences with Admissible Matrices

In this section, we show that by using admissible integers to construct the matrices A_i of an LSH sequence, we obtain sequences satisfying the same improved discrepancy bound as in [1, Theorem 2.3], obtained there for modified Halton sequences, which use permutations based on admissible integers. We first need a few definitions, including that of admissible integers and the "generating–matrices" analog of these integers, which we call "admissible matrices".

Definition 5. Given non-negative integers $\alpha_1, \ldots, \alpha_s, \beta_1, \ldots, \beta_s$ and $k_1, \ldots, k_s$, we define the quantity

$$P_i^{(\beta_i)}(k_i; (\alpha_1, \ldots, \alpha_s)) := k_i^{\alpha_i + \beta_i} \prod_{1 \le j \le s, j \ne i} p_j^{\alpha_j} \bmod p_i, \qquad i = 1, \ldots, s. \quad (17)$$

Definition 6. We say that $k_1, \ldots, k_s$ are *admissible* for the primes $p_1, \ldots, p_s$ if $p_i \not| k_i$ and for each set of integers $(b_1, \ldots, b_s)$, $p_i \not| b_i$, there exists a set of integers $(\alpha_1, \ldots, \alpha_s)$ such that

$$P_i^{(0)}(k_i; (\alpha_1, \ldots, \alpha_s)) \equiv b_i \bmod p_i, \quad i = 1, \ldots, s.$$

Lemma A.4.1. Let $p_1, \ldots, p_s$ be distinct primes. Then there exist admissible integers $k_1, \ldots, k_s$.

Definition 7. Let $A_1, \ldots, A_s$ be NLT matrices in distinct prime bases $p_1, \ldots, p_s$ and let $k_1, \ldots, k_s$ be admissible integers for these bases. Then the matrices $A_i, i = 1, \ldots, s$ are *admissible* if the jth entry on their diagonal has the form $k_i^{\beta_i + j}$, $j \ge 1$, where $\beta_1, \ldots, \beta_s$ are non-negative integers. An LSH sequence based on admissible matrices $A_1, \ldots, A_s$ is called a *modified linearly scrambled Halton* (MLSH) sequence.

Atanassov's modified Halton sequence corresponds to the case where A_i is diagonal and $\beta_i = 0$ for all i, while if we take A_i diagonal and $\beta_i = 1$, then we obtain the sequences used in the experiments in [2] (where the authors also apply digital shifts chosen independently (mod p_i)). It is important to take $\beta_i \ge 1$ for applications in QMC methods, otherwise the sequences behave like original Halton sequences in the usual ranges of sample sizes [8, Sect. 3, Paragraph 2].

We can now state the main result of this section.

Theorem 3. *The discrepancy of an MLSH sequence based on distinct primes bases $p_1, \ldots, p_s$, non-negative integers $\beta_1, \ldots, \beta_s$ and admissible integers $k_1, \ldots, k_s$ satisfies the bound (1) with constant*

$$c_s(p_1, \ldots, p_s) = \frac{1}{s!} \sum_{i=1}^{s} \log p_i \prod_{i=1}^{s} \frac{p_i(1 + \log p_i)}{(p_i - 1) \log p_i}.$$

The proof of Theorem 3 follows closely that of [1, Theorem 2.3], which in turn essentially proceeds through an intermediate result called *Proposition 4.1* in [1]. Here this result must be adapted to the more general setting of admissible matrices, and is described in a slightly different version in the following proposition.

Proposition 1. *For an MLSH sequence based on distinct primes $p_1, \dots, p_s$, nonnegative integers $\beta_1, \dots, \beta_s$ and admissible integers $k_1, \dots, k_s$, we have that*

$$\sum_{\mathbf{j} \in T(N)} |A(I(\mathbf{j}); [\mathscr{P}_N]) - NV(I(\mathbf{j}))| \le \sum_{\mathbf{j} \in T(N)} \left(1 + \sum_{\mathbf{l} \in M(\mathbf{p})} \frac{\| \sum_{i=1}^s (l_i/p_i) P_i^{(\beta_i)}(k_i; \mathbf{j}) \|^{-1}}{2R(\mathbf{l})} \right) + O((\log N)^{s-1}),$$

where $T(N) = \{\mathbf{j} | p_1^{j_1} \cdots p_s^{j_s} \le N, j_1, \dots, j_s \ge 0\}$, $M(\mathbf{p}) = \{\mathbf{j} | \, 0 \le j_i \le p_i - 1, j_1 + \cdots + j_s > 0\}$, $R(\mathbf{j}) = \prod_{i=1}^s r_i(j_i)$, with $r_i(m) = \max(1, \min(2m, 2(p_i - m)))$ and $\| \cdot \|$ denotes the "distance to the nearest integer" function.

Before presenting the proof of this result, we first need to recall a technical lemma from [1] and an adapted version of a key lemma used in the proof of [1, Prop. 4.1]. **Lemma A.4.2.** Let $\mathbf{p} = (p_1, \dots, p_s)$ and let $\omega = (\omega_n^{(1)}, \dots, \omega_n^{(s)})_{n=0}^{\infty}$ be a sequence in $\mathbb{Z}^s$. Let $\mathbf{b}, \mathbf{c}$ be fixed elements in $\mathbb{Z}^s$, such that $0 \le b_i < c_i \le p_i$, for $1 \le i \le s$. For $C \ge 1$, denote by $a_C(\mathbf{b}, \mathbf{c})$ the number of terms of ω among the first C such that for all $1 \le i \le s$, we have $b_i \le \omega_n^{(i)} \bmod p_i < c_i$. Then

$$\sup_{\mathbf{b},\mathbf{c}} \left| a_C(\mathbf{b}, \mathbf{c}) - C \prod_{i=1}^s \frac{c_i - b_i}{p_i} \right| \le \sum_{\mathbf{j} \in M(\mathbf{p})} \frac{|S_C(\mathbf{j}, \omega)|}{R(\mathbf{j})}, \tag{18}$$

where $S_C(\mathbf{j}, \omega) = \sum_{n=0}^{C-1} e\left(\sum_{k=1}^s \frac{j_k \omega_n^{(k)}}{p_k} \right)$ and $e(x) = \exp(2i\pi x)$.

This result is applied in Lemma 6 below, but to the counting function $A(J; [\mathscr{P}_N])$ in place of $a_C(\mathbf{b}, \mathbf{c})$. Hence, the discrepancy function will be estimated by means of a trigonometrical sum, which in turn will give the part $\| \sum_{i=1}^s (l_i/p_i) P_i^{(\beta_i)}(k_i; \mathbf{j}) \|^{-1}$ in the upper bound of Proposition 1.

Lemma 6. *Let X be an MLSH sequence in bases $p_1, \dots, p_s$ as in Definition 7. Fix some elementary interval $I = \prod_{i=1}^s [a_i p_i^{-\alpha_i}, (a_i + 1) p_i^{-\alpha_i})$ with $0 \le a_i < p_i^{\alpha_i}$, and a subinterval $J = \prod_{i=1}^s [a_i p_i^{-\alpha_i} + b_i p_i^{-\alpha_i - 1}, a_i p_i^{-\alpha_i} + c_i p_i^{-\alpha_i - 1})$ with $0 \le b_i < c_i \le p_i$.*
Let $N > \prod_{i=1}^s p_i^{\alpha_i}$ and let n_0 (whose existence will be proved) be the smallest integer such that $[X_{n_0}] \in I$ (the notation $[X_n] = ([X_n^{(1)}]_{p_1, D_1}, \dots, [X_n^{(s)}]_{p_s, D_s})$ has been introduced in Definition 4). Suppose that $[X_{n_0}]$ belongs to

$$\prod_{i=1}^s [a_i p_i^{-\alpha_i} + d_i p_i^{-\alpha_i - 1}, a_i p_i^{-\alpha_i} + (d_i + 1) p_i^{-\alpha_i - 1}),$$

and let $\omega = \{\omega_t\}_{t=0}^{\infty}$ *in* $\mathbb{Z}^s$ *be defined by* $\omega_t^{(i)} = d_i + t P_i^{(\beta_i)}(k_i; (\alpha_1, \ldots, \alpha_s))$*. Then*

1. *We have that* $n_0 < \prod_{i=1}^s p_i^{\alpha_i}$ *and the indices n of the terms* $[X_n]$ *of* $[\mathscr{P}_N]$ *that belong to I are of the form* $n = n_0 + t \prod_{i=1}^s p_i^{\alpha_i}$*.*
2. *For these n,* $[X_n] \in J$ *if and only if for some integers* $(l_1, \ldots, l_s)$*,* $l_i \in \{b_i, \ldots, c_i - 1\}$*, the following system of congruences is satisfied by t:*

$$\omega_t^{(i)} = d_i + t P_i^{(\beta_i)}(k_i; (\alpha_1, \ldots, \alpha_s)) \equiv l_i \bmod p_i, \quad i = 1, \ldots, s. \qquad (19)$$

3. *If C is the largest integer with* $n_0 + (C-1)\prod_{i=1}^s p_i^{\alpha_i} < N$*, then*

$$|A(J; [\mathscr{P}_N]) - NV(J)| < 1 + \sum_{\mathbf{l} \in M(\mathbf{p})} \frac{|S_C(\mathbf{l}, \omega)|}{R(\mathbf{l})}.$$

Proof. We consider each of the three claims one by one.

1. This has been dealt with in the proof of Lemma 4 (first part with $k = t$), which applies here since an MLSH sequence is a special case of an LSH sequence.
2. We first note that for $[X_n]$ to be in J, for each fixed i the $(\alpha_i + 1)$st digit of $[X_n^{(i)}]$ must be in $\{b_i, \ldots, c_i - 1\}$. Hence we need to show that this digit is given by (19). By the definition of n_0, we know that $A_i\,(a_0(n_0), \ldots, a_{d_i-1}(n_0))^T = (*, \ldots, *, d_i, *, \ldots)^T$ (where d_i is the $(\alpha_i + 1)$st digit), $(a_0(n_0), \ldots, a_{d_i-1}(n_0))$ coming from the expansion of $n_0 - 1$ in base p_i. For brevity, let $P_i := \prod_{j=1, j\neq i}^s p_j^{\alpha_j} \bmod p_i$. Since the $(\alpha_i + 1)$st digit of $\prod_{j=1}^s p_j^{\alpha_j}$ in base p_i is tP_i, we have that $(a_0(n), \ldots, a_{d_i-1}(n)) = (a_0(n_0), \ldots, a_{d_i-1}(n_0)) + (0, \ldots, 0, tP_i, *, \ldots)$. Note that possible carries to higher order digits are absorbed in the stars $*$. Now,

$$\begin{aligned} A_i(a_0(n), \ldots, a_{d_i-1}(n))^T &= A_i(a_0(n_0), \ldots, a_{d_i-1}(n_0))^T + A_i(0, \ldots, 0, tP_i, *, \ldots)^T \\ &= (*, \ldots, *, d_i, *, \ldots)^T + (0, \ldots, 0, tk_i^{\alpha_i+\beta_i} P_i, *, \ldots) \end{aligned}$$

by definition of A_i. Therefore, the first α_i digits of $[X_n^{(i)}]$ and $[X_{n_0}^{(i)}]$ are equal and the $(\alpha_i + 1)$st digit of $[X_n^{(i)}]$ is $d_i + tk_i^{\alpha_i+\beta_i} P_i \equiv d_i + t P_i^{(\beta_i)}(k_i; \boldsymbol{\alpha}) \bmod p_i$, as desired.
3. We apply Lemma A.4.2 with $a_C(\mathbf{b}, \mathbf{c}) = A(J; [\mathscr{P}_N])$ and use the inequalities

$$C \prod_{i=1}^s \frac{c_i - b_i}{p_i} - 1 \le NV(J) \le (1 + C) \prod_{i=1}^s \frac{c_i - b_i}{p_i} \le 1 + C \prod_{i=1}^s \frac{c_i - b_i}{p_i}$$

resulting from the hypothesis of item 3. □

Proof. (**Proposition 1**) As in [19] we first consider the case where $j_i \ge 1$ for all i, as this allows use to use Lemma 6 The interval $I(\mathbf{j})$ is contained inside some elementary interval $G = \prod_{i=1}^s [c_i p_i^{-j_i}, (c_i + 1) p_i^{-j_i})$. We define a sequence ω as in

Lemma 6, where the integers d_i are determined by the condition that the first term of the sequence σ that falls in G fits into the interval

$$\prod_{i=1}^{s}[c_i p_i^{-j_i} + d_i p_i^{-j_i-1}, c_i p_i^{-j_i} + (d_i+1)p_i^{-j_i-1}). \tag{20}$$

Hence $\omega_n^{(i)} = d_i + nP_i^{(\beta_i)}(k_i, \mathbf{j})$. From part (3) of Lemma 6, it follows that

$$|A(I(\mathbf{j});[\mathscr{P}_N]) - NV(I(\mathbf{j}))| < 1 + \sum_{\mathbf{l}\in M(\mathbf{p})} \frac{|S_K(\mathbf{l},\omega)|}{R(\mathbf{l})}, \tag{21}$$

where K is the number of terms of the MLSH sequence among the first N terms that fall into G. Since the p_i's are coprime, we see that $P_i^{(\beta_i)}(k_i, \mathbf{j}) \neq 0$, in particular, it is not divisible by p_i and hence coprime to p_i. For any $\mathbf{l} \in M(\mathbf{p})$, by definition, there is an l_t, with $1 \le t \le s$ such that $l_t \neq 0$, and so $p_t \nmid l_t$. These properties imply that $\alpha = \sum_{i=1}^{s} \frac{l_i}{p_i} P_i^{(\beta_i)}(k_i;\mathbf{j})$ is not an integer. Thus we have

$$\begin{aligned}|S_K(\mathbf{l},\omega)| &= \left|\sum_{n=0}^{K-1} e\left(\sum_{i=1}^{s}\frac{l_i}{p_i}(d_i + nP_i^{(\beta_i)}(k_i;\mathbf{j}))\right)\right| = \left|\sum_{n=0}^{K-1} e(n\alpha + \sum_{i=1}^{s} l_i d_i/p_i)\right| \\ &= \frac{|e(K\alpha)-1|}{|e(\alpha)-1|} \le \frac{1}{2}\left\|\sum_{i=1}^{s}\frac{l_i}{p_i}P_i^{(\beta_i)}(k_i;\mathbf{j})\right\|^{-1},\end{aligned}$$

where the last inequality is obtained by noticing that $|e(\alpha)-1| \ge 2\pi|\alpha|2/\pi = 4|\alpha|$ for $-1/2 \le \alpha \le 1/2$. Combining this result with (21), we obtain

$$\sum_{\substack{\mathbf{j}\in T(N),\\ j_i \ge 1}} |(A(I(\mathbf{j});[\mathscr{P}_N]) - NV(I(\mathbf{j}))| \le \sum_{\substack{\mathbf{j}\in T(N),\\ j_i \ge 1}} \left(1 + \sum_{\mathbf{l}\in M(\mathbf{p})} \frac{\|\sum_{i=1}^{s}\frac{l_i}{p_i}P_i^{(\beta_i)}(k_i;\mathbf{j})\|^{-1}}{2R(\mathbf{l})}\right). \tag{22}$$

In the second case, the fact that at least one j_i is 0 implies that we can use a similar approach to the one used to bound Σ_1 in Theorem 1, and the obtained bound in $O(\log^{s-1} N)$ as we are essentially working in at most $s-1$ dimensions. Observing that $T(N)$ contains the vectors $\mathbf{j}$ such that $j_i \ge 1$ for all i completes the proof.

We still need two more technical lemmas before proceeding to the proof of Theorem 2.3. The first one is directly from [1], and is useful to bound the upper bound derived in Proposition 1.

Lemma A.4.4. Let $\mathbf{p} = (p_1, \ldots, p_s)$, then

$$\sum_{\mathbf{j}\in M(\mathbf{p})} \sum_{m_1=1}^{p_1-1} \cdots \sum_{m_s=1}^{p_s-1} \frac{\| \frac{j_1 m_1}{p_1} + \ldots + \frac{j_s m_s}{p_s} \|^{-1}}{2R(\mathbf{j})} \leq \sum_{i=1}^{s} \log p_i \prod_{i=1}^{s} p_i \left(\prod_{j=1}^{s} (1 + \log p_j) - 1 \right).$$

The next one is useful to count the vectors $\mathbf{j} \in T(N)$, over which the sum that is bounded in Proposition 1 is defined. In [1, p. 30–31], this is achieved in the text of the proof but, for the sake of clarity, we prefer to state it as a last lemma.

Lemma 7. *Let* $\mathbf{a} \in \mathbb{Z}^s$ *be a vector of non-negative integers and let* $U(\mathbf{a}) := \{\mathbf{j}\ ;\ a_i K \leq j_i < (a_i + 1)K$ *for all* $1 \leq i \leq s\}$, *where* $K = \prod_{i=1}^{s}(p_i - 1)$. *The* s *functions* $P_i^{(\beta_i)}(k_i; \mathbf{j})$, $1 \leq i \leq s$, *are such that for each* $\mathbf{b} = (b_1, \ldots, b_s) \in \mathbb{Z}^s$, *with* $1 \leq b_i \leq p_i - 1$ *for all* $1 \leq i \leq s$, *there are exactly* K^{s-1} s*-tuples* $\mathbf{j} \in U(\mathbf{a})$ *such that* $P_i^{(\beta_i)}(k_i; \mathbf{j}) \equiv b_i \bmod p_i$ *for all* $1 \leq i \leq s$.

Proof. The proof essentially follows from the fact that the s functions $P_i^{(0)}(k_i; \mathbf{j})$ satisfy the property described in this Lemma 7 (see [1, p. 30]), and then the observation that $P_i^{(\beta_i)}(k_i; \mathbf{j}) \equiv b_i \bmod p_i$ if and only if $P_i^{(0)}(k_i; \mathbf{j}) \equiv k_i^{-\beta_i} b_i \bmod p_i$. □

We are now ready to prove Theorem 3.

Proof. As in the proof of Theorems 1 and 2, we first write the discrepancy function of $[\mathscr{P}_N]$ on J using (13) and similarly get

$$A(J; [\mathscr{P}_N]) - NV(J) = \Sigma_1 + \Sigma_2.$$

The terms gathered in Σ_2 are still in $O((\log N)^{s-1})$ and those in Σ_1 are divided in two sums bounded separately as follows:

$$\left|\sum\nolimits_1\right| \leq \sum_{\substack{\mathbf{j}\in T(N) \\ j_i > 0}} t(\mathbf{j}) + \sum_{\substack{\mathbf{j}\in T(N) \\ \text{some } j_i = 0}} t(\mathbf{j}). \tag{23}$$

Now, using Proposition 1 and the fact that each $\mathbf{j} \in T(N)$ is inside a box $U(\mathbf{a})$ such that the s-tuples $\mathbf{a}$ satisfy $\prod_{i=1}^{s} p_i^{a_i K} \leq \prod_{i=1}^{s} p_i^{j_i} \leq N$, we get that the first term on the right-hand side of (23) is bounded by

$$\sum_{\mathbf{a} | \prod_{i=1}^{s} p_i^{a_i \cdot K} \leq N} \ \sum_{\mathbf{j}\in U(\mathbf{a})} \left(1 + \sum_{\mathbf{l}\in M(\mathbf{p})} \frac{\| \sum_{i=1}^{s} \frac{l_i}{p_i} P_i^{(\beta_i)}(k_i; \mathbf{j}) \|^{-1}}{2R(\mathbf{l})} \right). \tag{24}$$

We also note that the second term on the right-hand side of (23) is in $O(\log^{s-1} N)$.

We then apply Lemma A.3.3 (whose base b version is given in Lemma 1) with $c = 1$ and $p_i' = p_i^K$ and get the bound $\frac{1}{s!} \prod_{i=1}^{s} \left(\frac{\log N}{K \log p_i} + s \right)$ for the number of

s-tuples $\mathbf{a}$ enumerated in the first sum of (24). Next we use Lemma 7 to enumerate and count the number of vectors $\mathbf{j}$ in $U(\mathbf{a})$ considered in the inner sum of (24). These two results together with Lemma A.4.4 give us the bound

$$\frac{1}{s!}\prod_{i=1}^{s}\left(\frac{\log N}{K\log p_i}+s\right)K^{s-1}\left(K+\sum_{i=1}^{s}\log p_i\prod_{i=1}^{s}p_i\left(-1+\prod_{i=1}^{s}(1+\log p_i)\right)\right)$$

for Σ_1. The final result can then be obtained after a few further simplifications and using the fact that, as explained in [15], a discrepancy bound holding for the truncated version of the sequence also applies to the untruncated version. □

Remark 1. The reader interested in the unfolding of the original proof by Atanassov has the choice between the text in [1, Theorem 2.3] (very terse) and its careful analysis in [19] (very detailed). With our proof of Theorem 3 in hand, we now have the opportunity to present an overview of Atanassov's proof and thus make it accessible to readers who do not wish to go over [1] or [19].

Atanassov's modified Halton sequences in bases $p_1,\ldots,p_s$, with admissible integers $k_1,\ldots,k_s$, are generalized Halton sequences in which the sequences of permutations $\Sigma_i=(\sigma_{i,r})_{r\geq 0}$ are defined by

$$\sigma_{i,r}(a) := ak_i^r \bmod p_i \qquad \text{for all } 0\leq a<p_i,\ r\geq 0,\ i=1,\ldots,s.$$

Of course they are a special case of MLSH sequences (see definitions and comments just before Theorem 3).

The basis of the proof of Theorem A.2.3 is Proposition A.4.1 which essentially reads as Proposition 1 where $\beta_i=0$.

Lemma A.4.1, which establishes the existence of admissible integers (using primitive roots modulo p_i), and Lemma A.4.2 have already been stated.

Lemma A.4.3 is the core of the proof. It reads as Lemma 6 where brackets have been removed, i.e., where the truncation is unnecessary, since Atanassov deals with diagonal matrices only.

Now, Lemma A.4.2 is applied in Lemma A.4.3 to the counting function $A(J;\mathscr{P}_N)$ in place of $a_C(\mathbf{b},\mathbf{c})$. Hence, as already noted, the discrepancy function is estimated by means of a trigonometrical sum, which gives the part $\|\sum_{i=1}^{s}(l_i/p_i)P_i^{(0)}(k_i;\mathbf{j})\|^{-1}$ in the upper bound of Proposition A.4.1. The end of the proof of Proposition A.4.1 together with the proof of Theorem A.2.3 are mainly the same as that of Proposition 1 and Theorem 3, respectively, where the brackets have to be removed and where $\beta_i=0$. The only subtle difference is in the split into two cases, $j_i\geq 1$ for all i or not. This distinction was ignored by Atanassov whereas it appears crucial at a stage of the proof (see [19] for complete details).

Acknowledgements We wish to thank the referee for his/her detailed comments, which were very helpful to improve the presentation of this manuscript. The second author acknowledges the support of NSERC for this work.

References

1. E. I. Atanassov, On the discrepancy of the Halton sequences, *Math. Balkanica, New Series* **18.1–2** (2004), 15–32.
2. E. I. Atanassov and M. Durchova, Generating and testing the modified Halton sequences. In *Fifth International Conference on Numerical Methods and Applications, Borovets 2002*, Springer-Verlag (Berlin), Lecture Notes in Computer Science **2542** (2003), 91–98.
3. J. Dick and F. Pillichshammer, Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration, Cambridge University Press, UK (2010).
4. H. Faure, Discrépance de suites associées à un système de numération (en dimension *un*), *Bull. Soc. math. France* **109** (1981), 143–182.
5. H. Faure, Discrépance de suites associées à un système de numération (en dimension s), *Acta Arith.* **61** (1982), 337–351.
6. H. Faure, On the star-discrepancy of generalized Hammersley sequences in two dimensions, *Monatsh. Math.* **101** (1986), 291–300.
7. H. Faure, Méthodes quasi-Monte Carlo multidimensionnelles, *Theoretical Computer Science* **123** (1994), 131–137.
8. H. Faure and C. Lemieux, Generalized Halton sequences in 2008: A comparative study, *ACM Trans. Model. Comp. Sim.* **19** (2009), Article 15.
9. H. Faure and C. Lemieux, Improvements on the star discrepancy of $(t,s)-$sequences. Submitted for publication, 2011.
10. J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.* **2** (1960), 184–90.
11. R. Hofer and G. Larcher, On the existence and discrepancy of certain digital Niederreiter–Halton sequences, *Acta Arith.* **141** (2010), 369–394.
12. P. Kritzer, Improved upper bounds on the star discrepancy of (t,m,s)-nets and (t,s)-sequences, *J. Complexity* **22** (2006), 336–347.
13. C. Lemieux, Monte Carlo and Quasi-Monte Carlo Sampling, Springer Series in Statistics, Springer, New York (2009).
14. H. Niederreiter, Point sets and sequences with small discrepancy, *Monatsh. Math.* **104** (1987), 273–337.
15. H. Niederreiter and F. Özbudak, Low-discrepancy sequences using duality and global function fields, *Acta Arith.* **130** (2007), 79–97.
16. H. Niederreiter and C. P. Xing, Quasirandom points and global function fields, *Finite Fields and Applications*, S. Cohen and H. Niederreiter (Eds), London Math. Soc. Lectures Notes Series **233** (1996), 269–296.
17. S. Tezuka, Polynomial arithmetic analogue of Halton sequences, *ACM Trans. Modeling and Computer Simulation* **3** (1993), 99–107.
18. S. Tezuka, A generalization of Faure sequences and its efficient implementation, Technical Report RT0105, IBM Research, Tokyo Research Laboratory (1994).
19. X. Wang, C. Lemieux, H. Faure, A note on Atanassov's discrepancy bound for the Halton sequence, Technical report, University of Waterloo, Canada (2008). Available at sas.uwaterloo.ca/stats_navigation/techreports/08techreports.shtml.

Applicability of Subsampling Bootstrap Methods in Markov Chain Monte Carlo

James M. Flegal

Abstract Markov chain Monte Carlo (MCMC) methods allow exploration of intractable probability distributions by constructing a Markov chain whose stationary distribution equals the desired distribution. The output from the Markov chain is typically used to estimate several features of the stationary distribution such as mean and variance parameters along with quantiles and so on. Unfortunately, most reported MCMC estimates do not include a clear notion of the associated uncertainty. For expectations one can assess the uncertainty by estimating the variance in an asymptotic normal distribution of the Monte Carlo error. For general functionals there is no such clear path. This article studies the applicability of subsampling bootstrap methods to assess the uncertainty in estimating general functionals from MCMC simulations.

1 Introduction

This article develops methods to evaluate the reliability of estimators constructed from Markov chain Monte Carlo (MCMC) simulations. MCMC uses computer-generated data to estimate some functional θ_π, where π is a probability distribution with support X. It has become a standard technique, especially for Bayesian inference, and the reliability of MCMC estimators has already been studied for cases where we are estimating an expected value [9, 13, 19]. Here, we investigate the applicability of subsampling bootstrap methods (SBM) for output analysis of an MCMC simulation. This work is appropriate for general functionals including expectations, quantiles and modes.

J.M. Flegal
University of California, Riverside, 92521, CA, USA
e-mail: jflegal@ucr.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_18, © Springer-Verlag Berlin Heidelberg 2012

The basic MCMC method entails constructing a Harris ergodic Markov chain $X = \{X_0, X_1, X_2, \ldots\}$ on X having invariant distribution π. The popularity of MCMC methods result from the ease with which an appropriate X can be simulated [4, 20, 23]. Suppose we simulate X for a finite number of steps, say n, and use the observed values to estimate θ_π with $\hat{\theta}_n$.

In practice the simulation is run sufficiently long until we have obtained an accurate estimate of θ_π. Unfortunately, we have no certain way to know when to terminate the simulation. At present, most analysts use convergence diagnostics for this purpose (for a review see [5]); although it is easily implemented, this method is mute about the quality of $\hat{\theta}_n$ as an estimate of θ_π. Moreover, diagnostics can introduce bias directly in to the estimates [6].

The approach advocated here will directly analyze output from an MCMC simulation to establish non-parametric or parametric confidence intervals for θ_π. There is already substantial research when θ_π is an expectation, but very little for general quantities.

Calculating and reporting an uncertainty estimate, or confidence interval, allows everyone to judge the reliability of the estimates. The main point is an uncertainty estimate should be reported along with the point estimate obtained from an MCMC experiment. This may seem obvious to most statisticians but this is not currently standard practice in MCMC [9, 13, 19].

Outside of toy examples, no matter how long our simulation, there will be an unknown *Monte Carlo error*, $\hat{\theta}_n - \theta_\pi$. While it is impossible to assess this error directly, we can estimate the error via a sampling distribution. That is, we need an asymptotic distribution for $\hat{\theta}_n$ obtained from a Markov chain simulation. Assume $\hat{\theta}_n$, properly normalized, has a limiting distribution J_π, specifically as $n \to \infty$

$$\tau_n \left(\hat{\theta}_n - \theta_\pi \right) \xrightarrow{d} J_\pi \tag{1}$$

where $\tau_n \to \infty$.

For general dependent sequences, there is a substantial amount of research about obtaining asymptotic distributions for a large variety of θ_π. These results are often applicable since the Markov chains in MCMC are special cases of strong mixing processes.

This article addresses how to estimate the uncertainty of $\hat{\theta}_n$ given a limiting distribution as at (1). Bootstrap methods may be appropriate for this task. Indeed, there is already sentiment that bootstrap methods used in stationary time series are appropriate for MCMC [1, 2, 7, 21]. However, my preliminary work [8] suggests that the SBM has superior computational and finite-sample properties.

The basic SBM provides a general approach to constructing asymptotically valid confidence intervals [22]. In short, SBM calculates the desired statistic over subsamples of the chain and then use these values to approximate the sampling distribution of θ_π. From the subsample values, one can construct a non-parametric confidence interval directly or estimate the unknown asymptotic variance of J_π and construct a parametric confidence interval.

The rest of this article is organized as follows. Section 2 overviews construction of non-parametric and parametric confidence intervals for general quantities θ_π via SBM. Section 3 examines the finite sample properties in a toy example and Sect. 4 illustrates the use of SBM in a realistic example to obtain uncertainty estimates for estimating quantiles.

2 Subsampling Bootstrap Methods

This section overviews SBM for constructing asymptotically valid confidence intervals of θ_π. Aside from a proposed diagnostic [14] and a brief summary for quantiles [11], there has been little investigation of SBM in MCMC. Nonetheless, SBM is widely applicable with only limited assumptions. The main requirement is that $\hat{\theta}_n$, properly normalized, has a limiting distribution as at (1).

SBM divides the simulation into overlapping subsamples of length b from the first n observations of X. In general, there are $n - b + 1$ subsamples for which we calculate the statistics over each subsample. Procedurally, we select a batch size b such that $b/n \to 0$, $\tau_b/\tau_n \to 0$, $\tau_b \to \infty$ and $b \to \infty$ as $n \to \infty$. If we let $\hat{\theta}_i^*$ for $i = 1, \ldots, n - b + 1$ denote the value of the statistic calculated from the ith batch, the assumptions on b imply as $n \to \infty$

$$\tau_b \left(\hat{\theta}_i^* - \theta_\pi \right) \xrightarrow{d} J_\pi \quad \text{for} \quad i = 1, \ldots, n - b + 1.$$

We can then use the values of $\hat{\theta}_i^*$ to approximate J_π and construct asymptotically valid inference procedures. Specifically, define the empirical distribution of the standardized $\hat{\theta}_i^*$s as

$$L_{n,b}(y) = \frac{1}{n - b + 1} \sum_{i=1}^{n-b+1} I \left\{ \tau_b \left(\hat{\theta}_i^* - \hat{\theta}_n \right) \leq y \right\}.$$

Further for $\alpha \in (0, 1)$ define

$$L_{n,b}^{-1}(1 - \alpha) = \inf\{y : L_{n,b}(y) \geq 1 - \alpha\}$$

and

$$J_\pi^{-1}(1 - \alpha) = \inf\{y : J_\pi(y) \geq 1 - \alpha\}.$$

Theorem 1. *Let X be a Harris ergodic Markov chain. Assume (1) and that $b/n \to 0$, $\tau_b/\tau_n \to 0$, $\tau_b \to \infty$ and $b \to \infty$ as $n \to \infty$.*

1. *If y is a continuity point of $J_\pi(\cdot)$, then $L_{n,b}(y) \to J_\pi(y)$ in probability.*
2. *If $J_\pi(\cdot)$ is continuous at $J_\pi^{-1}(1 - \alpha)$, then as $n \to \infty$*

$$Pr\left\{\tau_n\left(\hat{\theta}_n - \theta_\pi\right) \leq L^{-1}_{n,b}(1-\alpha)\right\} \to 1-\alpha.$$

Proof. Note that Assumption 4.2.1 of [22] holds under (1) and the fact that X possesses a unique invariant distribution. Then the proof is a direct result of Theorem 4.2.1 of [22] and the fact that Harris ergodic Markov chains are strongly mixing [18].

Theorem 1 provides a consistent estimate of the limiting law J_π for Harris ergodic Markov chains through the empirical distribution of $\hat{\theta}^*_i$. Hence a theoretically valid $(1-\alpha)100\%$ non-parametric interval can be expressed as

$$\left[\hat{\theta}_n - \tau_n^{-1}L^{-1}_{n,b}(1-\alpha/2), \quad \hat{\theta}_n - \tau_n^{-1}L^{-1}_{n,b}(\alpha/2)\right]. \tag{2}$$

Alternatively, one can also estimate the asymptotic variance [3, 22] using

$$\hat{\sigma}^2_{SBM} = \frac{\tau_b^2}{n-b+1}\sum_{i=1}^{n-b+1}\left(\hat{\theta}^*_i - \hat{\theta}_n\right)^2. \tag{3}$$

If J_π is Normal then a $(1-\alpha)100\%$ level parametric confidence interval can be obtained as

$$\left[\hat{\theta}_n - t_{n-b,\alpha/2}\tau_n^{-1}\hat{\sigma}_{SBM}, \quad \hat{\theta}_n + t_{n-b,\alpha/2}\tau_n^{-1}\hat{\sigma}_{SBM}\right]. \tag{4}$$

SBM is applicable for any $\hat{\theta}_n$ such that (1) holds and the rate of convergence τ_n is known as required in (2)–(4). Implementation requires selection of b, the subsample size. We will use the naive choice of $b_n = \lfloor n^{1/2} \rfloor$ in later examples. The following sections consider two common quantities where SBM is appropriate, expectations and quantiles.

2.1 Expectations

Consider estimating an expectation of π, that is

$$\theta_\pi = E_\pi g = \int_{\mathsf{X}} g(x)\pi(dx).$$

Suppose we use the observed values to estimate $E_\pi g$ with a sample average

$$\bar{g}_n = \frac{1}{n}\sum_{i=0}^{n-1} g(x_i).$$

The use of this estimator is justified through the Markov chain strong law of large numbers. Further assume a Markov chain CLT holds [18, 26], that is

$$\sqrt{n}(\bar{g}_n - E_\pi g) \overset{d}{\to} N(0, \sigma^2_\infty) \tag{5}$$

as $n \to \infty$ where $\sigma^2_\infty \in (0, \infty)$. Then we can use (2) or (4) to form non-parametric or parametric confidence intervals, respectively.

Alternatively, one can consider the overlapping batch means (OLBM) variance estimator [10]. As the name suggests, OLBM divides the simulation into overlapping batches of length b resulting in $n - b + 1$ batches for which $\bar{Y}_j(b) = b^{-1}\sum_{i=0}^{b-1} g(X_{j+i})$ for $j = 0, \dots, n - b$. Then the OLBM estimator of σ^2_∞ is

$$\hat{\sigma}^2_{OLBM} = \frac{nb}{(n-b)(n-b+1)} \sum_{j=0}^{n-b} (\bar{Y}_j(b) - \bar{g}_n)^2. \tag{6}$$

It is easy to show that (3) is asymptotically equivalent to (6).

2.2 Quantiles

It is routine when summarizing an MCMC experiment to include sample quantiles, especially in Bayesian applications. These are based on quantiles of the univariate marginal distributions associated with π. Let F be the marginal cumulative distribution function, then consider estimating the quantile function of F, i.e. the generalized inverse $F^{-1} : (0, 1) \mapsto \mathbb{R}$ given by

$$\theta_\pi = F^{-1}(q) = \inf\{y : F(y) \geq q\}.$$

We will say a sequence of quantile functions *converges weakly* to a limit quantile function, denoted $F_n^{-1} \rightsquigarrow F^{-1}$, if and only if $F_n^{-1}(t) \to F^{-1}(t)$ at every t where F^{-1} is continuous. Lemma 21.2 of [28] shows $F_n^{-1} \rightsquigarrow F^{-1}$ if and only if $F_n \rightsquigarrow F$. Thus we consider estimating F with the empirical distribution function defined as

$$\mathbb{F}_n(y) = \frac{1}{n} \sum_{i=1}^{n} I\{Y_i \leq y\},$$

where $Y = \{Y_1, \dots, Y_n\}$ is the observed univariate sample from F and I is the usual indicator function on $\mathbb{Z}_+$. The ergodic theorem gives pointwise convergence ($\mathbb{F}_n(y) \to F(y)$ for every y almost surely as $n \to \infty$) and the Glivenko-Cantelli theorem extends this to uniform convergence ($\sup_{y \in \mathbb{R}} |\mathbb{F}_n(y) - F(y)| \to 0$ almost surely as $n \to \infty$).

Letting $Y_{n(1)}, \dots, Y_{n(n)}$ be the order statistics of the sample, the empirical quantile function is given by:

$$\mathbb{F}_n^{-1} = Y_{n(j)} \quad \text{for} \quad q \in \left(\frac{j-1}{n}, \frac{j}{n} \right].$$

Often the empirical distribution function $\mathbb{F}_n$ and the empirical quantile function $\mathbb{F}_n^{-1}$ are directly used to estimate F and F^{-1}.

Construction of interval estimate of F^{-1} requires existence of a limiting distribution as at (1). We will assume a CLT exists for the Monte Carlo error [12], that is

$$\sqrt{n}\left(\mathbb{F}_n^{-1}(q) - F^{-1}(q) \right) \xrightarrow{d} \mathrm{N}(0, \sigma_\infty^2) \tag{7}$$

as $n \to \infty$ where $\sigma_\infty^2 \in (0, \infty)$. Then we can use (2) or (4) to form non-parametric or parametric confidence intervals respectively by setting $\hat{\theta}_i^*$ to the estimated quantile from the ith subsample.

3 Toy Example

Consider estimating the quantiles of an Exp(1) distribution, i.e. $f(x) = e^{-x} I(x > 0)$, using the methods outlined above. It is easy to show that $F^{-1}(q) = \log(1-q)^{-1}$, and simulation methods are not necessary; accordingly, we use the true values to evaluate the resulting coverage probability of the parametric and non-parametric intervals.

Monte Carlo sampling. SBM is also valid using i.i.d. draws from π, that is for Monte Carlo simulations. Here the subsamples need not be overlapping, hence there are $N := \binom{n}{b}$ subsamples. Calculation over N subsamples will often be computational extensive. Instead, a suitably large $N << \binom{n}{b}$ can be selected resulting in a estimate based on a large number of subsamples rather than all the subsamples.

Consider sampling from π using i.i.d. draws. For each simulation, with $n = 1\text{e}4$ iterations, CIs were calculated for $q \in \{.025, .1, .5, .9, .975\}$ based on $b \in \{100, 4000\}$. For both values of b, calculation of $\hat{\sigma}_{SBM}^2$ was based on $N = 1,000$ random subsamples rather than $\binom{n}{b}$ subsamples. This procedure was repeated 2,000 times to evaluate the resulting confidence intervals, see Table 1 for a summary of the simulation results.

For $b = 100$, the mean values of $\hat{\sigma}_{SBM}/\sigma_\infty^2$ are close to 1 for all values of q implying there is no systematic bias in the variance estimates. When $q \in \{.1, .5, .9\}$, the coverage probabilities are close to the nominal value of 0.95. For more extreme values of $q \in \{.025, .975\}$, the results are worse, which should not be surprising given $b = 100$. The use of non-parametric CIs at (2) show a similar trend, though the overall results are considerably worse.

Table 1 Coverage probabilities for Exp(1) example using i.i.d. sampler. Coverage probabilities reported have 0.95 nominal level with standard errors equal to $\sqrt{\hat{p}(1-\hat{p})/2{,}000} \leq 0.0082$.

	q	0.025	0.1	0.5	0.9	0.975
b =100	SBM	0.9705	0.9490	0.9480	0.9485	0.9400
	NP SBM	0.8595	0.9260	0.9415	0.9410	0.9210
b =4e3	SBM	0.8660	0.8690	0.8700	0.8715	0.8670
	NP SBM	0.8375	0.8575	0.8600	0.8575	0.8395

Table 2 Coverage probabilities for Exp(1) example using independence Metropolis sampler. Coverage probabilities reported have 0.95 nominal level with standard errors equal to $\sqrt{\hat{p}(1-\hat{p})/2{,}000} \leq 0.0109$.

	q	0.025	0.1	0.5	0.9	0.975
$\theta = 1/4$	SBM	0.9790	0.9530	0.9370	0.9380	0.9360
	NP SBM	0.7305	0.8795	0.9305	0.9425	0.9450
$\theta = 1/2$	SBM	0.9710	0.9495	0.9445	0.9385	0.9470
	NP SBM	0.8125	0.9215	0.9465	0.9415	0.9520
$\theta = 2$	SBM	0.9930	0.9905	0.9885	0.6145	0.1720
	NP SBM	0.8630	0.9120	0.8765	0.6295	0.1695

One may consider increasing b to improve the results for $q \in \{.025, .975\}$. However, if $b = 4,000$ without increasing n, the resulting coverage probabilities are significantly worse for both types of CIs (see Table 1). The simulations also show the mean value of $\hat{\sigma}_{SBM}/\sigma^2_\infty$ is less that 1, hence the variance estimates are biased down. Instead, as b increases, the overall simulation effort should also increase.

Rather than increasing b, it may be useful to consider different quantile estimates including continuous estimators [16] or a finite sampler correction [22]. Given our interest in MCMC, these were not considered here.

MCMC sampling. Consider sampling from π using an independence Metropolis sampler with an Exp(θ) proposal [19, 25, 27]. If $\theta = 1$ the sampler simply provides i.i.d. draws from π. The chain is geometrically ergodic if $0 < \theta < 1$ and sub-geometric (slower than geometric) if $\theta > 1$.

We calculated intervals for $q \in \{.025, .1, .5, .9, .975\}$; each chain contained $n = 1\text{e}4$ iterations and the procedure was repeated 2,000 times. The simulations began at $X_0 = 1$, with $\theta \in \{1/4, 1/2, 2\}$, and $b = 100$. Table 2 summarizes the results. For $\theta \in \{1/4, 1/2\}$ and $q \in \{.1, .5, .9, .975\}$, the coverage probabilities are close to the nominal value of 0.95. Increasing b would likely improve the results, but with a concurrent requirement for larger n. These limited results for parametric confidence intervals are very encouraging. In contrast, non-parametric CIs derived from (2) perform worse, especially for $q \in \{.025, .1\}$.

When $\theta = 2$, the chain is sub-geometric and it is unclear if $\sqrt{n}$-CLT holds as at (7). In fact, the independence sampler fails to have a $\sqrt{n}$-CLT at (5) for all suitably non-trial functions g when $\theta > 2$ [24, 27]. However, it is possible via SBM to obtain parametric and non-parametric CIs at (2) or (4) if one assumes a CLT with rate of convergence $\tau_n = \sqrt{n}$. The results from this simulation are also contained in Table 2.

We can see the coverage probabilities are close to the 0.95 nominal level for small quantiles, but this is likely because 0.95 is close to 1. In the case of large quantiles, the results are terrible, as low as 0.17. This example highlights the importance of obtaining a Markov chain CLT.

4 A Realistic Example

In this section, we consider the analysis of US government HMO data [15] under the following proposed model [17]. Let y_i denote the individual monthly premium of the ith HMO plan for $i = 1, \ldots, 341$ and consider a Bayesian version of the following frequentist model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i \tag{8}$$

where ϵ_i are i.i.d. $N\left(0, \lambda^{-1}\right)$, x_{i1} denotes the centered and scaled average expenses per admission in the state in which the ith HMO operates, and x_{i2} is an indicator for New England. (Specifically, if $\tilde{x}_{i1}$ are the original values and $\bar{x}_1$ is the overall average per admission then $x_{i1} = \left(\tilde{x}_{i1} - \bar{x}_1\right)/1,000$.)

Our analysis is based on the following Bayesian version of (8)

$$\begin{aligned} y|\beta, \lambda &\sim N_N\left(X\beta, \lambda^{-1} I_N\right) \\ \beta|\lambda &\sim N_3\left(b, B^{-1}\right) \\ \lambda &\sim \text{Gamma}\left(r_1, r_2\right) \end{aligned}$$

where $N = 341$, y is the 341×1 vector of individual premiums, $\beta = (\beta_0, \beta_1, \beta_2)$ is the vector of regression coefficients, and X is the 341×3 design matrix whose ith row is $x_i^T = (1, x_{i1}, x_{i2})$. (We will say $W \sim \text{Gamma}(a, b)$ if it has density proportional to $w^{a-1} e^{-bw}$ for $w > 0$.) This model requires specification of the hyper-parameters (r_1, r_2, b, B) which we assign based on estimates from the usual frequentist model [17]. Specifically, $r_1 = 3.122e-06$, $r_2 = 1.77e-03$,

$$b = \begin{pmatrix} 164.989 \\ 3.910 \\ 32.799 \end{pmatrix}, \text{and } B^{-1} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 36 \end{pmatrix}.$$

We will sample from $\pi\left(\beta, \lambda | y\right)$ using a two-component block Gibbs sampler requiring the following full conditionals

$$\begin{aligned} \lambda|\beta &\sim \text{Gamma}\left(r_1 + \frac{N}{2}, r_2 + \frac{1}{2} V(\beta)\right) \\ \beta|\lambda &\sim N_3\left(\left(\lambda X^T X + B\right)^{-1}\left(\lambda X^T y + Bb\right), \left(\lambda X^T X + B\right)^{-1}\right) \end{aligned}$$

Table 3 HMO parameter estimates with MCSEs.

	q	Estimate	MCSE
	0.05	163.40	1.05e−2
β_0	0.5	164.99	5.86e−3
	0.95	166.56	9.72e−3
	0.05	2.06	1.28e−2
β_1	0.5	3.92	7.24e−3
	0.95	5.79	1.19e−2
	0.05	25.86	4.61e−2
β_2	0.5	32.78	2.50e−2
	0.95	39.69	4.37e−2

where $V(\beta) = (y - X\beta)^T (y - X\beta)$ and we have suppressed the dependency on y. We consider the sampler which updates λ followed by β in each iteration, i.e. $(\beta', \lambda') \to (\beta', \lambda) \to (\beta, \lambda)$.

Our goal is estimating the median and reporting a 90% Bayesian credible region for each of the three marginal distributions. Denote the qth quantile associated with the marginal for β_j as $\phi_q^{(i)}$ for $j = 0, 1, 2$. Then the vector of parameters to be estimated is

$$\Phi = \left(\phi_{.05}^{(0)}, \phi_{.5}^{(0)}, \phi_{.95}^{(0)}, \phi_{.05}^{(1)}, \phi_{.5}^{(1)}, \phi_{.95}^{(1)}, \phi_{.05}^{(2)}, \phi_{.5}^{(2)}, \phi_{.95}^{(2)}\right).$$

Along with estimating Φ, we calculated the associated MCSEs using SBM. Table 3 summarizes estimates for Φ and MCSEs from 40,000 total iterations ($b_n = \lfloor 40,000^{1/2} \rfloor = 200$).

Acknowledgements I am grateful to Galin L. Jones and two anonymous referees for their constructive comments in preparing this article.

References

1. Patrice Bertail and Stéphan Clémençon. Regenerative block-bootstrap for Markov chains. *Bernoulli*, 12:689–712, 2006.
2. Peter Bühlmann. Bootstraps for time series. *Statistical Science*, 17:52–72, 2002.
3. Edward Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *The Annals of Statistics*, 14:1171–1179, 1986.
4. Ming-Hui Chen, Qi-Man Shao, and Joseph George Ibrahim. *Monte Carlo Methods in Bayesian Computation*. Springer-Verlag Inc, 2000.
5. Mary Kathryn Cowles and Bradley P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91:883–904, 1996.
6. Mary Kathryn Cowles, Gareth O. Roberts, and Jeffrey S. Rosenthal. Possible biases induced by MCMC convergence diagnostics. *Journal of Statistical Computing and Simulation*, 64:87–104, 1999.

7. Somnath Datta and William P. McCormick. Regeneration-based bootstrap for Markov chains. *The Canadian Journal of Statistics*, 21:181–193, 1993.
8. James M. Flegal. *Monte Carlo standard errors for Markov chain Monte Carlo*. PhD thesis, University of Minnesota, School of Statistics, 2008.
9. James M. Flegal, Murali Haran, and Galin L. Jones. Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*, 23:250–260, 2008.
10. James M. Flegal and Galin L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 38:1034–1070, 2010.
11. James M. Flegal and Galin L. Jones. Implementing Markov chain Monte Carlo: Estimating with confidence. In S.P. Brooks, A.E. Gelman, G.L. Jones, and X.L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Press, 2010.
12. James M. Flegal and Galin L. Jones. Quantile estimation via Markov chain Monte Carlo. *Work in progress*, 2011.
13. Charles J. Geyer. Practical Markov chain Monte Carlo (with discussion). *Statistical Science*, 7:473–511, 1992.
14. S. G. Giakoumatos, I. D. Vrontos, P. Dellaportas, and D. N. Politis. A Markov chain Monte Carlo convergence diagnostic using subsampling. *Journal of Computational and Graphical Statistics*, 8:431–451, 1999.
15. James S. Hodges. Some algebra and geometry for hierarchical models, applied to diagnostics (Disc: P521–536). *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 60:497–521, 1998.
16. Rob J. Hyndman and Yanan Fan. Sample quantiles in statistical packages. *The American Statistician*, 50:361–365, 1996.
17. Alicia A. Johnson and Galin L. Jones. Gibbs sampling for a Bayesian hierarchical general linear model. *Electronic Journal of Statistics*, 4:313–333, 2010.
18. Galin L. Jones. On the Markov chain central limit theorem. *Probability Surveys*, 1:299–320, 2004.
19. Galin L. Jones and James P. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16:312–334, 2001.
20. Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, 2001.
21. Dimitris N. Politis. The impact of bootstrap methods on time series analysis. *Statistical Science*, 18:219–230, 2003.
22. Dimitris N. Politis, Joseph P. Romano, and Michael Wolf. *Subsampling*. Springer-Verlag Inc, 1999.
23. Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 1999.
24. Gareth O. Roberts. A note on acceptance rate criteria for CLTs for Metropolis-Hastings algorithms. *Journal of Applied Probability*, 36:1210–1217, 1999.
25. Gareth O. Roberts and Jeffrey S. Rosenthal. Markov chain Monte Carlo: Some practical implications of theoretical results (with discussion). *Canadian Journal of Statistics*, 26:5–31, 1998.
26. Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.
27. Gareth O. Roberts and Jeffrey S. Rosenthal. Quantitative non-geometric convergence bounds for independence samplers. *Methodology and Computing in Applied Probability*, 13:391–403, 2011.
28. A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, New York, 1998.

QMC Computation of Confidence Intervals for a Sleep Performance Model

Alan Genz and Amber Smith

Abstract A five-dimensional Bayesian forecasting model for cognitive performance impairment during sleep deprivation is used to approximately determine confidence intervals for psychomotor vigilance task (PVT) prediction. Simulation is required to locate the boundary of a confidence region for the model pdf surface. Further simulation is then used to determine PVT lapse confidence intervals as a function of sleep deprivation time. Quasi-Monte Carlo simulation methods are constructed for the two types of simulations. The results from these simulations are compared with results from previous methods, which have used various combinations of grid-search, numerical optimization and simple Monte Carlo methods.

1 Introduction

A Bayesian forecasting model for cognitive performance impairment during sleep deprivation has been developed by Van Dongen and collaborators (see Van Dongen et al. [13, 15] and Van Dongen and Dinges [14]). This model uses an individual performance impairment function $P(\boldsymbol{\theta}, t) = P(\xi, \lambda, \eta, \nu, \phi, t)$ in the form

$$P(\xi, \lambda, \eta, \nu, \phi, t) = \xi e^{-\rho e^{\nu}(t-t_0)} + \gamma e^{\eta} \sum_{q=1}^{5} a_q \sin(\frac{2q\pi}{24}(t-\phi)) + \kappa + \lambda,$$

where t denotes time (in hours) and t_0 is the start time (i.e., time of awakening). The model contains several fixed parameters: ρ is the population-average buildup rate of sleep pressure across time awake; γ is the population-average amplitude of the circadian oscillation; κ determines the population-average basal performance capability;

A. Genz · A. Smith
Mathematics Department, Washington State University, Pullman, WA 99164-3113
e-mail: genz@math.wsu.edu; asmith@math.wsu.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_19, © Springer-Verlag Berlin Heidelberg 2012

and the coefficients a_q are the relative amplitudes of harmonics of the circadian oscillation (see Borbély and Achermann [1]). These fixed parameters were obtained from experimental data using many individuals (see Van Dongen et al. [15]), with $(t_0, \rho, \gamma, \kappa) = (7.5,\ 0.0350,\ 4.3,\ 29.7)$ and $\mathbf{a} = (0.97,\ 0.22,\ 0.07,\ 0.03,\ 0.001)$. There are five unknown model parameters ξ, ϕ, ν, η, and λ. The parameter ξ represents the specific individual's initial sleep pressure from prior sleep loss; ϕ determines the temporal alignment of the individual's circadian oscillation; ν is the buildup rate of sleep pressure across time awake for the individual; η is the amplitude of the circadian oscillation for the individual; and λ is the basal performance capability of the individual. The values of P for this model express cognitive performance in terms of the number of lapses (reaction times exceeding 500 ms) on a 10-min psychomotor vigilance task (PVT) [3].

If PVT performance measurements y_l at time points t_l for $l = 1, 2, \ldots, m$ are given for an individual, the likelihood function for this data is assumed to have the form

$$L(\xi, \lambda, \eta, \nu, \phi) = \prod_{l=1}^{m} p_N(y_l, P(\xi, \lambda, \eta, \nu, \phi, t_l), \sigma_L^2),$$

where $p_N(y, \mu, \sigma^2)$ denotes the standard univariate normal pdf with mean μ and standard deviation σ. The model also uses zero mean univariate normal priors for the variables ν, η, and λ with respective variances σ_ν^2, σ_η^2 and σ_λ^2. The variance values $\sigma_L^2, \sigma_\nu^2, \sigma_\eta^2, \sigma_\lambda^2 = (77.6, 1.15, 0.294, 36.2)$ that we use in this paper were also determined using averages from many individuals (see Van Dongen et al. [15]). The posterior probability density function for the individual performance impairment model is then given by

$$f(\boldsymbol{\theta}) \equiv f(\xi, \lambda, \eta, \nu, \phi) = cL(\xi, \lambda, \eta, \nu, \phi) p_N(\nu, 0, \sigma_\nu^2) p_N(\eta, 0, \sigma_\eta^2) p_N(\lambda, 0, \sigma_\lambda^2),$$

where c is a normalization constant.

The primary computational task when using the model is to find the smallest region in the multidimensional parameter space

$$S = \{\boldsymbol{\theta} = (\xi, \lambda, \eta, \nu, \phi) | (\xi, \lambda, \eta, \nu, \phi) \in (-\infty, 0] \times (-\infty, \infty)^3 \times [0, 24]\}$$

that captures a required percentage (e.g., 95%) of the (hyper)volume under the posterior pdf. To be more precise, given an α with $0 \le \alpha \le 1$, we define the *confidence region* R_α to be the smallest (in a sense to be specified later) subset of S satisfying

$$1 - \alpha = \int_{R_\alpha} f(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

After the determination of R_α, the future performance of the individual can be estimated by evaluating the performance function $P(\boldsymbol{\theta}, t)$ over R_α at a selected future time t. The purpose of this paper is to compare the use of Monte Carlo (MC) and Quasi-Monte Carlo (QMC) methods for these computational tasks.

2 Determination of the Confidence Region

2.1 Safe Height Approximation

The first step in our algorithm for the determination of R_α is the computation of the normalization constant c. This requires the evaluation of the five-dimensional integral

$$1/c \equiv C = \int_{-\infty}^{0}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{24} L(\xi,\lambda,\eta,\nu,\phi)p_N(\nu,0,\sigma_\nu^2)p_N(\eta,0,\sigma_\eta^2)p_N(\lambda,0,\sigma_\lambda^2)d\xi d\nu d\eta d\lambda d\phi,$$

which has the explicit form

$$C = \frac{1}{(2\pi)^{\frac{m+3}{2}}(\sigma_\nu\sigma_\eta\sigma_\lambda\sigma_L)}\int_{-\infty}^{0}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{24} e^{-\frac{\nu^2}{2\sigma_\nu^2}-\frac{\eta^2}{2\sigma_\eta^2}-\frac{\lambda^2}{2\sigma_\lambda^2}} e^{-\frac{\sum_{l=1}^{m}\left(\xi e^{-\rho e^{\nu}(t-t_0)}+\gamma e^{\eta}\sum_{q=1}^{5}a_q\sin(\frac{2q\pi}{24}(t-\phi))+\kappa+\lambda-y_l\right)^2}{2\sigma_L^2}} d\phi d\nu d\eta d\lambda d\xi \tag{1}$$

The boundary delineating the smallest confidence region for a multidimensional, continuous pdf always projects to a level (i.e., fixed-height) contour on the surface of the pdf (see Box and Tiao [2]; Tanner [12]). We define the *safe height*, denoted by h_α, to be the value of the pdf $f(\boldsymbol{\theta})$ along this level this contour, so that the confidence region R_α is implicitly defined by the condition $f(\boldsymbol{\theta}) \geq h_\alpha$.

We will consider the use of several numerical integration methods to estimate C. If C is approximated using an equal-weight numerical integration method in the form

$$C \approx \hat{C} = \frac{W}{N}\sum_{i=1}^{N} H(\boldsymbol{\theta}_i),$$

where

$$H(\boldsymbol{\theta}) \equiv H(\xi,\lambda,\eta,\nu,\phi) = L(\xi,\lambda,\eta,\nu,\phi)p_N(\nu,0,\sigma_\nu^2)p_N(\eta,0,\sigma_\eta^2)p_N(\lambda,0,\sigma_\lambda^2)$$

is the unnormalized posterior pdf and W is the integration domain volume, then an approximate value for h_α can be determined by selecting the smallest $H(\boldsymbol{\theta}_i)$ value in the set which contains the largest $100(1-\alpha)\%$ of the $H(\boldsymbol{\theta}_i)$ values. To be more precise, if we let $H(\boldsymbol{\theta}_{(i)})$ be the $H(\boldsymbol{\theta}_i)$ values sorted in ascending order and define

$$H_\alpha = H(\boldsymbol{\theta}_{(i^*)}), \text{ with } i^* = \lceil \alpha N \rceil,$$

then we can approximate the safe height h_α using

$$h_\alpha \approx \hat{h}_{\alpha,N} \equiv WH_\alpha/N. \tag{2}$$

Given $\hat{h}_{\alpha,N}$, we can also (approximately) determine a set of points from the confidence region R_α. We denote these *confidence* sets by $\hat{R}_{\alpha,N}$, with

$$\hat{R}_{\alpha,N} = \{\boldsymbol{\theta}_{(i)} \mid i \geq i^*\}.$$

This set contains the points which determine $100(1-\alpha)\%$ of the volume for the value of $\hat{C}_N$. These points can be saved and used for future performance prediction.

2.2 MC and QMC Integration Methods

The simplest numerical integration method for estimating C uses a crude Monte-Carlo method on a truncated integration domain. This was considered in Smith et al. [10], where domains in the form

$$\hat{S} = [c_1, d_1] \times [c_2, d_2] \times [c_3, d_3] \times [c_5, d_4] \times [c_5, d_5],$$

with all limits finite, and $d_1 = 0$, $[c_5, d_5] = [0, 24]$, were used. The unspecified limits were determined after investigating the decay of $H(\boldsymbol{\theta})$ for large values of $-\xi$, $\pm\nu$, $\pm\eta$ and $\pm\lambda$. Then

$$C \approx \int_{\boldsymbol{\theta}\in\hat{S}} H(\boldsymbol{\theta})d\boldsymbol{\theta} = W\int_0^1\int_0^1\int_0^1\int_0^1\int_0^1 H(\mathbf{c} + (\mathbf{d}-\mathbf{c})\cdot\mathbf{x})d\mathbf{x},$$

with $W = \prod_{k=1}^5 (d_k - c_k)$, and "$\cdot$" denotes componentwise multiplication. A Monte Carlo (MC) estimate for C is

$$\hat{C}_N = \frac{W}{N}\sum_{i1}^{N} H(\mathbf{c} + (\mathbf{d}-\mathbf{c})\cdot\mathbf{x}_i), \tag{3}$$

given $\mathbf{x}_i$'s with uniform random [0,1] components ($x_{ki} \sim U(0,1)$). Associated with these approximations are error estimates which can be obtained using the standard errors (see Fishman [6])

$$E_N = \Big(\frac{1}{N(N-1)}\sum_{i=1}^{N}(WH(\mathbf{c} + (\mathbf{d}-\mathbf{c})\cdot\mathbf{x}_i) - \hat{C}_N)^2\Big)^{\frac{1}{2}}.$$

These quantities are typically scaled by 3 to give approximate 99% confidence.

MC methods using N points have errors that are typically $O(1/N^{\frac{1}{2}})$, a convergence rate which is too slow for many problems. An alternative is to use quasi-Monte Carlo (QMC) methods (see Fox [7]), with asymptotic errors which can be approximately $O(1/N)$ for N points.

A typical N-point QMC approximation has the same form as (3), but the QMC points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are selected to provide a more uniform distribution than MC points. The simplest QMC sequences have the form

$$\mathbf{x}_i = \{i\ \mathbf{Z}\},$$

where $\mathbf{Z}$ is an appropriately chosen *generating* vector, and $\{\mathbf{T}\}$ denotes the vector obtained from $\mathbf{T}$ by taking, for each component, the fractional part belonging to $[0, 1]$. Two important classes of these QMC point sets are *Kronecker* and *lattice* sequences. Kronecker sequences (see Drmota and Tichy, [4], and also Fang and Wang, [5]) are point sets where the components of $\mathbf{Z}$ are irrational and linearly independent over the rational numbers. One simple Kronecker sequence choice for $\mathbf{Z}$ has $Z_i = \sqrt{p_i}$, with $p_i = ith$ prime (often referred to as the "Richtmyer" sequence). Lattice sequence generating vectors are vectors where $N\mathbf{Z}$ is an integer vector with ("good lattice") components chosen to minimize an appropriately chosen error measure for specified classes of integrands (see, for example Sloan and Joe [11] for more details). The paper by Nuyens and Cools [9] describes a method for the efficient determination of a good lattice vector, given N and the number of dimensions for $\mathbf{x}$. Many other QMC point sequences (e.g., Halton, Hammersley, Sobol and other digital-net sequences, see Drmota and Tichy, [4]) have also been studied, with similar asymptotic convergence properties.

Error estimates for a QMC $\hat{C}_N$ estimate can be computed if the QMC method is randomized. A simple method for randomization uses random shifts of a selected set (or batch) of QMC points to provide the QMC approximations in the form

$$\hat{C}_N(\mathbf{u}) = \frac{W}{N}\sum_{i=1}^{N} H(\mathbf{c} + (\mathbf{d} - \mathbf{c}) \cdot (\{\mathbf{x}_i + \mathbf{u}\})),$$

where $\mathbf{u}$ has independent random $U(0, 1)$ components. An unbiased randomized QMC (*RQMC*) approximation for C is then given by

$$\hat{C}_{N,K} = \frac{1}{K}\sum_{k=1}^{K} \hat{C}_N(\mathbf{u}_k) \tag{4}$$

with standard error

$$E_{N,K} = \Big(\frac{1}{K(K-1)}\sum_{k=1}^{K}(\hat{C}_N(\mathbf{u}_k) - \hat{C}_{N,K})^2\Big)^{\frac{1}{2}}. \tag{5}$$

For these approximations, K is usually chosen to be small (e.g., $K = 10$) relative to N, and N is increased to most efficiently reduce the error in $\hat{C}_{K,N}$. The $E_{K,N}$ quantities are typically scaled by 3 to give approximate 99% confidence.

If we use $\hat{h}_{\alpha,N}(\mathbf{u})$ to denote the approximate safe height determined (from (2)) using a $\mathbf{u}$ randomly shifted QMC point set, then averages of these values in the form

$$\hat{h}_{\alpha,N,K} = \frac{1}{K}\sum_{k=1}^{K} \hat{h}_{\alpha,N}(\mathbf{u}_k)$$

are RQMC estimates for h_α, and standard errors can also be computed for these estimates. The associated confidence sets, denoted by $\hat{R}_{\alpha,N}(\mathbf{u}_k)$, can be saved and used for future performance prediction.

2.3 Some Numerical Tests

The data for the tests described in this section is taken from [15] (for individual C) where 24 past performance measurements $\{(t_l, y_l)|l = 1,\ldots,24\}$ were given for 48 h of total sleep deprivation. For the particular individual that we consider, $t_l = 5.5 + 2l, l = 1,\ldots,24$, and

$$\mathbf{y} = (8\ 17\ 19\ 19\ 13\ 15\ 11\ 22\ 9\ 33\ 24\ 27\ 34\ 36\ 25\ 31\ 39\ 31\ 38\ 46\ 39\ 34\ 27\ 46).$$

With this data, the maximum (*mode*) for the posterior $H(\boldsymbol{\theta})$ occurs approximately (using the Matlab constrained minimization function *fmincon*, applied to $-\log(H)$) at $\boldsymbol{\theta} \equiv \boldsymbol{\mu} = (-32.725,\ 8.2011,\ -.15275,\ .48695,\ 4.4711)$. We first considered the truncated domain

$$\hat{S} = [-60, 0] \times [-20, 40] \times [-4, 4] \times [-3, 3] \times [0, 24]$$

which was used by Smith et al. [10], based on investigation of the rates of decrease in $H(\boldsymbol{\theta})$ for large values of $-\xi$, $\pm\nu$, $\pm\eta$ and $\pm\lambda$. Table 1 shows some results for MC and two RQMC methods RQMCK and RQMCL, (using (4) and (5) with $K = 10$, and $3 \times E_{N,K}$ used for the **Error** columns). For the MC results, the same $K = 10$, batching strategy was used to compute the $\hat{C}$ and $\hat{h}$ approximations and errors. The RQMCK method used Richtmyer (square roots of primes) generators and the RQMCL method used lattice rule generators computed using the Nuyens-Cools CBC algorithm ([9], with all weights = 1). These results show the superiority of the QMC methods, particularly the lattice rules. Note: the $\hat{C}$ approximations (and errors) in Table 1 and the other Tables in this paper have all been scaled by $(2\pi)^{\frac{m+3}{2}}$ (from the posterior pdf denominator) to avoid tiny values in the Tables.

We also studied several reparameterizations based on standardizing transformations in the form $\boldsymbol{\theta}(\mathbf{x}) = \boldsymbol{\mu} +, L\mathbf{y}$, where $\boldsymbol{\mu}$ is posterior mode, L is the lower

Table 1 Computation of $\hat{C}$ using MC and RQMC methods.

NK	MC	Error	RQMCK	Error	RQMCL	Error
100,000	.30389	.121680	.31763	.036703	.30878	.016796
200,000	.31481	.074530	.31050	.036681	.29962	.036017
400,000	.28589	.041518	.29384	.030151	.29163	.026106
800,000	.29615	.027958	.29406	.017439	.30077	.001658
$\hat{h}_{\alpha,80000,10}$	1.67e-6	2.9e-7	1.64e-6	1.1e-7	1.74e-6	6e-8

triangular Cholesky factor for the posterior covariance matrix $\boldsymbol{\Sigma}$ ($\boldsymbol{\Sigma} = LL^t$), given by $\boldsymbol{\Sigma} = G^{-1}$, when G is the Hessian matrix for $-\log(H(\boldsymbol{\theta}))$ at $\boldsymbol{\theta} = \boldsymbol{\mu}$. Note: G can easily be approximated with sufficient accuracy using standard second difference approximations to the second order partial derivatives for G. This type of reparameterization is often used with Bayesian analysis of posterior densities which have a dominant peak [8], and then a multivariate normal model for $H(\boldsymbol{\theta}(\mathbf{y}))$ with $H(\boldsymbol{\theta}(\mathbf{y})) \sim e^{-\mathbf{y}^t\mathbf{y}/2}$ is often used as a basis for importance sampling or related integration methods. However, $H(\boldsymbol{\theta})$ is (slowly varying) periodic (not decaying) in the ϕ variable, so a full 5-variable model of this type is not directly applicable. We further studied the behavior of H by computing the mode and $\boldsymbol{\Sigma}$ for several different fixed values of $\phi \in [0, 24]$, and found that the $(\xi, \lambda, \eta, \nu)$ components of the mode and the corresponding 4×4 $\boldsymbol{\Sigma}$'s did not change significantly as ϕ varies $\in [0, 24]$. So we focused on the reparameterization $\boldsymbol{\theta}(\mathbf{y}) = \boldsymbol{\mu} + L\mathbf{y}$ where L is the lower triangular Cholesky factor for the $\boldsymbol{\Sigma}$ determined from the Hessian of $-\log(H(\boldsymbol{\theta}))$ with $\phi = 4.4711$ fixed; then

$$L \approx \begin{bmatrix} 6.3834 & 0 & 0 & 0 & 0 \\ -2.1172 & 3.1661 & 0 & 0 & 0 \\ -.092616 & -.03576 & .40832 & 0 & 0 \\ -.000407 & -.27182 & .01002 & .17602 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

With this reparameterization C is given by

$$\begin{aligned} C &= |L| \int_{-\infty}^{-\mu_1/l_{11}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{0}^{24} H(\boldsymbol{\mu} + L\mathbf{y}) d\mathbf{y} \\ &\approx |L| \int_{-D}^{-\mu_1/l_{11}} \int_{-D}^{D} \int_{-D}^{D} \int_{-D}^{D} \int_{0}^{24} H(\boldsymbol{\mu} + L\mathbf{y}) d\mathbf{y} \end{aligned}$$

where $|L| = \det(L) = \prod_{k=1}^{5} l_{kk}$, D is a selected cutoff value, and $d\mathbf{y} = \prod_{k=5}^{1} dy_i$. The upper y_1 limit μ_1/l_{11} corresponds to $\xi = \mu_1 + l_{11}y_1 = 0$.

Table 2 shows some results for the MC, RQMCK and RQMCL methods, with this reparameterization, followed by the transformation $\mathbf{y} = D\mathbf{x}$ to $\mathbf{x} \in [0, 1]^5$

Table 2 Standardized $\hat{C}$ computation with MC and RQMC methods.

NK	MC	Error	RQMCK	Error	RQMCL	Error
100,000	.28809	.028872	.29858	.0489840	.30135	.0066589
200,000	.30396	.026594	.30069	.0029917	.29955	.0021110
400,000	.29539	.022879	.30215	.0022066	.30275	.0035099
800,000	.29797	.010646	.30025	.0009061	.30035	.0000884
$\hat{h}_{\alpha,80000,10}$	1.69e-6	9e-8	1.74e-6	4e-8	1.73e-6	3e-8

variables, with $D = 6$. These results are generally much more accurate than the unstandardized results and also show the superiority of the QMC methods.

We also studied the use of further transformations of the $(\xi, \lambda, \eta, \nu)$ variables based on the multivariate normal model $H(\boldsymbol{\theta}(\mathbf{y})) \sim e^{-\mathbf{y}^t\mathbf{y}/2}$ (and other related statistical distribution models) with, for example, $x_i = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y_i} e^{-t_i^2/2}dt, i = 1, 2, 3, 4$. Results using these models as a basis for transforming the $(\xi, \lambda, \eta, \nu)$ variables resulted in less accurate $\hat{C}$ approximations, given the same amount of computational work (NK values), so we do not report the details for those results here.

We studied one additional transformation. With this transformation, we first transform ξ to a $w \in (-\infty, \infty)$ variable using $\xi = -e^w$, followed by a standardizing transformation computed for $H(-e^w, \lambda, \eta, \nu, \phi)e^w$, with ϕ free to determine $\boldsymbol{\mu}$, and fixed at μ_5 to determine $\boldsymbol{\Sigma}$ and L. The extra e^w factor multiplying the original posterior is needed because the integration of the transformed posterior uses $d\xi = -e^w dw$. After the standardizing transformation, we can then use a "spherical-radial" transformation, with only one unbounded variable.

The standardizing transformation parameters were determined to be

$$\boldsymbol{\mu} \approx (3.5261,\ 8.6272,\ -.13415,\ .48632,\ 4.5866),$$

(note $-e^{\mu_1} \approx -34$ corresponding to previous μ_1), and

$$L \approx \begin{bmatrix} .18491 & 0 & 0 & 0 & 0 \\ 2.0887 & 3.2427 & 0 & 0 & 0 \\ .094273 & -.037663 & .40780 & 0 & 0 \\ .000021 & -.27323 & .00993 & .17107 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then

$$C = |L| \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{24} H(\boldsymbol{\theta}(\mathbf{w}(\mathbf{y})))e^{w_1(y_1)}d\mathbf{y}$$

with $\boldsymbol{\theta}(\mathbf{w}) = (-e^{w_1}, w_2, w_3, w_4, w_5)$, and $\mathbf{w}(\mathbf{y}) = \boldsymbol{\mu} + L\mathbf{y}$. The new transformation is completed with a spherical-radial (SR) transformation for the first four $\mathbf{y}$ components $(y_1, y_2, y_3, y_4) = r(z_1, z_2, z_3, z_4)$ with $r \in [0, \infty)$ and $\mathbf{z} \in U_4$, the surface of the unit 4-sphere,

Table 3 Standardized SR $\hat{C}$ Computation with MC and RQMC Methods.

NK	MC	Error	RQMCK	Error	RQMCL	Error
100,000	.30330	.007941	.30023	.0006252	.30013	.0002710
200,000	.29894	.005011	.29993	.0003043	.29995	.0001891
400,000	.29991	.002590	.30003	.0003585	.30003	.0000604
800,000	.29973	.002574	.29999	.0001324	.30002	.0000358
$\hat{h}_{\alpha,80000,10}$	.19878	.00240	.19917	.00148	.20020	.00084

$$U_4 = \{\, \mathbf{z} \mid z_1^2 + z_2^2 + z_3^2 + z_4^2 = 1 \,\}.$$

Now

$$\begin{aligned} C &= |L| \int_0^\infty \int_{||\mathbf{z}||_2=1} \int_0^{24} H(\boldsymbol{\theta}(\mathbf{w}(\mathbf{y}(\mathbf{z}))))e^{w_1(y_1(\mathbf{z}))} r^3 dr d\mathbf{z} dy_5 \\ &\approx |L| \int_0^D \int_{||\mathbf{z}||_2=1} \int_0^{24} H(\boldsymbol{\theta}(\mathbf{w}(\mathbf{y}(\mathbf{z}))))e^{w_1(y_1(\mathbf{z}))} r^3 dr d\mathbf{z} dy_5, \end{aligned}$$

where $d\mathbf{z}$ is the U_4 surface measure, the r^3 term comes from the Jacobian of the transformation from (y_1, y_2, y_3, y_4) to $r\mathbf{z}$ and the final approximation uses a cutoff value of D to replace the ∞ upper r limit.

MC and QMC methods require $\mathbf{x} \in [0, 1]^5$ variables, so we used the transformation (see [5])

$$(z_1, z_2, z_3, z_4) = \big(\sqrt{x_1}(\sin(2\pi x_2), \cos(2\pi x_2)), \sqrt{1 - x_1}(\sin(2\pi x_3), \cos(2\pi x_3))\big),$$

with constant Jacobian $2\pi^2$, $r = Dx_4$ and $y_5 = 24x_5$, so that

$$\hat{C}_N = \frac{24D^4|L|2\pi^2}{N} \sum_{i=1}^{N} H(\boldsymbol{\theta}(\mathbf{w}(\mathbf{y}(\mathbf{z}(\mathbf{x}_i)))))e^{w_1(y_1(\mathbf{z}(\mathbf{x}_i)))} x_4^3,$$

can be used to provide MC or QMC approximations to C, as determined by the choice of the $\mathbf{x}_i$ points.

Table 3 shows some results for the MC, RQMCK and RQMCL methods, with this reparameterization, followed by a transformation to $\mathbf{x} \in [0, 1]^5$ variables, for $D = 6$. These results are even more accurate than the previous standardized results and also show the superiority of the QMC methods. The h_α approximations here differ from the ones in the previous two Tables because the standardized SR transformed posterior density, including Jacobian factors, results in a different set of values for the $100(1 - \alpha)$ percentile computation. These R_α confidence sets, associated with h_α's, can still be used for performance prediction. The Table 3 accuracy levels obtained using the spherical-radial transformed lattice-rule QMC combination are not typically needed for practical performance prediction. Further

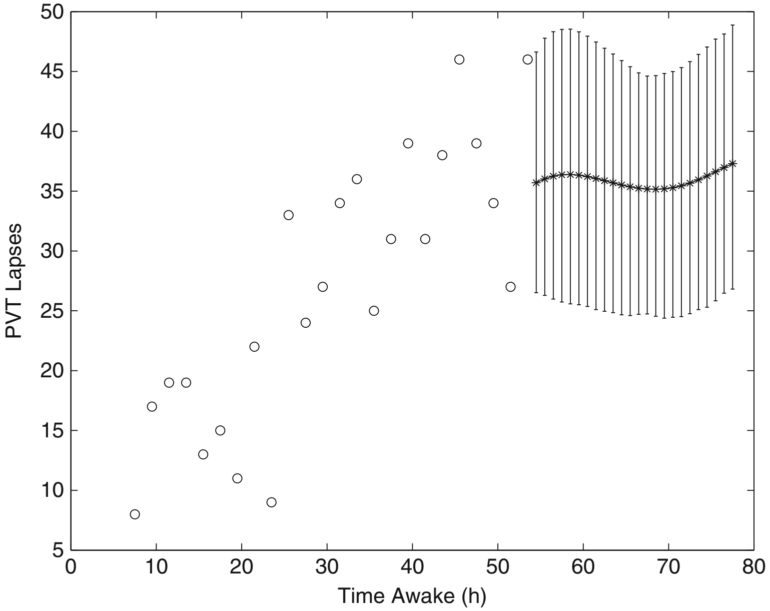


Fig. 1 Predicted Cognitive Performance (PVT Lapse) with Confidence Intervals: 'o' points denote individual data values; '*' points denote predicted values.

tests with this combination have shown that sufficient accuracy is obtained with values for $NK \approx 10,000$, allowing these computations to be completed in less than a second using Matlab on a laptop computing platform.

3 Performance Prediction Results

The predicted performance can be computed from the data collected during the computation of C, where we also compute K $\hat{R}_{\alpha,N}$ sets, the sets of $\boldsymbol{\theta}$ points used for $\hat{C}_N$ that are inside the approximate safe height region. Given a set $\hat{R}_{\alpha,N}$ containing M $\boldsymbol{\theta}_i$ points, and a future time t, we compute predicted average $\hat{P}_N(t)$, minimum $\underline{P}_N(t)$, and maximum $\overline{P}_N(t)$ performance, using

$$\hat{P}_N(t) = \frac{1}{M} \sum_{\boldsymbol{\theta}_i \in \hat{R}_{\alpha,N}} P(\boldsymbol{\theta}_i, t), \ \underline{P}_N(t) = \min_{\boldsymbol{\theta}_i \in \hat{R}_{\alpha,N}} P(\boldsymbol{\theta}_i, t), \ \overline{P}_N(t) = \max_{\boldsymbol{\theta}_i \in \hat{R}_{\alpha,N}} P(\boldsymbol{\theta}_i, t).$$

In Fig. 1 we show the PVT lapse data values for individual C from [15] followed by the average (over K $\hat{R}_{\alpha,N}$ sets) predicted $\hat{P}_N(t)$ values every hour for 24 additional hours; for each $\hat{P}_N(t)$ value, the error bars computed using $\underline{P}_N(t)$ and $\overline{P}_N(t)$ values provide confidence intervals. The data for the ($\hat{P}_N(t)$, confidence interval) values in Fig. 1 were collected during computations for the $NK = 100{,}000$ for Table 3. These results are similar to those shown in [10].

4 Concluding Remarks

In this paper we considered a two-part algorithm to efficiently estimate confidence intervals for a Bayesian model for sleep performance predictions that can be formulated in terms of a performance model $P(\boldsymbol{\theta}, t)$ for a 5-dimensional parameter space $\boldsymbol{\theta}$ described by a continuous posterior pdf $f(\boldsymbol{\theta})$ constructed using past performance data from a particular individual. The major part of the algorithm deals with finding the smallest region R_α that captures the $100(1-\alpha)$ percentage of the (hyper)area under the pdf surface. This boundary projects to a level contour on the surface of the pdf, with height h_α, which can be approximated during the computation for the normalizing constant c for $f(\boldsymbol{\theta})$. The simulation points, used for the computation of c, which are inside R_α can then be used to compute average $P(\boldsymbol{\theta}, t)$ values at future times, with associated confidence intervals.

We have shown that the use of QMC simulation points combined with an appropriate transformation of the parameter space can significantly increase the accuracy of the computations for c, h_α and future performance predictions. The methods described here some of the more computationally intensive methods considered previously for this problem involving spline approximations, numerical optimization and grid searches [10, 15]. These new methods make it possible to provide confidence intervals for Bayesian model predictions in real time.

References

1. Borbély, A. A., and Achermann, P., 'Sleep homeostasis and models of sleep regulation'. *J. Biol. Rhythms* **14**, pp. 557–568, 1999.
2. Box, G. E. P., and Tiao, G. C., *Bayesian Inference in Statistical Analysis*, Wiley-Interscience, New York, p. 123, 1992.
3. Dorrian, J., Rogers, N. l., and Dinges, D. F., 'Psychomotor Vigilance Performance: Neurocognitive Assay Sensitive to Sleep Loss', pp. 39–70 in Kushida, C. A. (ed.) *Sleep Deprivation: Clinical Issues, Pharmacology, and Sleep Loss Effects*, Marcel Dekker, New York, 2005.
4. Drmota, M. and Tichy, R. F., *Sequences, Discrepancies and Applications*, Lecture Notes in Mathematics 1651, Springer-Verlag, New York, 1997.
5. Fang, K.-T., and Wang, Y., *Number-Theoretic Methods in Statistics*, Chapman and Hall, London, pp. 26–32, 1994.
6. Fishman, G. S., *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, 1996.
7. Fox, B. L, *Strategies for Quasi-Monte Carlo* (International Series in Operations Research & Management Science, 22), Kluwer Academic Publishers, 1999.
8. Genz, A., and Kass, R., 'Subregion Adaptive Integration of Functions Having a Dominant Peak', *J. Comp. Graph. Stat.* **6**, pp. 92–111, 1997.

9. Nuyens, D., and Cools, R., 'Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces', *Math. Comp* **75**, pp. 903–920, 2006.
10. Smith, A., Genz, A., Freiberger, D. M., Belenky, G., and Van Dongen, H. P. A., 'Efficient computation of confidence intervals for Bayesian model predictions based on multidimensional parameter space', *Methods in Enzymology #454: Computer Methods*, M. Johnson and L. Brand (Eds), Elsevier, pp. 214–230, 2009.
11. Sloan, I. H., and Joe, S., *Lattice Methods for Multiple Integration*, Oxford University Press, Oxford, 1994.
12. Tanner, M. A., *Tools for Statistical Inference*, 2^{nd} Ed., Springer-Verlag, New York, 1993.
13. Van Dongen, H. P. A., Baynard, M. D., Maislin, G., and Dinges, D. F., 'Systematic interindividual differences in neurobehavioral impairment from sleep loss: Evidence of trait-like differential vulnerability', *Sleep* **27**, pp. 423–433, 2004.
14. Van Dongen, H. P. A., and Dinges, D. F., 'Sleep, Circadian rhythms, and Psychomotor Vigilance', *Clin. Sports Med.* **24**, pp. 237–249, 2005.
15. Van Dongen, H. P. A., Mott, C. G., Huang, J.-K., Mollicone, D. J., McKenzie, F. D., and Dinges, D. F., 'Optimization of biomathematical model predictions for cognitive performance impairment in individuals: Accounting for unknown traits and uncertain states in homeostatic and circadian processes', *Sleep* **30**, pp. 1129–1143, 2007.

Options Pricing for Several Maturities in a Jump-Diffusion Model

Anatoly Gormin and Yuri Kashtanov

Abstract Estimators for options prices with different maturities are constructed on the same trajectories of the underlying asset price process. The weighted sum of their variances (the weighted variance) is chosen as a criterion of minimization. Optimal estimators with minimal weighted variance are pointed out in the case of a jump-diffusion model. The efficiency of the constructed estimators is discussed and illustrated on particular examples.

1 Introduction

Monte Carlo calculations are useful for options pricing, especially in multidimensional models since the rate of convergence is independent of the dimension. Variance reduction for such calculations has been examined in many works; for example, in [3, 4, 11, 12] the authors use the methods of importance sampling and control variates to reduce the variance of particular option estimators.

It is often necessary to calculate option prices for different contract parameters such as strike, maturity, etc. In our previous articles, we already considered the problem of effective estimation of several option prices in a diffusion model [7, 8] and a model with jumps [9]. It was noticed in [7] that the case of several maturities dates has some specific features; in the present paper we concentrate on this problem.

Model description. Let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space, $p(dt, dy)$ is a Poisson measure on $[0, T) \times E$, $E \subset \mathbb{R}^d$ with intensity measure $\nu(dt, dy) = \lambda m(dy)dt$, where λ is a constant, $m(dy)$ is a probability measure on a measurable space $(E, \mathscr{E})$,

A. Gormin · Y. Kashtanov
Faculty of Mathematics and Mechanics, Department of Statistical Simulation, Saint-Petersburg State University, 198504 Saint-Petersburg, Russia
e-mail: Anatoliy.Gormin@pobox.spbu.ru; Yuri.Kashtanov@paloma.spbu.ru

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_20, © Springer-Verlag Berlin Heidelberg 2012

$m(E) = 1$. Denote by $\tilde{p}(dt, dy) = p(dt, dy) - \nu(dt, dy)$ the compensated version of $p(dt, dy)$. Let the underlying asset price process $X_t = (X_t^1, \ldots, X_t^d)$ satisfies the following SDE

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t + \int_E \gamma(t, X_{t-}, y)p(dt, dy), \qquad (1)$$

where W_t is a d-dimensional standard Brownian motion. Let a filtration $\mathbb{F} = (\mathscr{F}_t)_{0 \leqslant t \leqslant T}$ be a natural filtration generated by the process W_t and the Poisson measure $p(dt, dy)$. Suppose that the coefficient functions $a : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$, $b : [0, T] \times \mathbb{R}^d \to \mathbb{R}^{d \times d}$, $\gamma : [0, T] \times \mathbb{R}^d \times E \to \mathbb{R}^d$ satisfy sufficient conditions for the existence and uniqueness of the strong solution of (1) (see [13], Chap. 3).

We denote by $f_\theta = f_\theta((X_t)_{t \leqslant T})$ a discounted payoff function of an option with parameter $\theta \in \Theta \subset \mathbb{R}^n$ and assume that constants c_1, c_2 exist such that

$$f_\theta(X) \leqslant c_1 \sup_{0 \leqslant t \leqslant T} |X_t| + c_2. \qquad (2)$$

We also assume that the discounted asset price process is a $\mathbb{P}$-martingale and the option price is given by the formula $C_\theta = \mathbb{E} f_\theta$.

In Sect. 2 we consider a combination of importance sampling and control variate methods. More precisely, we define a measure $\mathbb{Q}$ absolutely continuous with respect to the measure $\mathbb{P}$, and assume

$$\widehat{C}_\theta = \rho f_\theta + \eta, \qquad (3)$$

where $\rho = d\mathbb{P}/d\mathbb{Q}$ and $\mathbb{E}^{\mathbb{Q}}\eta = 0$. Note that $\widehat{C}_\theta$ are unbiased estimators for C_θ: $\mathbb{E}^{\mathbb{Q}}\widehat{C}_\theta = C_\theta$. We solve the problem

$$\min_{\rho, \eta} \int_\Theta \mathrm{Var}^{\mathbb{Q}}(\widehat{C}_\theta)\mathsf{Q}(d\theta) \qquad (4)$$

under different assumptions on Q. Maybe it is more natural to solve the minimax problem $\min\limits_{\rho, \eta} \max\limits_{\theta \in \Theta} \sqrt{\mathrm{Var}^{\mathbb{Q}}(\widehat{C}_\theta)} q_\theta$, because it determines the best accuracy of the worst estimator. But it is a much harder problem and we do not consider it.

In Sect. 3, the results of Sect. 2 are specified for the case when the payoff function depends only on the prices of the underlying assets at maturity. The issue of optimal function approximation and other aspects of the constructed estimators application are considered. The results of simulation are shown in Sect. 4.

2 Theoretical Results

Consider the problem of option price valuation with different maturities $t \in \mathscr{T} = (0, T]$ and parameters $k \in \mathscr{K} \subset \mathbb{R}^n$. Let $\theta = \{k, t\}$, $\Theta = \mathscr{K} \times \mathscr{T}$ and $f_\theta = f_{k,t}\left((X_s)_{0 \leqslant s \leqslant t}\right)$. Consider some approaches to the options valuation.

2.1 The First Approach

First, we can simply apply the general results developed in [9]. Let υ_t be an $\mathbb{F}$-adapted d-dimensional process, $\varkappa_t(y)$ be a one-dimensional $\mathbb{F}$-predictable E-marked process and $\varkappa_t(y) > -1$. Denote by δ the pair $\{\upsilon, \varkappa\}$. Let the process $(L_t^\delta)_{0\leqslant t\leqslant T}$ has the form

$$L_t^\delta = \exp\left(\int_0^t \upsilon_s dW_s - \frac{1}{2}\int_0^t |\upsilon_s|^2 ds - \int_0^t \int_E \varkappa_s(y)\nu(ds, dy) + \right.$$
$$\left. + \int_0^t \int_E \log\left(1 + \varkappa_t(y)\right) p(dt, dy)\right). \tag{5}$$

If for each $t \in [0, T]$

$$|\upsilon_t|^2 + \int_E |\varkappa_t(y)|^2 m(dy) \leqslant c(t) \quad \mathbb{P} - \text{a.s.}, \tag{6}$$

where $c(t) \geqslant 0$ is non-random such that $\int_0^T c(t)dt < \infty$, then L_t^δ is a $\mathbb{P}$-martingale and $\mathbb{E}L_T^\delta = 1$ (see [13], Chap. 3). Therefore, we can define the measure $\mathbb{P}^\delta$ by $d\mathbb{P}^\delta = L_T^\delta d\mathbb{P}$. Under the measure $\mathbb{P}^\delta$ the process $W_t^\upsilon = W_t - \int_0^t \upsilon_s ds$ is a Wiener process and $p(dt, dy)$ has the intensity measure $(1 + \varkappa_t(y))\nu(dt, dy)$ (see [13], Chap. 3). Let us denote by $\mathbb{E}^\delta$ the expectation under $\mathbb{P}^\delta$. Define $\rho_t^\delta = \left(L_t^\delta\right)^{-1}$. Denote by φ the pair $\{z, \zeta\}$, where $(z_t)_{0\leqslant t\leqslant T}$ is an $\mathbb{F}$-adapted d-dimensional process and $(\zeta_t(y))_{0\leqslant t\leqslant T}$ is a one-dimensional $\mathbb{F}$-predictable E-marked process such that

$$\mathbb{E}\int_0^T |z_t|^2 dt < \infty, \quad \mathbb{E}\int_0^T \int_E |\zeta_t(y)|^2 \nu(dt, dy) < \infty.$$

Define

$$M_t^{\delta,\varphi} = \int_0^t z_s dW_s^\upsilon + \int_0^t \int_E \zeta_s(y)\tilde{p}(ds, dy), \tag{7}$$

where $\tilde{p}(ds, dy) = p(ds, dy) - (1 + \varkappa_t(y))\nu(dt, dy)$. The martingale $M_t^{\delta,\varphi}$ is square integrable with respect to the measure $\mathbb{P}^\delta$ (see [1], Chap. 8). Consider estimators of the from

$$\widetilde{C}_\theta(\delta, \varphi) = f_\theta \rho_T^\delta + M_T^{\delta,\varphi}. \tag{8}$$

Since the estimator is unbiased, the problem of the weighted variance minimization is reduced to the weighted second moment minimization

$$\min_{\delta,\varphi} \int_\Theta \mathbb{E}^\delta \widetilde{C}_\theta^2(\delta, \varphi) \mathsf{Q}(d\theta). \tag{9}$$

Denote

$$\bar{G} = \int_{\Theta} f_{\theta} \mathsf{Q}(d\theta), \quad \widetilde{G} = \sqrt{\left(\int_{\Theta} f_{\theta}^2 \mathsf{Q}(d\theta)\right) - \bar{G}^2}. \tag{10}$$

As shown in [6], the minimum (9) equals $\left(\mathbb{E}\widetilde{G}\right)^2 + \left(\mathbb{E}\bar{G}\right)^2$ and is attained when

$$\upsilon_t = \frac{\tilde{\alpha}_t}{\tilde{\mu}_t}, \quad \varkappa_t(y) = \frac{\tilde{\beta}_t(y)}{\tilde{\mu}_{t-}}, \quad z_t = -\rho_t^{\delta}(\bar{\alpha}_t - \upsilon_t \bar{\mu}_t),$$
$$\zeta_t(y) = -\rho_{t-}^{\delta}\left(\bar{\beta}_t(y) - \bar{\mu}_{t-}\frac{\varkappa_t(y)}{1+\varkappa_t(y)}\right), \tag{11}$$

where

$$\bar{\mu}_t = \mathbb{E}(\bar{G}|\mathscr{F}_t), \quad \tilde{\mu}_t = \mathbb{E}(\widetilde{G}|\mathscr{F}_t). \tag{12}$$

Processes $\tilde{\alpha}_t, \bar{\alpha}_t, \tilde{\beta}_t(y), \bar{\beta}_t(y)$ are defined by representations

$$d\tilde{\mu}_t = \tilde{\alpha}_t dW_t + \int_E \tilde{\beta}_t(y)\tilde{p}(dt, dy),$$
$$d\bar{\mu}_t = \bar{\alpha}_t dW_t + \int_E \bar{\beta}_t(y)\tilde{p}(dt, dy).$$

The efficiency of estimator (8) can be improved by incorporating additional weights which depend on the parameter θ. Fix $\delta = \{\upsilon, \varkappa\}$, $\varphi = \{z, \zeta\}$ and consider estimators of the form

$$\widetilde{C}_{\theta}(a) = \widetilde{C}_{\theta}(a; \delta, \varphi) = f_{\theta}\rho_T^{\delta} + a(\theta)M_T^{\delta,\varphi}, \tag{13}$$

where $M_T^{\delta,\varphi}$ is defined in (7) and the function $a : \Theta \to \mathbb{R}$ solves the problem

$$\min_a \int_{\Theta} \mathbb{E}^{\delta}\widetilde{C}_{\theta}^2(a)\mathsf{Q}(dk). \tag{14}$$

Since $\mathbb{E}M_T^{\delta,\varphi} = 0$, we get from ([5], Chap. 4) that the optimal function $a^*(\theta)$ which minimizes (14) has the form

$$a^*(\theta) = -\frac{\mathbb{E}(f_{\theta}M_T^{\delta,\varphi})}{\mathbb{E}^{\delta}\left(M_T^{\delta,\varphi}\right)^2}.$$

The minimum equals

$$\int_{\Theta} \mathbb{E}^{\delta}(f_{\theta}\rho_T^{\delta})^2\mathsf{Q}(d\theta) - \int_{\Theta} a^{*2}(\theta)\mathsf{Q}(d\theta)\mathbb{E}^{\delta}\left(M_T^{\delta,\varphi}\right)^2.$$

The function $a^*(\theta)$ is constructed such that $\operatorname{Var}\left(\widetilde{C}_\theta(a^*;\delta,\varphi)\right) \leqslant \operatorname{Var}\left(\widetilde{C}_\theta(\delta,\varphi)\right)$ for any $\theta \in \Theta$. In practice, $a^*(\theta)$ is replaced by its sample counterpart calculated on realizations of the X_t trajectory. This introduce some bias, which is typically $O\,(1/n)$ (see [5], Chap. 4), whereas the standard error of Monte Carlo estimators is $O\left(1/\sqrt{n}\right)$.

2.2 The Second Approach

The second approach is represented by estimators of the form

$$\widehat{C}_\theta(\delta) = \rho_t^\delta f_\theta. \tag{15}$$

These estimators were examined in [7] in the case of a diffusion model and $\theta = t$. Under additional condition on Q and f_t, the problem of the weighted variance minimization was reduced to solving a nonlinear partial differential equation (see Theorem 2.3 in [7]). For the jump-diffusion model, we can deduce a similar nonlinear partial integro-differential equation such that the optimal estimator is expressed by its solution. But this approach has evident disadvantages: the approximation of the nonlinear PIDE solution is quite complicated; the payoff function may depend only on the underlying asset price at maturity; additional conditions (like Hölder continuity) should be imposed on the coefficients of SDE (1) and the payoff function; the measure Q is assumed to be absolutely continuous.

We consider more realistic and less restrictive case when Q is a discrete measure.

Let $\mathscr{T} = \{t_i\}_{i=1}^n$, where $0 = t_0 < t_1 < \ldots < t_n = T$. Since the estimator $\widehat{C}_\theta(\delta)$ is unbiased, the problem of the weighted variance minimization is reduced to the problem of the second moment minimization

$$\min_\delta \int_\Theta \mathbb{E}^\delta \widehat{C}^2_\theta(\delta)\mathsf{Q}(\theta). \tag{16}$$

Assume that the measure $\mathsf{Q}(dk,dt) = P(dk)Q(dt)$, $P(\mathscr{K}) = 1$, $Q(\mathscr{T}) = 1$. Denote by q_i the weights $Q\{t_i\}$. Define sequences

$$H_{t_i} = \sqrt{\int_{\mathscr{K}} f^2_{k,t_i} P(dk) q_i}, \quad G_{t_i} = \sqrt{\left(\mathbb{E}(G_{t_{i+1}}|\mathscr{F}_{t_i})\right)^2 + H^2_{t_i}}, \quad G_{t_n} = H_{t_n},$$

then

$$\int_\Theta \mathbb{E}^\delta \widehat{C}^2_\theta(\delta)\mathsf{Q}(dk,dt) = \sum_{i=1}^n q_i \mathbb{E}^\delta \int_{\mathscr{K}} \left(f_{k,t_i}\rho^\delta_{t_i}\right)^2 P(dk) = \sum_{i=1}^n \mathbb{E}^\delta \left(H_{t_i}\rho^\delta_{t_i}\right)^2.$$

Theorem 1. *There exist an $\mathbb{F}$-adapted processes $\hat{\alpha}_t^{(i)}$, and an $\mathbb{F}$-predictable E-marked processes $\hat{\beta}_t^{(i)}(y)$, $i = 1, \ldots, n$ such that*

$$d\hat{\mu}_t^{(i)} = \hat{\alpha}_t^{(i)} dW_t + \int_E \hat{\beta}_t^{(i)}(y)\tilde{p}(dt, dy), \tag{17}$$

where $\hat{\mu}_t^{(i)} = \mathbb{E}(G_{t_i} | \mathscr{F}_t)$. Minimum (16) equals

$$\min_{\delta} \sum_{i=1}^{n} \mathbb{E}^{\delta} \left(H_{t_i} \rho_{t_i}^{\delta} \right)^2 = \left(\mathbb{E} G_{t_1} \right)^2,$$

and is attained when

$$\hat{\upsilon}_t = \sum_{i=1}^{n} \mathbf{1}_{(t_{i-1}, t_i]}(t) \frac{\hat{\alpha}_t^{(i)}}{\hat{\mu}_t^{(i)}}, \quad \hat{\varkappa}_t(y) = \sum_{i=1}^{n} \mathbf{1}_{(t_{i-1}, t_i]}(t) \frac{\hat{\beta}_t^{(i)}(y)}{\hat{\mu}_{t-}^{(i)}}, \tag{18}$$

if the condition (6) for $\hat{\upsilon}$, $\hat{\varkappa}$ holds.

Proof. Prove that the martingale $\hat{\mu}_t^{(i)}$ on $[0, t_i]$ is square integrable for $i = 1, \ldots, n$. We have

$$\mathbb{E} \left(\hat{\mu}_t^{(i)} \right)^2 \leqslant \mathbb{E} G_{t_i}^2 \leqslant \mathbb{E} \left(G_{t_{i+1}}^2 + H_{t_i}^2 \right) \leqslant \ldots \leqslant \mathbb{E} \left(\sum_{j=i}^{n} H_{t_j}^2 \right).$$

From (2) and the inequality $\mathbb{E}\left(\sup\limits_{0 \leqslant t \leqslant T} |X_t|^2 \right) < \infty$ (see [13], Chap. 3) it follows that $\mathbb{E} H_{t_i}^2 \leqslant \mathbb{E} \left(c_1 \sup\limits_{0 \leqslant t \leqslant T} |X_t| + c_2 \right)^2 \leqslant C < \infty$ and therefore, $\mathbb{E} \left(\hat{\mu}_t^{(i)} \right)^2 < C$ for any i and $t \in [0, t_i]$. Applying the martingale representation theorem (see [13], Chap. 2) we get that there exist processes $\hat{\alpha}_t^{(i)}$, $\hat{\beta}_t^{(i)}(y)$ such that the differential for $\hat{\mu}_t^{(i)}$ has the form (17). Define $\hat{\upsilon}_t$, $\hat{\varkappa}_t(y)$ as in (18), then from Itô's lemma we have that

$$\begin{aligned}
\ln \hat{\mu}_{t_i}^{(i)} = {} & \ln \hat{\mu}_{t_{i-1}}^{(i)} + \int_{t_{i-1}}^{t_i} \frac{\hat{\alpha}_t^{(i)}}{\hat{\mu}_t^{(i)}} dW_t - \frac{1}{2} \int_{t_{i-1}}^{t_i} \frac{|\hat{\alpha}_t^{(i)}|^2}{(\hat{\mu}_t^{(i)})^2} dt - \int_{t_{i-1}}^{t_i} \int_E \frac{\hat{\beta}_t(y)^{(i)}}{\hat{\mu}_{t-}^{(i)}} \nu(dt, dy) \\
& + \int_{t_{i-1}}^{t_i} \int_E \ln \left(1 + \frac{\hat{\beta}_t^{(i)}(y)}{\hat{\mu}_{t-}^{(i)}} \right) p(dt, dy) = \ln \hat{\mu}_{t_{i-1}}^{(i)} + \int_{t_{i-1}}^{t_i} \hat{\upsilon}_t dW_t - \quad (19) \\
& - \frac{1}{2} \int_{t_{i-1}}^{t_i} |\hat{\upsilon}_t|^2 dt - \int_{t_{i-1}}^{t_i} \int_E \hat{\varkappa}_t(y) \nu(dt, dy) \\
& + \int_{t_{i-1}}^{t_i} \int_E \ln \left(1 + \hat{\varkappa}_t(y) \right) p(dt, dy).
\end{aligned}$$

Since $\hat{\upsilon}$, $\hat{\varkappa}$ satisfy (6), $\mathbb{E}L_T^{\hat{\upsilon},\hat{\varkappa}} = 1$ and we can construct the probability measure $\mathbb{P}^{\hat{\upsilon},\hat{\varkappa}} = \mathbb{P}^{\hat{\delta}}$ with the density $L_T^{\hat{\delta}} = d\mathbb{P}^{\hat{\delta}}/d\mathbb{P}$. From (5) and (19) it follows that

$$\frac{L^{\hat{\delta}}_{t_i}}{L^{\hat{\delta}}_{t_{i-1}}} = \exp\left(\ln \hat{\mu}^{(i)}_{t_i} - \ln \hat{\mu}^{(i)}_{t_{i-1}}\right) = \frac{G_{t_i}}{\mathbb{E}\left(G_{t_i}|\mathscr{F}_{t_{i-1}}\right)}.$$

Define $\rho^{\delta}_{t_{i-1},t_i} = \rho^{\delta}_{t_i}/\rho^{\delta}_{t_{i-1}}$. From Jensen's inequality it follows that for any δ

$$\mathbb{E}^{\delta}\left(\left(G_{t_i}\rho^{\delta}_{t_i}\right)^2|\mathscr{F}_{t_{i-1}}\right) = \left(\rho^{\delta}_{t_{i-1}}\right)^2\mathbb{E}^{\delta}\left(\left(G_{t_i}\rho^{\delta}_{t_{i-1},t_i}\right)^2|\mathscr{F}_{t_{i-1}}\right) \geqslant \left(\rho^{\delta}_{t_{i-1}}\mathbb{E}\left(G_{t_i}|\mathscr{F}_{t_{i-1}}\right)\right)^2$$

for $i = 1,\dots,n$. Since

$$\rho^{\delta}_{t_{i-1},t_i} = \frac{L^{\hat{\delta}}_{t_{i-1}}}{L^{\hat{\delta}}_{t_i}} = \frac{\mathbb{E}\left(G_{t_i}|\mathscr{F}_{t_{i-1}}\right)}{G_{t_i}}, \tag{20}$$

we have

$$\mathbb{E}^{\hat{\delta}}\left(\left(G_{t_i}\rho^{\hat{\delta}}_{t_i}\right)^2|\mathscr{F}_{t_{i-1}}\right) = \left(\rho^{\hat{\delta}}_{t_{i-1}}\mathbb{E}\left(G_{t_i}|\mathscr{F}_{t_{i-1}}\right)\right)^2 \tag{21}$$

for $i = 1,\dots,n$.

Denote $A_n(\delta) = \sum_{i=1}^{n}(H_{t_i}\rho^{\delta}_{t_i})^2$ and recall that $G_{t_n} = H_{t_n}$. From Jensen's inequality it follows that for any δ

$$\begin{aligned}
\mathbb{E}^{\delta}A_n(\delta) &= \mathbb{E}^{\delta}\left(A_{n-1}(\delta) + \mathbb{E}^{\delta}\left(\left[G_{t_n}\rho^{\delta}_{t_n}\right]^2\Big|\,\mathscr{F}_{t_{n-1}}\right)\right) \geqslant \\
&\geqslant \mathbb{E}^{\delta}\left(A_{n-1}(\delta) + \left[\rho^{\delta}_{t_{n-1}}\mathbb{E}\left(G_{t_n}|\,\mathscr{F}_{t_{n-1}}\right)\right]^2\right) = \\
&= \mathbb{E}^{\delta}\left(A_{n-2}(\delta) + \left(\rho^{\delta}_{t_{n-1}}\right)^2\left(H^2_{t_{n-1}} + [\mathbb{E}\left(G_{t_n}|\,\mathscr{F}_{t_{n-1}}\right)]^2\right)\right) = \\
&= \mathbb{E}^{\delta}\left(A_{n-2}(\delta) + \mathbb{E}^{\delta}\left(\left[G_{t_{n-1}}\rho^{\delta}_{t_{n-1}}\right]^2\Big|\,\mathscr{F}_{t_{n-2}}\right)\right) \geqslant \dots \\
&\dots \geqslant \mathbb{E}^{\delta}\left(\left[G_{t_1}\rho^{\delta}_{t_1}\right]^2\right) \geqslant \left(\mathbb{E}G_{t_1}\right)^2.
\end{aligned}$$

From (21) it follows that for $\hat{\delta} = \{\hat{\upsilon},\hat{\varkappa}\}$

$$\sum_{i=1}^{n}\mathbb{E}^{\hat{\delta}}\left(H_{t_i}\rho^{\hat{\delta}}_{t_i}\right)^2 = \left(\mathbb{E}G_{t_1}\right)^2.$$

□

Note that in the case $M_T^{\delta,\varphi} = 0$ the minimal weighted variance of estimator (15) is not greater than the minimal weighted variance of estimator (8). It follows from the following inequality

$$\mathbb{E}^{\delta}\left(f_\theta \rho_T^{\delta}\right)^2 = \mathbb{E} f_\theta^2 \rho_T^{\delta} \geqslant \mathbb{E} f_\theta^2 \rho_t^{\delta} = \mathbb{E}^{\delta}\left(f_\theta \rho_t^{\delta}\right)^2.$$

3 Application to Options Pricing

In this section the results of Sect. 2 are applied for options pricing. We construct approximation for optimal functionals G_{t_i} defined in Theorem 1. This approximation is applied in simulations. After that we consider options with payoff functions which depend on the underlying asset price at maturity and specify the theoretical results in this case.

In practice we approximate the optimal functionals $G_{t_i} = \sqrt{\mathbb{E}^2(G_{t_{i+1}}|\mathscr{F}_{t_i}) + H_{t_i}^2}$ defined in Theorem 1 by $\widetilde{G}_{t_i}$

$$\widetilde{G}_{t_i} = \mathbb{E}\left(\sqrt{\sum_{j=i}^{n} H_{t_j}^2}\,\middle|\,\mathscr{F}_{t_i}\right). \tag{22}$$

Now we specify the results of Sect. 2 for rainbow options for which the payoff depends only on the prices of underlying assets at maturity. A rainbow option is an option whose payoff function depends on more than one underlying asset. For example, a call-on-max option has a payoff function of the form $\max(\max(S_T^1, \dots, S_T^n) - K, 0)$. For other types of rainbow options see [10].

We assume for simplicity that the interest rate is constant and the discounted payoff function has the form $f_\theta = f_{k,t}(X_t)$, where X_t is the solution of SDE (1). Note that not all components of X_t are prices of underlying assets. One of them could be a stochastic volatility, for example. We estimate option prices $C_{k,t} = \mathbb{E} f_{k,t}$.

3.1 *The Case of Estimator (13)*

Recall that the functional $\bar{G}$ is defined in (10). Let $Z_1(s,t) = \int_s^t \int_{\mathscr{K}} f_{k,\tau} \mathsf{Q}(dk, d\tau)$, then we have

$$\bar{\mu}_t = \mathbb{E}(\bar{G}|\mathscr{F}_t) = Z_1(0,t) + \mathbb{E}\left(Z_1(t,T)|\, X_t\right) = Z_1(0,t) + \bar{u}\,(t, X_t)\,,$$

where $\bar{u}(t,x) = \mathbb{E}(Z_1(t,T)|X_t = x)$. Assume that the function $\bar{u}(t,x)$ is smooth enough, then using Itô's lemma for $\bar{u}(t,X_t)$, we get the differential of the $\mathbb{P}$-martingale $\bar{\mu}_t$ in the form

$$
\begin{aligned}
d\bar{\mu}_t &= b^T(t, X_t)\nabla\bar{u}(t, X_t)dW_t \\
&\quad + \int_E \left(\bar{u}\left(t, X_{t-} + \gamma(t, X_{t-}, y)\right) - \bar{u}(t, X_{t-})\right)\tilde{p}(dt, dy).
\end{aligned}
\tag{23}
$$

Consider $\tilde{\mu}_t$ defined in (12). Let $Z_2(s,t) = \int_s^t \int_{\mathcal{K}} f_{k,\tau}^2 \mathsf{Q}(dk, d\tau)$, denote by Z_t the pair $(Z_1(0,t), Z_2(0,t))$, then we have

$$
\begin{aligned}
\tilde{\mu}_t &= \mathbb{E}(\widetilde{G}|\mathscr{F}_t) = \mathbb{E}\left(\sqrt{Z_2(0,T) - Z_1^2(0,T)} \middle| \mathscr{F}_t \right) = \\
&= \mathbb{E}\left(\sqrt{Z_2(0,t) + Z_2(t,T) - (Z_1(0,t) + Z_1(t,T))^2} \middle| X_t, Z_1(0,t), Z_2(0,t) \right) = \\
&= \widetilde{u}(t, X_t, Z_t),
\end{aligned}
$$

where $\tilde{u}(t,x,z) = \mathbb{E}\left(\sqrt{z_2 + Z_2\,(t,\,T) - (z_1 + Z_1\,(t,\,T))^2} \middle| X_t = x,\ Z_t = z \right)$. Denote by $\nabla_x\tilde{u}$ the vector of derivatives of $\tilde{u}(t,x,z)$ with respect to x_i, i = 1,..., d. Applying Itô's lemma for $\tilde{u}(t, X_t, Z_t)$, we get the differential of the $\mathbb{P}$-martingale $\tilde{\mu}_t$ in the form

$$
\begin{aligned}
d\tilde{\mu}_t &= b^T(t, X_t)\nabla_x\tilde{u}(t, X_t, Z_t)dW_t \\
&\quad + \int_E \left(\tilde{u}\left(t, X_{t-} + \gamma(t, X_{t-}, y), Z_{t-}\right) - \tilde{u}(t, X_{t-}, Z_{t-})\right)\tilde{p}(dt, dy).
\end{aligned}
$$

Thus, we obtain the representations for υ, $\varkappa$, z, ζ defined in (11):

$$
\begin{aligned}
\upsilon_t &= \frac{b^T(t, X_t)\nabla_x\tilde{u}(t, X_t, Z_t)}{\tilde{u}(t, X_t, Z_t)}, \quad \varkappa_t(y) = \frac{\tilde{u}(t, X_{t-} + \gamma(t, X_{t-}, y), Z_{t-})}{\tilde{u}(t, X_{t-}, Z_{t-})} - 1, \\
z_t &= -\rho_t^\delta b^T(t, X_t)\left(\nabla\bar{u}(t, X_t) - \frac{\nabla_x\tilde{u}(t, X_t, Z_t)}{\tilde{u}(t, X_t, Z_t)}(\bar{u}(t, X_t) + Z_1(0,t)) \right) \qquad (24) \\
\zeta_t(y) &= -\rho_{t-}^\delta \left(\bar{u}(t, X_{t-} + \gamma(t, X_{t-}, y)) - 2\bar{u}(t, X_{t-}) - Z_1(0,t)\right. \\
&\quad \left. + (Z_1(0,t) + \bar{u}(t, X_{t-}))\frac{\tilde{u}(t, X_{t-}, Z_{t-})}{\tilde{u}(t, X_{t-} + \gamma(t, X_{t-}, y), Z_{t-})} \right).
\end{aligned}
$$

3.2 The Case of Estimator (15)

The martingale $\hat{\mu}_t^{(i)}$ defined in the statement of Theorem 1 has the differential of the form (23), where $\bar{u}(t,x)$ should be replaced by $\hat{u}^{(i)}(t,x) = \mathbb{E}\left(G_{t_i} \,|\, X_t = x\right), t \leqslant t_i$. Thus, the optimal υ, $\varkappa$ defined in (18) have the representation

$$\upsilon_t = \frac{b^T(t,X_t)\nabla\hat{u}(t,X_t)}{\hat{u}(t,X_t)}, \quad \varkappa_t(y) = \frac{\hat{u}(t,X_{t-}+\gamma(t,X_{t-},y))}{\hat{u}(t,X_{t-})} - 1,$$

where $\hat{u}(t,x) = \sum_{i=1}^n \mathbf{1}_{(t_{i-1},t_i]}(t)\hat{u}^{(i)}(t,x)$.

To simplify the calculations, the function $\hat{u}^{(i)}(t,x)$ is approximated by $\tilde{u}^{(i)}(t,x) = \mathbb{E}\left(\widetilde{G}_{t_i} \,\middle|\, X_t = x\right)$, where $\widetilde{G}_{t_i}$ is defined in (22). Since $H_{t_i} = \sqrt{\int_{\mathscr{K}} f_k^2(t_i,X_{t_i})P(dk)q_i}$, we have for $t \leqslant t_i$

$$\begin{aligned}\tilde{u}^{(i)}(t,x) &= \mathbb{E}\left(\sqrt{\sum_{j=i}^n H_{t_j}^2}\,\middle|\, X_t = x\right) \\ &= \mathbb{E}\left(\sqrt{\sum_{j=i}^n q_j e^{-2\int_t^{t_j} r_s ds}\int_{\mathscr{K}} f_k^2(t_j,X_{t_j})P(dk)}\,\middle|\, X_t = x\right).\end{aligned} \tag{25}$$

3.3 Application to Simulation

Optimal processes υ, $\varkappa$, z, ζ for the estimator (13), which minimize the weighted variance (9), are given in (24). In order to simplify the computations, we calculate υ, $\varkappa$, z, ζ as the solution of the problem

$$\min_{\upsilon,\varkappa,z,\zeta}\int_{\mathscr{K}} \mathbb{E}^{\upsilon,\varkappa}\widetilde{C}^2_{k,T}(\upsilon,\varkappa,z,\zeta)P(dk).$$

Then in formulas (24) we have $Z_1(0,t) = 0$, $Z_2(0,t) = 0$ for $t < T$ and

$$\bar{u}(t,x) = \mathbb{E}\left(\int_{\mathscr{K}} f_{k,T}P(dk)|X_t = x\right), \quad \tilde{u}(t,X_t,0,0) = \tilde{u}(t,X_t),$$

where

$$\tilde{u}(t,x) = \mathbb{E}\left(\sqrt{\int_{\mathscr{K}} f_{k,T}^2 P(dk) - \left(\int_{\mathscr{K}} f_{k,T}P(dk)\right)^2}\,\middle|\, X_t = x\right).$$

There are some difficulties in application of the constructed estimator. First, we need to approximate the functions $\bar{u}(t,x)$, $\tilde{u}(t,x)$. In order to do this, we simplify the original model for X_t. Approximate the process X_t by the process $\widetilde{X}_t$, which satisfies SDE (1) with coefficient functions $\tilde{a}(t,x)$, $\tilde{b}(t,x)$, $\tilde{\gamma}(t,x,y)$ such that

$$\widetilde{X}_t = h\left(t, W_t, \int_0^t\int_E \beta(t,y)p(dt,dy)\right) \tag{26}$$

for some deterministic functions h, β. The form of the process $\widetilde{X}_t$ allows us to reduce significantly the quantity of pseudo-random numbers generated for $\widetilde{X}_t$ simulation. The function $\tilde{u}(t,x)$ is approximated by

$$\tilde{v}(t,x) = \mathbb{E}\left(\sqrt{\int_{\mathscr{K}} f_{k,T}^2(\widetilde{X}_T)P(dk) - \left(\int_{\mathscr{K}} f_{k,T}(\widetilde{X}_T)P(dk)\right)^2} \,\middle|\, \widetilde{X}_t = x \right).$$

The standard Monte carlo estimator is applied for $\tilde{v}(t,x)$ calculation on a grid, and then we use interpolation for $\tilde{v}(t,x)$ calculation at particular points. The gradient $\nabla\tilde{u}(t,x)$ is approximated by $\nabla\tilde{v}(t,x)$. In the same way $\bar{u}(t,x)$ and $\nabla\bar{u}(t,x)$ can be approximated. Computation of the approximations of $\bar{u}(t,x)$ and $\tilde{u}(t,x)$ on the grid is performed before the main simulations of the trajectories of X_t.

The same method is applied for approximation of the function $\tilde{u}^{(i)}(t,x)$ defined in (25). It is approximated by

$$\tilde{v}^{(i)}(t,x) = \mathbb{E}\left(\sqrt{\sum_{j=i}^{n} q_j e^{-2\int_t^{t_j} r_s ds} \int_{\mathscr{K}} f_k^2(t_j, \widetilde{X}_{t_j})P(dk)} \,\middle|\, \widetilde{X}_t = x \right),$$

where $\widetilde{X}_t$ has the form (26). For example, if X_t follows the Heston model (the volatility is stochastic), then we can chose $\widetilde{X}_t$ such that it follows the Black–Scholes model (the volatility is constant). The function $\tilde{u}(t,x) = \sum_{i=1}^n \mathbf{1}_{(t_{i-1},t_i]}(t)\tilde{u}^{(i)}(t,x)$ is approximated by $\tilde{v}(t,x) = \sum_{i=1}^n \mathbf{1}_{(t_{i-1},t_i]}(t)\tilde{v}^{(i)}(t,x)$. The standard Monte carlo estimator is applied for $\tilde{v}(t,x)$ calculation on a grid.

The other difficulty in application of the estimators (13) and (15) is that we also need to calculate the following integrals

$$\int_0^T \int_E \varkappa_t(y)m(dy)dt, \quad \int_0^T \int_E \zeta_t(y)m(dy)dt,$$

which lead to computation of the integrals of the type $I_t = \int_E \tilde{v}(t, x + \gamma(t,x,y))m(dy)$. If $m(dy)$ is a discrete distribution, then I_t could be easily calculated. If not, then I_t can be calculated by Monte Carlo method, which introduces additional error and reduces the efficiency of the constructed estimators. So, in applications we consider only the case when $m(dy)$ is a discrete distribution.

3.4 Computational Time

Below we introduce the formula for the total time of options prices estimation with parameters $\theta \in \Theta$. Denote by ϵ_θ the standard error of the estimator $\widehat{C}_\theta$. Let N be a number of simulated trajectories of X_t, $\sigma_\theta = \sqrt{\mathrm{Var}(\widehat{C}_\theta)}$, then $\epsilon_\theta = c_\alpha \sigma_\theta/\sqrt{N}$, where c_α is a constant dependent on the confidence level α. Define the weighted error of the estimators as

$$\epsilon = \sqrt{\int_\Theta \epsilon_\theta^2 \mathsf{Q}(d\theta)},$$

then $\epsilon = c_\alpha\sqrt{D/N}$, where D is the weighted variance of the estimators $\widehat{C}_\theta$. Thus, the total computational time T_ϵ, which is necessary to reach the accuracy ϵ is expressed in the form

$$T_\epsilon = t_0 + c_\alpha^2 \frac{D}{\epsilon^2}(t_1 + t_2), \tag{27}$$

where t_0 is the time of preliminary computations, t_1 is the time of one trajectory simulation, t_2 is the time of the estimators $\widehat{C}_\theta$ computation on one trajectory for different $\theta \in \Theta$. In comparison with the standard Monte Carlo estimator the times t_1, t_2 increase insignificantly due to preliminary computations of optimal functions approximations on a grid and calculation of the intermediate values between the grid points by interpolation. Thus, if we are interested in option prices calculation with high accuracy, it is more efficient to apply estimators with smaller weighted variance however big t_0 is.

4 Simulation Results

Here we apply the constructed estimators in simulations. As a rate of efficiency we use the ratio E introduced in [9] as a ratio of computational times $T_\varepsilon^{(1)}/T_\varepsilon^{(0)}$, where $T_\varepsilon^{(i)}$ is defined in (27) and $T_\varepsilon^{(0)}$ corresponds to the standard Monte Carlo estimator. Since $t_0^{(0)} = 0$ for the standard Monte Carlo estimator, we have the following formula for E:

$$E = \frac{t_0^{(1)}}{T_\varepsilon^{(0)}} + \frac{D^{(1)}}{D^{(0)}} \frac{t_1^{(1)} + t_2^{(1)}}{t_1^{(0)} + t_2^{(0)}}.$$

We will use the following notations to present the results of simulations: $\mathscr{D} = \frac{D^{(1)}}{D^{(0)}}$, $\mathscr{T} = \frac{t_0^{(1)}}{T_\varepsilon^{(0)}}$, $\tau = \frac{t_1^{(1)}+t_2^{(1)}}{t_1^{(0)}+t_2^{(0)}}$. Consider the process $X_t = \left((S_t^{(1)}, V_t^{(1)}), \dots, (S_t^{(d)}, V_t^{(d)})\right)$, where $S_t^{(i)}$ is a price process of the i-th underlying asset, $V_t^{(i)}$ is a volatility process. We assume that for $i = 1, \dots, d$

$$dS_t^{(i)} = \mu_i(t) S_t^{(i)} dt + S_t^{(i)} \sqrt{V_t^{(i)}} dW_t^{(i)} + d \sum_{n=1}^{N_t} \left(e^{Y_n^{(i)}} - 1\right) S_{T_n-}^{(i)},$$

$$dV_t^{(i)} = \xi_i(\eta_i - V_t^{(i)})dt + \alpha_i \sqrt{V_t^{(i)}} \left(\rho_i dW_t^{(i)} + \sqrt{1-\rho_i^2} d\widetilde{W}_t^{(i)}\right),$$

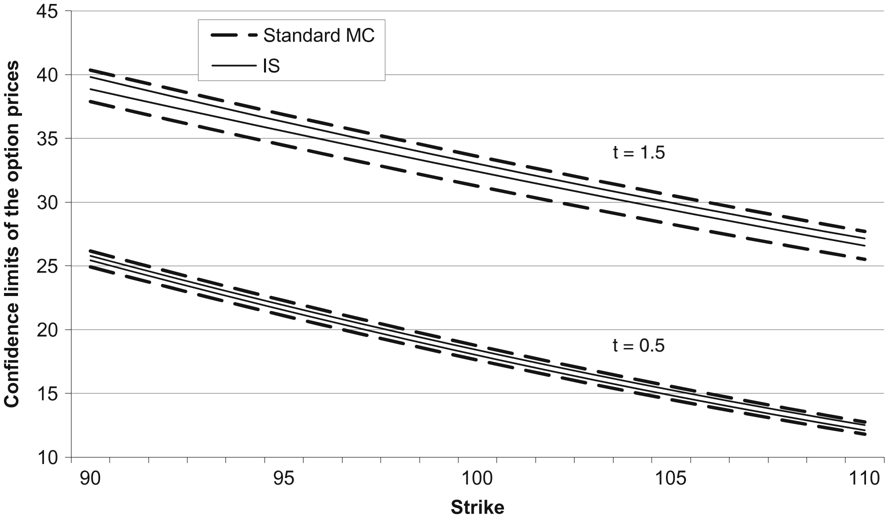


Fig. 1 Confidence limits for the option price.

where $\widetilde{W}_t = \left(\widetilde{W}_t^{(1)}, \dots, \widetilde{W}_t^{(d)}\right)^T$ is the d-dimensional standard Brownian motion; $W_t = \left(W_t^{(1)}, \dots, W_t^{(d)}\right)^T$ is the Brownian motion with correlation matrix Σ; W_t and $\widetilde{W}_t$ are independent; N_t is a Poisson process with intensity λ; relative jumps $Y_n = \left(Y_n^{(1)}, \dots, Y_n^{(d)}\right)^T$ have the following distribution:

$$Y_n = \begin{cases} U_c, & \text{with prob. } \frac{\lambda_c}{\lambda} p_c, \\ D_c, & \text{with prob. } \frac{\lambda_c}{\lambda}(1 - p_c), \end{cases} \qquad Y_n^{(i)} = \begin{cases} U_i, & \text{with prob. } \frac{\lambda_i}{\lambda} p_i, \\ D_i, & \text{with prob. } \frac{\lambda_i}{\lambda}(1 - p_i), \end{cases}$$

where $U_c, D_c \in \mathbb{R}^d$, $U_i, D_i \in \mathbb{R}$, $\lambda = \lambda_1 + \ldots + \lambda_d + \lambda_c$. This model can be considered as a multivariate Heston model (see [2]) with jumps. We consider three underlying assets, i.e. $d = 3$ with $S_0^{(i)} = 100$. Let the confidence level $\alpha = 0.99$.

Example 1. We estimate call-on-max options with maturities $t \in \{0.5 + 0.25i\}_{i=0}^{4}$ and strikes $K \in \{90 + i\}_{i=0}^{21}$. We apply the estimator defined in (15). For 10^4 simulated trajectories we have $E = 1.17$, $\tau = 1.4$, $\mathscr{T} = 1.08$, $\mathscr{D} = 0.065$. The weighted variance was reduced $1/\mathscr{D} = 15.43$ times. For 10^6 simulations $E = 0.1$. The confidence limits of the option prices are shown in Fig. 1. The dashed lines "Standard MC" correspond to the standard Monte Carlo estimator, the solid lines "IS" correspond to the importance sampling estimator.

Applying the estimator defined in (13) without changing measure (i.e. $\rho_T^{\delta} \equiv 1$), we get that for 10^4 simulations $E = 0.69$, $\tau = 1.36$, $\mathscr{T} = 0.26$, $\mathscr{D} = 0.31$. The weighted variance was reduced $1/\mathscr{D} = 3.18$ times. For 10^6 simulations $E = 0.43$.

Example 2. We estimate call-on-max options with two maturities $t \in \{0.75, 1\}$ and strikes $K \in \{100 + i\}_{i=1}^{5}$. Applying the estimator defined in (15) we get for 10^6 simulated trajectories that the weighted variance was reduced $1/\mathscr{D} = 30.59$ times, $E = 0.058$.

Acknowledgements This research was supported by RFBR under the grant number 11-01-00769-a.

References

1. Cont, R., Tankov, P.: Financial Modelling with Jump Processes. Chapman & Hall/CRC, Boca Raton (2004)
2. Dimitroff, G., Lorenz, S., Szimayer, A.: A parsimonious milti-asset heston model: Calibration and derivative pricing. http://ssrn.com/abstract=1435199 (19 April, 2010)
3. Fouque, J.P., Han, C.H.: Variance reduction for Monte Carlo methods to evaluate option prices under multi-factor stochastic volatility models. Quantitative Finance **4**(5), 597–606 (2004)
4. Fouque, J.P., Tullie, T.: Variance reduction for Monte Carlo simulation in a stochastic volatiltiy environment. Quantitative Finance **2**, 24–30 (2002)
5. Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer-Verlag, New York (2004)
6. Gormin, A.A.: Importance sampling and control variates in the weighted variance minimization. Proceedings of the 6th St.Petersburg Workshop on Simulation, 109–113 (2009)
7. Gormin, A.A., Kashtanov, Y.N.: The weighted variance minimization for options pricing. Monte Carlo Methods and Applications **13**(5–6), 333–351 (2007)
8. Gormin, A.A., Kashtanov, Y.N.: Variance reduction for multiple option parameters. Journal of Numerical and Applied Mathematics **1**(96), 96–104 (2008)
9. Gormin, A.A., Kashtanov, Y.N.: The weighted variance minimization in jump-diffusion stochastic volatility models. Monte Carlo and Quasi-Monte Carlo Methods 2008, 383–395 (2009)
10. Lyuu, Y.-D.,Teng, H.-W.: Unbiased and efficient greeks of financial options. Finance and Stochastic, Online First$^{\text{TM}}$ (3 September, 2010)
11. Newton, N.J.: Variance reduction for simulated diffusions. SIAM Journal on Applied Mathematics **54**(6), 1780–1805 (1994)
12. Schoenmakers, J. G. M., Heemink, A. W. : Fast valuation of financial derivatives. The Journal of Computational Finance **1**(1), 47–62 (1997)
13. Situ, R.: Theory of Stochastic Differential Equations with Jumps and Applications. Springer-Verlag, New York (2005)

Enumerating Quasi-Monte Carlo Point Sequences in Elementary Intervals

Leonhard Grünschloß, Matthias Raab, and Alexander Keller

Abstract Low discrepancy sequences, which are based on radical inversion, expose an intrinsic stratification. New algorithms are presented to efficiently enumerate the points of the Halton and (t, s)-sequences per stratum. This allows for consistent and adaptive integro-approximation as for example in image synthesis.

1 Introduction

Similar to real world digital cameras, pixel colors can be modeled as sensor response to the radiance function. The discrete, pixel-based image thus results from projecting the radiance function onto a regular lattice of sensor functions.

These functionals can be computed by applying the Monte Carlo method on a per pixel basis, which allows one to adaptively choose the number of samples per pixel. The straightforward application of quasi-Monte Carlo methods per pixel in order to improve convergence reveals correlation artifacts, which can be removed by giving up determinism, for example by random scrambling [12, 15].

These issues can be avoided by interpreting image synthesis as a parametric integration problem, i.e., by estimating multiple functionals using a single quasi-Monte Carlo point sequence over the whole image plane: In [10] the stratification properties of the (finite) Hammersley point set have been used to efficiently map pixels to samples. This approach has been generalized for the Halton sequence in order to allow for pixel adaptive sampling [11]: As illustrated in Fig. 1, a large table of the size of the number of pixels has been used to look up the index of

L. Grünschloß
Rendering Research, Weta Digital, New Zealand
e-mail: leonhard@gruenschloss.org

M. Raab · A. Keller
NVIDIA ARC GmbH, Berlin, Germany
e-mail: iovis@gmx.net; keller.alexander@googlemail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_21, © Springer-Verlag Berlin Heidelberg 2012

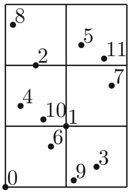


Fig. 1 A plot of the first 12 points of the scaled two-dimensional Halton sequence $(2\Phi_2(i), 3\Phi_3(i))$ labeled with their index i. While the point sequence jumps across the domain, it can be seen that points inside each of the depicted 2 × 3 strata can be enumerated using a stride of 6, which is the number of strata.

the first Halton sequence point in a pixel, while the subsequent samples have been enumerated using a fixed stride.

In the following we improve these approaches for low discrepancy sequences, whose intrinsic stratification is based on radical inversion: Algorithms, which only require a lookup table size linear in dimension, are derived for the Halton and (t, s)-sequences.

The results are applied to image synthesis, where by using the first two dimensions of the Sobol' sequence for parametric quasi-Monte Carlo integration over the whole image plane the good uniformity properties across pixels are maintained. In particular, the consistent and deterministic framework allows one to adaptively determine the number of samples per pixel according to an arbitrary density as illustrated in Fig. 4.

2 Radical Inversion and Stratification

Many low discrepancy sequences are based on the principle of radical inversion

$$\begin{aligned} \Phi_b : \mathbb{N}_0 &\to \mathbb{Q} \cap [0,1) \\ i = \sum_{k=0}^{\infty} a_k(i) b^k &\mapsto \sum_{k=0}^{\infty} a_k(i) b^{-k-1}, \end{aligned} \tag{1}$$

where $a_k(i)$ denotes the $(k+1)^{\text{st}}$ digit of the integer $i \in \mathbb{N}_0$ in base b. In fact, the radical inverse (also known as van der Corput sequence [2, 13]) mirrors the digits at the decimal point. Using permutations $\sigma_b\,(a_k(i))$ of $\{0, \ldots, b-1\}$ instead of the original digits can improve discrepancy [5, 10]. Note that this generalization as well as the original construction are bijections.

Inserting $i = b^d \cdot h + l$ with $l \in \{0, \ldots, b^d - 1\}$ yields

$$\Phi_b(i) = \Phi_b(b^d \cdot h + l) = b^{-d} \cdot \Phi_b(h) + \Phi_b(l), \tag{2}$$

revealing that

- The d least significant digits l select an interval $b^d \cdot \Phi_b(l) \in \{0, \dots, b^d - 1\}$, while
- The most significant digits h determine the point inside that interval.

Therefore any subsequence of the van der Corput sequence at a step size of b^d falls into the same interval of width b^{-d}.

3 Enumerating the Halton Sequence per Stratum

The s-dimensional points

$$\mathbf{x}_i := (\Phi_{b_1}(i), \Phi_{b_2}(i), \dots, \Phi_{b_s}(i)) \in [0, 1)^s$$

constitute the Halton sequence [7], where typically b_j is the j-th prime number, although for low discrepancy it is sufficient that the b_j are relatively prime.

As illustrated in Fig. 1, the stratification properties of radical inversion (2) allude to an s-dimensional stratification, where each dimension $1 \le j \le s$ is partitioned into $b_j^{d_j}$ uniform intervals for fixed $d_j \in \mathbb{N}_0$. Now, given coordinates $(p_1, \dots, p_s)$ of such a resulting interval, where $0 \le p_j < b_j^{d_j}$, the indices

$$l_j := \Phi_{b_j}^{-1}\left(\frac{p_j}{b_j^{d_j}}\right) \in \{0, \dots, b_j^{d_j}\}$$

uniquely identify an index $i \in \{0, \dots, \prod_{j=1}^s b_j^{d_j} - 1\}$ specified by

$$l_j \equiv i \mod b_j^{d_j}, \tag{3}$$

because the bases $b_1, \dots, b_s$ have been chosen relatively prime. Consequently the prime powers $b_j^{d_j}$ are relatively prime as well and therefore the simultaneous solution of the Eq. 3 is provided by the Chinese remainder theorem [1, Sect. 31.5].

With $m_j := \left(\prod_{k=1}^s b_k^{d_k}\right) / b_j^{d_j}$ and the multiplicative inverse $\left(m_j^{-1} \bmod b_j^{d_j}\right)$ the index

$$i = \left(\sum_{j=1}^{s} l_j \cdot m_j \left(m_j^{-1} \bmod b_j^{d_j}\right)\right) \bmod \prod_{j=1}^{s} b_j^{d_j} \tag{4}$$

can be computed efficiently by means of the extended Euclidean algorithm [1, Sect. 31.2]. Immediate consequences are that

1. The first $\prod_{j=1}^s b_j^{d_j}$ points are stratified such that there is exactly one point in each stratum and that

2. All Halton sequence points with indices $i + t \cdot \prod_{j=1}^{s} b_j^{d_j}$, $t \in \mathbb{N}_0$, fall into the same stratum.

Storing a lookup table for the offsets i per stratum [11] is simple, however, the size $\prod_{j=1}^{s} b_j^{d_j}$ of the lookup table can be prohibitive even in $s = 2$ dimensions. It is much more efficient to compute the subsequence offset i by Eq. 4 for a selected stratum, because only s multiplicative inverses need to be stored once.

4 Enumerating Digital (t, s)-Sequences per Elementary Interval

Opposite to Halton's construction, the components

$$x_i^{(j)} = \begin{pmatrix} b^{-1} \\ b^{-2} \\ \vdots \end{pmatrix}^T \left[C^{(j)} \begin{pmatrix} a_0(i) \\ a_1(i) \\ \vdots \end{pmatrix} \right] \in [0, 1), \tag{5}$$

of digital (t, s)-sequences [13] are all generated in the same base b, while the matrix-vector multiplication takes place in a finite field. For finite fields other than $\mathbb{Z}_b$, the digits need to be mapped to the finite field and the resulting vector needs to be mapped back [13], which has been omitted for the sake of clarity. Eq. 1 is an illustrative example, where the generator matrix $C^{(j)}$ is the infinite unit matrix.

The stratification properties resulting from such a construction are illustrated in Fig. 2 and are formalized by

Definition 1 (see [13, p. 48]). An interval of the form

$$E(p_1, \ldots, p_s) := \prod_{j=1}^{s} \left[p_j b^{-d_j}, (p_j + 1) b^{-d_j} \right) \subseteq [0, 1)^s$$

for $0 \leq p_j < b^{d_j}$ and integers $d_j \geq 0$ is called an *elementary interval in base b*.

Given the numbers d_j of digits that determine the number of intervals b^{d_j} in dimension j and the elementary interval $E(p_1, \ldots, p_s)$, we have

$$\begin{pmatrix} C^{(1)}_{[(1,1),(d_1, \sum_{j=1}^{s} d_j + e)]} \\ \vdots \\ C^{(s)}_{[(1,1),(d_s, \sum_{j=1}^{s} d_j + e)]} \end{pmatrix} \cdot \begin{pmatrix} a_0(i) \\ \vdots \\ a_{\sum_{j=1}^{s} d_j - 1}(i) \\ a_0(q) \\ \vdots \\ a_{e-1}(q) \end{pmatrix} = \begin{pmatrix} a_{d_1 - 1}(p_1) \\ \vdots \\ a_0(p_1) \\ \vdots \\ a_{d_s - 1}(p_s) \\ \vdots \\ a_0(p_s) \end{pmatrix} \tag{6}$$

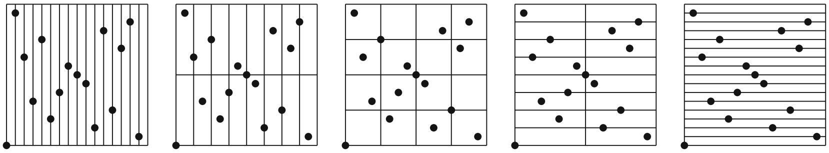


Fig. 2 All kinds of elementary intervals with area $\frac{1}{16}$ for $s = b = 2$. In this case the set of elementary intervals in the middle consists of square strata. The first $2^4 = 16$ points of Sobol's $(0, 2)$-sequence, which form a $(0, 4, 2)$-net in base $b = 2$, are superimposed over each set of elementary intervals.

for the $(q + 1)^{\text{st}}$ point in that elementary interval, where q constitutes the e most significant digits of the index i of that point and the shorthand

$$C^{(j)}_{[(u,v),(u',v')]} := \begin{matrix} c^{(j)}_{u,v} & c^{(j)}_{u,v+1} & \cdots & c^{(j)}_{u,v'} \\ \vdots & \vdots & \ddots & \vdots \\ c^{(j)}_{u',v} & c^{(j)}_{u',v+1} & \cdots & c^{(j)}_{u',v'} \end{matrix}$$

is used to select a block from the first d_j rows of $C^{(j)}$. As $a_0(q), \ldots, a_{e-1}(q)$ are specified by q, rearranging yields

$$\underbrace{\begin{pmatrix} C^{(1)}_{[(1,1),(d_1,\sum_{j=1}^{s} d_j)]} \\ \vdots \\ C^{(s)}_{[(1,1),(d_s,\sum_{j=1}^{s} d_j)]} \end{pmatrix}}_{A} \cdot \begin{pmatrix} a_0(i) \\ \vdots \\ a_{\sum_{j=1}^{s} d_j - 1}(i) \end{pmatrix}$$

$$= \begin{pmatrix} a_{d_1-1}(p_1) \\ \vdots \\ a_0(p_1) \\ \vdots \\ a_{d_s-1}(p_s) \\ \vdots \\ a_0(p_s) \end{pmatrix} - \begin{pmatrix} C^{(1)}_{[(1,\sum_{j=1}^{s} d_j+1),(d_1,\sum_{j=1}^{s} d_j+e)]} \\ \vdots \\ C^{(s)}_{[(1,\sum_{j=1}^{s} d_j+1),(d_s,\sum_{j=1}^{s} d_j+e)]} \end{pmatrix} \cdot \begin{pmatrix} a_0(q) \\ \vdots \\ a_{e-1}(q) \end{pmatrix}, \qquad (7)$$

which can be solved uniquely for the index digits $a_0(i), \ldots, a_{\sum_{j=1}^{s} d_j - 1}(i)$ if $\det(A) \neq 0$.

Upon existence, the inverse A^{-1} is computed once and stored for computing the indices of all samples, which in fact just costs about as much as evaluating an additional component of the sequence.

4.1 $(0, s)$-Sequences

The general definitions of (t, m, s)-nets and (t, s)-sequences in base b are based on the concept of elementary intervals (for a profound introduction see [13, Chap. 4]):

Definition 2 (see [13, Definition 4.1]). For integers $0 \leq t \leq m$, a (t, m, s)-*net in base* b is a point set of b^m points in $[0, 1)^s$ such that there are exactly b^t points in each elementary interval E with volume b^{t-m}.

Definition 3 (see [13, Definition 4.2]). 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* b if, for all integers $k \geq 0$ and $m > t$, the point set $\mathbf{x}_{kb^m}, \ldots, \mathbf{x}_{(k+1)b^m-1}$ is a (t, m, s)-net in base b.

According to these definitions, a $(0, s)$-sequence is a sequence of $(0, m, s)$-nets as illustrated in Fig. 3. This especially includes $(0, ms, s)$-nets, where in each hypercube shaped elementary intervals of side length b^{-m}, there is exactly one point.

For the case of digital constructions, as for example the construction by Faure [4], the generator matrices $C^{(j)}$ of $(0, s)$-sequences in base b thus yield a unique solution of Eq. 7. Note that $(0, s)$-sequences can only exist for $s \leq b$ [13, Corollary 4.24, p. 62].

Often integro-approximation problems expose a structure that matches uniform hypercubes like for example pixels of an image. Out of the elementary interval therefore hypercubes with $d_j = m$ are most interesting for applications. Enumerating b^e points per elementary interval thus results in $(b^m)^s \cdot b^e = b^{ms+e}$ points requiring $ms + e$ digits in total.

4.2 Sobol' Sequence

As opposed to Faure's construction, Sobol's construction [16] is restricted to base $b = 2$, which allows for $t = 0$ only up to $s = 2$ dimensions. However, the restriction to base $b = 2$ enables the use of efficient bit-vector operations [6, 18], which is not possible for $b > 2$.

The sequence can be constructed for any dimension and in fact each component is a $(0, 1)$-sequence in base 2 itself. A description of how to compute the binary generator matrices can be found in [8, 9].

In addition, the first two components form a $(0, 2)$-sequence in base 2 (for an efficient implementation see [12]). As a consequence the first 2^{2m} two-dimensional points are stratified such that there is exactly one point in each voxel of a $2^m \times 2^m$ regular grid over $[0, 1)^2$ as illustrated in Fig. 3.

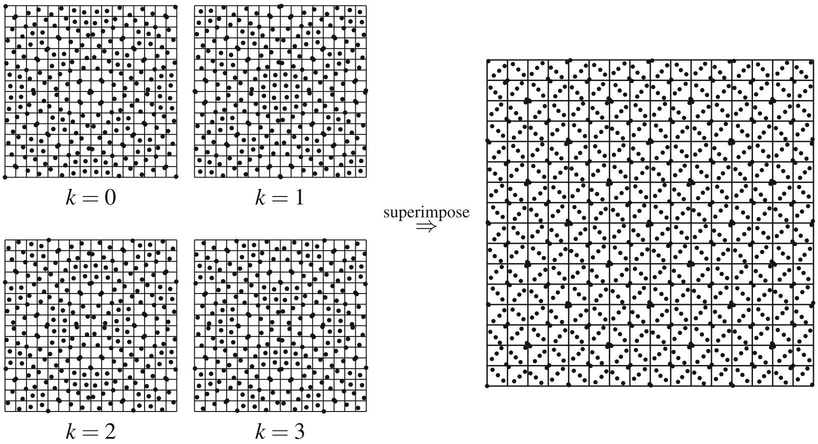


Fig. 3 Since the Sobol' sequence is a $(0, 2)$-sequence, each block of points $\mathbf{x}_{k\cdot 2^8}, \dots, \mathbf{x}_{(k+1)2^8-1}$ constitutes a $(0, 8, 2)$-net in base 2 for $k \geq 0$. Consequently exactly one sample falls into each of the 16 × 16 pixels shown here. When the four consecutive sample blocks shown here are superimposed, there are four samples in each pixel. Our algorithm allows for directly enumerating these samples based on the pixel-coordinates. Note that while in this two-dimensional projection of the Sobol' sequence there are clearly some very regular patterns, the sequence is highly uniformly distributed when more dimensions are considered.

5 Consistent Image Synthesis

Partitioning the unit cube into uniform, axis-aligned intervals results in a number of strata that is exponential in the dimension s. Hence an implementation requires special attention in order to avoid overflows in standard integer arithmetic. We therefore provide illustrative source code [17] in the Python programming language, which transparently handles arbitrarily long integers.

In practice, the enumeration algorithms for both the Halton sequence and the (t, s)-sequences are useful only in small dimensions, as for example computing the average color of pixels for image synthesis. For that purpose the numbers d_j of digits are chosen such that the resulting numbers $b_j^{d_j}$ or b^{d_j}, respectively, of strata are larger or equal to the number of pixels along each dimension. While square pixels directly match the square elementary intervals of $(0, 2m, 2)$-nets from $(0, 2)$-sequences (see Fig. 3), the components of the Halton sequence need to be scaled individually per dimension [11] as illustrated in Fig. 1.

Similar to [11], the entire image plane now can be sampled using only one quasi-Monte Carlo sequence, while still being able to control the sampling rate per pixel. Aside from the first two dimensions, further components are used for sampling the remaining dimensions of the integrand. This includes depth of field, area light sampling, BSDF sampling, etc. [15]. Therefore the quasi-Monte Carlo sequence needs to be extensible in the dimension like for example the Halton or Sobol' sequence.

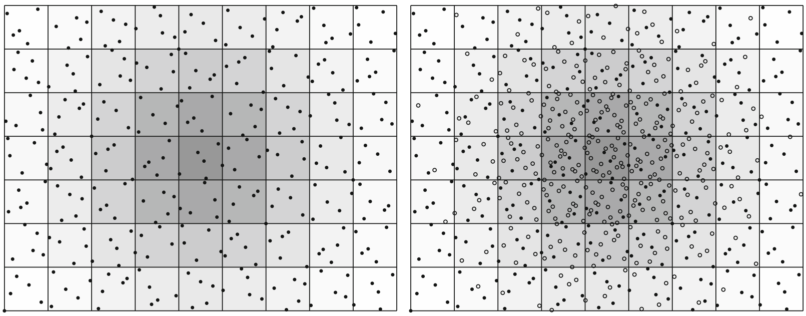


Fig. 4 On the *left*, five samples (depicted by *filled circles*) have been generated in each of the 9 × 7 pixels. Note that both the sample distributions inside each of the pixels and also across pixels is very uniform (in fact of low discrepancy). On the *right*, samples (depicted by *stroked circles*) have been added according to the underlying density proportional to the intensity level of a pixel, depicted by its *gray coloring*. Note that these additional samples naturally fill the remaining space. The overall distribution globally remains well distributed although additional samples have been added locally.

In contrast to [3, 11] sampling per pixel is not terminated by an error threshold. Instead pixel-adaptive sampling is realized by determining a number of samples per pixel based on an error estimate, sampling each pixel according to that *speed*, and repeating this procedure, which results in a consistent integro-approximation algorithm as illustrated in Fig. 4. In particular, this progressive algorithm enables strictly deterministic pixel-adaptive sampling in parallel computing environments at the cost of storing only the current number of samples per pixel.

In addition and at any point of the progressive computation a user can define pixel regions of high importance. More samples will be placed in those regions. Even with this user interaction, the determinism is not lost, i.e., if the image is accumulated up to a certain number of samples for all pixels afterwards, the user interaction does not change the result.

5.1 Enumerating the Sobol' Sequence in Pixels

The Sobol' sequence can be enumerated much more efficiently, if whole $(0, 2m, 2)$-nets (see Fig. 3) are generated instead of single points. In order to explain the optimization, the index

$$i = \sum_{l=0}^{\infty} a_l(i) \cdot 2^l = \sum_{l=2m}^{\infty} a_l(i) \cdot 2^l + \underbrace{\sum_{l=m}^{2m-1} a_l(i) \cdot 2^l}_{\text{MSB}} + \underbrace{\sum_{l=0}^{m-1} a_l(i) \cdot 2^l}_{\text{LSB}}$$

is partitioned into three parts: The m least significant bits (LSB), the m most significant bits, and the remaining bits.

Now the points can be determined as follows: For each component (5) of each $(0, 2m, 2)$-net, two tables with 2^m entries each are computed: The first table stores the results of the matrix-vector multiplications for $0 \leq i \leq 2^m - 1$, while the second stores the results for $i = k \cdot 2^m$ for $0 \leq k \leq 2^m - 1$. A component for an arbitrary value of i then is found by looking up the entry from the first table using the LSB of i, looking up the entry from the second table using the MSB of i, and the result of the matrix-vector multiplication using the remaining bits, all combined using an exclusive-or operation. As compared to evaluating Eq. 5 for each single component, the lookup tables save a lot of operations and can be considered an extension of the initial ideas in [11, Sect. 2.1].

Before applying this optimization to efficiently determine the index i of each point of a $(0, 2m, 2)$-net of the Sobol's sequence (see Fig. 3), the solution of Eq. 6 for the first two components needs to be established:

Given integer pixel coordinates (p_1, p_2), with $0 \leq p_1, p_2 < 2^m$, the m least significant bits $a_0(i), \ldots, a_{m-1}(i)$ of the index i are determined by applying the inverse of $C^{(1)}$ to the bits of p_1. Then the bits of p_2 are combined with $C^{(2)}$ multiplied by the just computed least significant bits using an exclusive-or operation. Applying the inverse of $C^{(2)}$ to the result yields the most significant bits $a_m(i), \ldots, a_{2m-1}(i)$ of the index i.

By Sobol's construction, $C^{(1)}$ is a unit matrix, while $C^{(2)}$ is not, which is the reason for correcting the bits of p_2 by subtracting the contribution of the least significant bits to the most significant bits.

The optimization now consists in replacing all matrix-vector multiplications by table lookups. This requires to compute the lookup tables of size 2^m for each of the $(0, 2m, 2)$-nets.

The resulting implementation [17] in fact is very simple. Note that special care has to be taken in order to avoid overflows due to insufficient word width.

6 Conclusion

We derived efficient algorithms to enumerate low discrepancy sequences in elementary intervals resulting from radical inversion. These algorithms can be used for consistent deterministic parallel quasi-Monte Carlo integro-approximation.

Instead of considering all elementary intervals, it is interesting to restrict observations to $(\mathscr{M}, \mu)$-uniform point sets as introduced in [14], which includes the interesting question, whether rank-1 lattice sequences can be efficiently enumerated inside the sets of $\mathscr{M}$ [11].

References

1. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, Second Edition. MIT Press (2001)
2. van der Corput, J.: Verteilungsfunktionen. Proc. Ned. Akad. v. Wet. **38**, 813–821 (1935)
3. Dammertz, H., Hanika, J., Keller, A., Lensch, H.: A hierarchical automatic stopping condition for Monte Carlo global illumination. In: Proc. of the WSCG 2009, pp. 159–164 (2009)
4. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. **41**(4), 337–351 (1982)
5. Faure, H.: Good permutations for extreme discrepancy. J. Number Theory **42**, 47–56 (1992)
6. Grünschloß, L.: Motion blur. Master's thesis, Ulm University (2008)
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. Joe, S., Kuo, F.: Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. ACM Trans. Math. Softw. **29**(1), 49–57 (2003)
9. Joe, S., Kuo, F.: Constructing Sobol' sequences with better two-dimensional projections. SIAM Journal on Scientific Computing **30**(5), 2635–2654 (2008)
10. Keller, A.: Strictly deterministic sampling methods in computer graphics. SIGGRAPH 2003 Course Notes, Course #44: Monte Carlo Ray Tracing (2003)
11. Keller, A.: Myths of computer graphics. In: D. Talay, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods, pp. 217–243. Springer (2004)
12. Kollig, T., Keller, A.: Efficient multidimensional sampling. Computer Graphics Forum **21**(3), 557–563 (2002)
13. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
14. Niederreiter, H.: Error bounds for quasi-Monte Carlo integration with uniform point sets. J. Comput. Appl. Math. **150**, 283–292 (2003)
15. Pharr, M., Humphreys, G.: Physically Based Rendering: From Theory to Implementation, 2nd edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2010)
16. Sobol', I.: On the distribution of points in a cube and the approximate evaluation of integrals. Zh. vychisl. Mat. mat. Fiz. **7**(4), 784–802 (1967)
17. Downloadable source code package for this article. http://gruenschloss.org/sample-enum/sample-enum-src.zip
18. Wächter, C.: Quasi-Monte Carlo light transport simulation by efficient ray tracing. Ph.D. thesis, Ulm University (2008)

Importance Sampling Estimation of Joint Default Probability under Structural-Form Models with Stochastic Correlation

Chuan-Hsiang Han

Abstract This paper aims to estimate joint default probabilities under the structural-form model with a random environment; namely stochastic correlation. By means of a singular perturbation method, we obtain an asymptotic expansion of a two-name joint default probability under a fast mean-reverting stochastic correlation model. The leading order term in the expansion is a joint default probability with an *effective* constant correlation. Then we incorporate an efficient importance sampling method used to solve a first passage time problem. This procedure constitutes a *homogenized* importance sampling to solve the full problem of estimating the joint default probability with stochastic correlation models.

1 Introduction

Estimation of a joint default probability under the structural-form model typically requires solving a first passage time problem. Black and Cox [1] and Zhou [17] provided financial motivations and technical details on the first passage time approach for one and two dimensional cases, respectively.

A high-dimensional setup of the first passage time problem is as follows. Assume that a credit portfolio includes n reference defaultable assets or names. Each asset value, S_{it} $1 \leq i \leq n$, is governed by

$$dS_{it} = \mu_i S_{it} dt + \sigma_i \, S_{it} dW_{it}, \tag{1}$$

where μ_i denotes a constant drift rate, σ_i denotes a constant volatility and the driving innovation dW_{it} is an infinitesimal increment of a Brownian motion (Wiener process) W_i with the instantaneous constant correlation

C.-H. Han
Department of Quantitative Finance, National Tsing Hua University, No. 101, Kung-Fu Rd, Section 2, Hsinchu 30013, Taiwan (ROC)
e-mail: chhan@mx.nthu.edu.tw

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_22, © Springer-Verlag Berlin Heidelberg 2012

$$d\left\langle W_i, W_j \right\rangle_t = \rho_{ij}\,dt.$$

Each name also has a barrier, B_i, $1 \leq i \leq n$, and default happens at the first time S_{it} falls below the barrier level. That is, the ith default time τ_i is defined by the first hitting time

$$\tau_i = \inf\{t \geq 0 : S_{it} \leq B_i\}. \tag{2}$$

Let the filtration $\mathscr{F}_{t\geq 0}$ be generated by all $S_{it}, i = 1, \cdots, n$ under a probability measure $I\!P$. At time 0, the joint default probability with a terminal time T is defined by

$$DP = I\!E\left\{\Pi_{i=1}^{n}\mathbf{I}(\tau_i \leq T)|\mathscr{F}_0\right\}. \tag{3}$$

Due to the high dimensional nature of this problem ($n = 125$ in a standard credit derivative [3], for example), Monte Carlo methods are very useful tools for computation. However, the basic Monte Carlo method converges slowly to the probability of multiple defaults defined in (3). We will review an efficient importance sampling scheme discussed in Han [10] to speed up the computation. This method is asymptotically optimal in reducing variance of the new estimator.

Engle [6] revealed the impact of correlation between multiple asset dynamics. A family of discrete-time correlation models called dynamic conditional correlation (DCC) has been widely applied in theory and practice. Hull et al. [15] examined the effect of random correlation in continuous time and suggested stochastic correlation for the structural-form model. This current paper studies the joint default probability estimation problem under the structural-form model with stochastic correlation. For simplicity, we consider a two-dimensional case, $n = 2$. This problem generalizes Zhou's study [17] with constant correlation.

Note that under stochastic correlation models, there exists no closed-form solution for the two-name joint default probability. A two-step approach is proposed to solve this estimation problem. First, we apply a singular perturbation technique [4] and derive an asymptotic expansion of the joint default probability. Its leading order term is a default probability with an *effective* constant correlation so that the limiting problem becomes the standard setup of the first passage time problem. Second, given the accuracy of this asymptotic approximation, we develop a *homogenized* likelihood function for measure change. It allows that the efficient importance sampling method [7, 10] can be applied for estimation of the two-name joint default probability under stochastic correlation models. Results of numerical simulation show that estimated joint default probabilities are sensitive to the change in correlation and our proposed method is efficient and robust even when the mean-reverting speed is not in a small regime.

The organization of this paper is as follows. Section 2 presents an asymptotic expansion of the joint default probability under a fast mean-reverting correlation by means of the singular perturbation analysis. Section 3 reviews the efficient importance sampling method to estimate joint default probabilities under the classical structural-form model with constant correlation. Section 4 constructs a homogenized importance sampling method to solve the full problem.

2 Stochastic Correlation Model: Two Dimensional Case

The closed-form solution of a two-name joint default probability under a constant correlation model is given in [2]. Assume that asset prices (S_{1t}, S_{2t}) driven by two geometric Brownian motions with a constant correlation $\rho, -1 \leq \rho \leq 1$ are governed by

$$\begin{aligned} dS_{1t} &= \mu_1\, S_{1t} dt + \sigma_1 S_{1t} dW_{1t} \\ dS_{2t} &= \mu_2\, S_{2t} dt + \sigma_2 S_{2t} (\rho dW_{1t} + \sqrt{1-\rho^2} dW_{2t}), \end{aligned}$$

following the usual setup in (1). When the default boundary is deterministic of an exponential type $Be^{\lambda t}$, each default time τ_i can be defined as

$$\tau_i = \inf\{t \geq 0; S_{it} \leq B_i e^{\lambda_i t}\} \tag{4}$$

for $i \in \{1, 2\}$. This setup is slightly more general than our constant barriers (2) but it causes no extra difficulty when log-transformation is applied. No initial default, i.e., $S_{i0} > B_i$ for each i, is assumed to avoid the trivial case. The joint default probability defined by

$$P(0, x_1, x_2) = I\!P\,(\tau_1 \leq T, \tau_2 \leq T)$$

can be expressed as

$$P(0, x_1, x_2) = P_1(0, x_1) + P_2(0, x_2) - Q^{1,2}(0, x_1, x_2) \tag{5}$$

where $P_i := I\!P\,(\tau_i \leq T)$ denotes the ith marginal default probability and $Q^{1,2} := I\!P\,(\tau_1 \leq T \text{ or } \tau_2 \leq T)$ denotes the probability that at least one default happens. The closed-form formula for each $P_i, i \in \{1, 2\}$, is

$$P_i = \mathcal{N}\left(-\frac{d_i}{\sqrt{T}} - \frac{\mu_i - \lambda_i}{\sigma_i}\sqrt{T}\right) + e^{\frac{2(\lambda_i - \mu_i)d_i}{\sigma_i}} \mathcal{N}\left(-\frac{d_i}{\sqrt{T}} + \frac{\mu_i - \lambda_i}{\sigma_i}\sqrt{T}\right),$$

where $d_i = \frac{\ln(S_0^i/K_i)}{\sigma_i}$. The last term $Q^{1,2}$ can be expressed as a series of modified Bessel functions (see [2] for details) and we skip it here.

Hull et al. [15] proposed a mean-reverting stochastic correlation for the structural-form model, and they found empirically a better fit to spreads of credit derivatives. We assume that the correlation process $\rho_t = \rho(Y_t)$ is driven by a mean-reverting process Y_t such as the Ornstein-Uhlenbeck process. A small time scale parameter ε is incorporated into the driving correlation process Y_t so that the correlation changes rapidly compared with the asset dynamics of S. The two-name dynamic system with a fast mean-reverting stochastic correlation is described by

$$\begin{aligned} dS_{1t} &= \mu_1 S_{1t} dt + \sigma_1 S_{1t} dW_{1t} \\ dS_{2t} &= \mu_2 S_{2t} dt + \sigma_2 S_{2t} \left(\rho(Y_t) dW_{1t} + \sqrt{1-\rho^2(Y_t)} dW_{2t} \right) \\ dY_t &= \frac{1}{\varepsilon}(m - Y_t)dt + \frac{\sqrt{2}\beta}{\sqrt{\varepsilon}} dZ_t, \end{aligned} \tag{6}$$

where the correlation function $\rho(\cdot)$ is assumed smooth and bounded in $[-1, 1]$, and the driving Brownian motions W's and Z are assumed to be independent of each other. The joint default probability under a fast mean-reverting stochastic correlation model is defined as

$$P^{\varepsilon}(t, x_1, x_2, y) := I\!E \left\{ \Pi_{i=1}^{2} \mathbf{I}\{ \min_{t \le u \le T} S_{iu} \le B_i \} | S_{1t} = x_1, S_{2t} = x_2, Y_t = y \right\}, \tag{7}$$

provided no default before time t.

From the modeling point of view, the assumption of a mean-reverting correlation is consistent with DCC model, see Engle [6], in which a quasi-correlation is often assumed mean-reverting. From the statistical point of view, a Fourier transform method developed by Malliavin and Mancino [16] provides a nonparametric way to estimate dynamic volatility matrix in the context of a continuous semi-martingale. Our setup of the stochastic correlation model (6) satisfies assumptions in [16]. This implies that model parameters of volatility and correlation defined in (6) can be estimated via the Fourier transform method. Moreover, from the computational point of view, stochastic correlation introduces a random environment into the classical first passage time problem in dynamic models. This situation is similar to Student-t distribution over the Gaussian distribution in static copula models [5] arising from reduced-form models in credit risk. Han and Wu [13] have recently solved this static Gaussian copula problem with a random environment; namely, Student-t copula. In contrast, the stochastic correlation estimation problem considered in this paper fills a gap of research work for a random environment in dynamic models.

2.1 Formal Expansion of The Perturbed Joint Default Probability

By an application of Feynman-Kac formula, $P^{\varepsilon}(t, x_1, x_2, y)$ solves a three-dimensional partial differential equation (PDE)

$$\left(\frac{1}{\varepsilon}\mathscr{L}_0 + \mathscr{L}_1\right) P^\varepsilon(t, x_1, x_2, y) = 0, \tag{8}$$

where partial differential operators are

$$\begin{aligned}
\mathscr{L}_0 &= \beta^2 \frac{\partial^2}{\partial y^2} + (m - y)\frac{\partial}{\partial y} \\
\mathscr{L}_1(\rho(y)) &= \mathscr{L}_{1,0} + \rho(y)\mathscr{L}_{1,1} \\
\mathscr{L}_{1,0} &= \frac{\partial}{\partial t} + \sum_{i=1}^{2} \frac{\sigma_i^2 x_i^2}{2}\frac{\partial^2}{\partial x_i^2} + \sum_{i=1}^{2} \mu_i x_i \frac{\partial}{\partial x_i} \\
\mathscr{L}_{1,1} &= \sigma_1\sigma_2 x_1 x_2 \frac{\partial^2}{\partial x_1 \partial x_2}.
\end{aligned}$$

The terminal condition is $P^\varepsilon(T, x_1, x_2, y) = I_{\{x_1 \le B_1\}} I_{\{x_2 \le B_2\}}$ and two boundary conditions are $P^\varepsilon(t, B_1, x_2, y) = P^\varepsilon(t, x_1, B_2, y) = 0$.

Suppose that the perturbed joint default probability admits the following expansion

$$P^\varepsilon(t, x_1, x_2, y) = \sum_{i=0}^{\infty} \varepsilon^i P_i(t, x_1, x_2, y).$$

Substituting this into (8),

$$\begin{aligned}
0 &= \left(\frac{1}{\varepsilon}\mathscr{L}_0 + \mathscr{L}_1\right)\left(P_0 + \varepsilon P_1 + \varepsilon^2 P_2 + \cdots\right) \\
&= \frac{1}{\varepsilon}\left(\mathscr{L}_0 P_0\right) + \left(\mathscr{L}_0 P_1 + \mathscr{L}_1 P_0\right) + \varepsilon\left(\mathscr{L}_0 P_2 + \mathscr{L}_1 P_1\right) \\
&\quad + \varepsilon^2\left(\mathscr{L}_0 P_3 + \mathscr{L}_1 P_2\right) + \cdots
\end{aligned}$$

is obtained. By equating each term in order of ε to zero, a sequence of PDEs must be solved.

For the $\mathscr{O}(\frac{1}{\varepsilon})$ term, $\mathscr{L}_0 P_0(t, x_1, x_2, y) = 0$. One can choose P_0 as variable $y-$independent. For the $\mathscr{O}(1)$ term, $(\mathscr{L}_0 P_1 + \mathscr{L}_1 P_0)(t, x_1, x_2, y) = 0$, which is a Poisson equation. Because $\mathscr{L}_0$ is the generator of an ergodic process Y_t, by centering condition we can obtain $< \mathscr{L}_1 > P_0 = 0$. The notation $< \cdot >$ means the averaging with respect to the invariance measure of the ergodic process Y. Thus the leading order term P_0 solves the *homogenized* PDE:

$$(\mathscr{L}_{1,0} + \bar{\rho}\mathscr{L}_{1,1}) P_0(t, x_1, x_2) = 0,$$

where $\bar{\rho} =< \rho(y) >_{OU}= \int \rho(y) \frac{1}{\sqrt{2\pi}\nu} e^{-\frac{(y-m)^2}{2\nu^2}} dy$ with the terminal condition is $P_0(T, x_1, x_2) = I_{\{x_1 \le B_1\}} I_{\{x_2 \le B_2\}}$ and two boundary conditions are $P_0(t, B_1, x_2) = P_0(t, x_1, B_2) = 0$. The closed-form solution of $P_0(t, x_1, x_2)$ exists with a similar formulation presented in (5).

Combining $\mathscr{L}_0 P_1 + \mathscr{L}_1 P_0 = 0$ with $< \mathscr{L}_1 > P_0 = 0$, we obtain $\mathscr{L}_0 P_1 = -(\mathscr{L}_1 P_0 - < \mathscr{L}_1 > P_0)$ such that

$$\begin{aligned} P_1(t,x_1,x_2,y) &= -\mathscr{L}_0^{-1}\,(\mathscr{L}_1 - < \mathscr{L}_1 >)\,P_0(t,x_1,x_2) \\ &= -\mathscr{L}_0^{-1}\,(\rho(y)-\bar{\rho})\,\mathscr{L}_{1,1}P_0(t,x_1,x_2) \\ &= -\varphi(y)\sigma_1\sigma_2 x_1 x_2 \frac{\partial^2}{\partial x_1 \partial x_2} P_0(t,x_1,x_2), \end{aligned}$$

where $\varphi(y)$ is assumed to solve the Poisson equation $\mathscr{L}_0\,\varphi(y) = \rho(y) - \bar{\rho}$.

Similar argument goes through successive expansion terms. We skip the lengthy derivation but simply summarize each successive term for $n \geq 0$

$$P_{n+1}(t,x_1,x_2,y) = \sum_{i \geq 0, j \geq 1}^{i+j=n+1} \varphi_{i,j}^{(n+1)}(y)\,\mathscr{L}_{1,0}^{i}\,\mathscr{L}_{1,1}^{j}\,P_n,$$

where a sequence of Poisson equations must be solved from

$$\begin{aligned} \mathscr{L}_0\,\varphi_{i+1,j}^{(n+1)}(y) &= \left(\varphi_{i,j}^{(n)}(y) - < \varphi_{i,j}^{(n)}(y) >\right) \\ \mathscr{L}_0\,\varphi_{i,j+1}^{(n+1)}(y) &= \left(\rho(y)\,\varphi_{i,j}^{(n)}(y) - < \rho\,\varphi_{i,j}^{(n)} >\right). \end{aligned}$$

Hence, a recursive formula for calculating the joint default probability $P^{\varepsilon} = P_0 + \varepsilon P_1 + \varepsilon^2 P_2 + \cdots$ is derived.

In summary, we have formally derived that

$$P^{\varepsilon}(t,x_1,x_2,y) = P_0(t,x_1,x_2;\bar{\rho}) + \mathscr{O}(\varepsilon), \tag{9}$$

where the accuracy result can be obtained by a regularization technique presented in [14].

Remark 1. The asymptotic expansion presented in this section can be generalized to multi-dimensional cases.

3 Efficient Importance Sampling for the First Passage Time Problem

In this section, we review the efficient importance sampling scheme proposed in [10] for the first passage time problem (3) in order to improve the convergence of Monte Carlo simulation. The basic Monte Carlo simulation approximates the joint default probability defined in (3) by the following estimator

$$DP \approx \frac{1}{N} \sum_{k=1}^{N} \Pi_{i=1}^{n} \mathbf{I} \left(\tau_i^{(k)} \leq T \right), \tag{10}$$

where $\tau_i^{(k)}$ denotes the kth i.i.d. sample of the ith default time defined in (4) and N denotes the total number of simulations.

By Girsanov theorem, one can construct an equivalent probability measure $\tilde{P}$ defined by the following Radon-Nikodym derivative

$$\frac{dP}{d\tilde{P}} = Q_T(h.) = \exp\left(\int_0^T h(s, S_s) \cdot d\tilde{W}_s - \frac{1}{2} \int_0^T ||h(s, S_s)||^2 ds \right), \tag{11}$$

where we denote by $S_s = (S_{1s}, \cdots, S_{ns})$ the state variable (asset value process) vector and $\tilde{W}_s = \left(\tilde{W}_{1s}, \cdots, \tilde{W}_{ns}\right)$ the vector of standard Brownian motions, respectively. The function $h(s, S_s)$ is assumed to satisfy Novikov's condition such that $\tilde{W}_t = W_t + \int_0^t h(s, S_s) ds$ is a vector of Brownian motions under $\tilde{P}$.

The importance sampling scheme proposed in [10] selects a **constant** vector $h = (h_1, \cdots, h_n)$ which satisfies the following n conditions

$$\tilde{E}\left\{ S_{iT} | \mathscr{F}_0 \right\} = B_i, i = 1, \cdots, n. \tag{12}$$

These equations can be simplified by using the explicit log-normal density of S_{iT}, so the following sequence of linear equations for h_i's:

$$\Sigma_{j=1}^{i} \rho_{ij} h_j = \frac{\mu_i}{\sigma_i} - \frac{\ln B_i / S_{i0}}{\sigma_i \, T}, i = 1, \cdots, n, \tag{13}$$

can be considered. If the covariance matrix $\Sigma = (\rho_{ij})_{1 \leq i, j, \leq n}$ is non-singular, the vector h exists uniquely and the equivalent probability measure $\tilde{P}$ is uniquely determined. The joint default probability defined from the first passage time problem (see (3)) can be estimated from

$$DP = \tilde{E}\left\{ \Pi_{i=1}^{n} \mathbf{I} \left(\tau_i \leq T \right) \, Q_T(h) | \mathscr{F}_0 \right\} \tag{14}$$

by simulation.

4 Homogenized Importance Sampling Under Stochastic Correlation

The objective of this paper is to estimate the joint default probability defined in (7) under a class of stochastic correlation models. A direct application of the efficient importance sampling described in Sect. 3 is impossible because it requires a constant correlation ρ to solve for the unique h in (13). Fortunately, this hurdle can be overcome by the asymptotic approximation of the joint default probability (see (9)) because its leading-order approximation term has a constant correlation $\bar{\rho}$. As a result, our methodology to estimate the two-name joint default probability with stochastic correlation is simply to apply the efficient importance sampling scheme associated with the *effective* correlation, derived from the singular perturbation analysis. Detailed variance analysis for this methodology is left as a future work. A recent large deviation theory derived in Feng et al. [8] can be a valuable source to provide a guideline for solving this theoretical problem.

Table 1 illustrates estimations of default probabilities of two names under stochastic correlation models by means of the basic Monte Carlo method and the homogenized importance sampling method. It is observed that the two-name joint default probabilities are of order 10^{-2} or 10^{-3}. Though these estimated probabilities are not considered very small, the homogenized importance sampling can still improve the variance reduction ration by 6.25 times at least. Note also that the performance of homogenized importance sampling is very robust to the time scale ε, even it is not in a small regime (for example $\varepsilon = 10$) as the singular perturbation method required.

Next, small probability estimations are illustrated in Table 2. The homogenized importance sampling method provides fairly accurate estimations, say in the 95% confidence interval. The variance reduction rations can raise up to 2,500 times for these small probability estimations. In addition, we observe again the robustness of this importance sampling to time scale parameter ε.

It is also interesting to observe the effect of time scale from these numerical estimation results. When the stochastic correlation is more volatile (small ε), the probability of joint default increases as well. This is consistent with what observed under stochastic volatility models for option pricing [12]. It shows that these estimations from variance reduction methods are sensitive to changes in correlation and volatility. Hence, it is possible to develop a Monte Carlo calibration method [11] allowing model parameters to fit the implied volatility surface [9] or spreads of credit derivatives [3].

Table 1 Two-name joint default probability estimations under a stochastic correlation model are calculated by the basic Monte Carlo (BMC) and the homogenized importance sampling (HIS), respectively. Several time scales ε are given to compare the effect of stochastic correlation. The total number of simulations is 10^4 and an Euler discretization scheme is used by taking time step size $T/400$, where T is 1 year. Other parameters include $S_{10} = S_{20} = 100, \sigma_1 = 0.4, \sigma_2 = 0.4, B_1 = 50, B_2 = 40, Y_0 = m = \pi/4, \beta = 0.5, \rho(y) = |sin(y)|$. Standard errors are shown in parenthesis.

$\alpha = \frac{1}{\varepsilon}$	BMC	HIS
0.1	$0.0037(6 * 10^{-4})$	$0.0032(1 * 10^{-4})$
1	$0.0074(9 * 10^{-4})$	$0.0065(2 * 10^{-4})$
10	$0.011(1 * 10^{-3})$	$0.0116(4 * 10^{-4})$
50	$0.016(1 * 10^{-3})$	$0.0137(5 * 10^{-4})$
100	$0.016(1 * 10^{-3})$	$0.0132(4 * 10^{-4})$

Table 2 Two-name joint default probability estimations under a stochastic correlation model are calculated by the basic Monte Carlo (BMC) and the homogenized importance sampling (HIS), respectively. Several time scales ε are given to compare the effect of stochastic correlation. The total number of simulations is 10^4 and an Euler discretization scheme is used by taking time step size $T/400$, where T is 1 year. Other parameters include $S_{10} = S_{20} = 100, \sigma_1 = 0.4, \sigma_2 = 0.4, B_1 = 30, B_2 = 20, Y_0 = m = \pi/4, \beta = 0.5, \rho(y) = |sin(y)|$. Standard errors are shown in parenthesis.

$\alpha = \frac{1}{\varepsilon}$	BMC	HIS
0.1	$-(-)$	$9.1 * 10^{-7}(7 * 10^{-8})$
1	$-(-)$	$7.5 * 10^{-6}(6 * 10^{-7})$
10	$-(-)$	$2.4 * 10^{-5}(2 * 10^{-6})$
50	$1 * 10^{-4}(1 * 10^{-4})$	$2.9 * 10^{-5}(3 * 10^{-6})$
100	$1 * 10^{-4}(1 * 10^{-4})$	$2.7 * 10^{-5}(2 * 10^{-6})$

Table 3 Four-name joint default probability estimations under a stochastic correlation model are calculated by the basic Monte Carlo (BMC) and the homogenized importance sampling (HIS), respectively. The time scale ε appearing in the stochastic correlation process is fixed as 10. Other parameters are $S_{i0} = 100, i \in \{1, 2, 3, 4\}, \sigma_1 = 0.5, \sigma_2 = 0.4, \sigma_3 = 0.3, \sigma_4 = 0.2, Y_0 = m = 0, \beta = 0.5, \rho(y) = sin(y)$. Standard errors are shown in parenthesis. Two sets of default thresholds B's are chosen to reflect a bigger and a smaller probability of joint defaults, respectively. The total number of simulations is 10^4 and an Euler discretization scheme is used by taking time step size $T/400$, where T is 1 year.

Default thresholds	BMC	HIS
$B_1 = B_2 = B_3 = B_4 = 70$	$0.0019(4 * 10^{-4})$	$0.0021(1 * 10^{-4})$
$B_1 = 30, B_2 = 40, B_3 = 50, B_4 = 60$	$-(-)$	$1.1 * 10^{-7}(2 * 10^{-8})$

Model parameters within Tables 1 and 2 are homogeneous. That is, dynamics (6) of these two firms are indistinguishable because their model parameters are chosen as the same. Here we consider an inhomogeneous case in a higher dimension, say 4, to illustrate the efficiency of our proposed importance sampling method in Table 3. For simplicity, we fix the time scale ε but use varying firm specific model parameters. A factor structure that generalizes dynamics (6) is chosen as $dS_{1t}/S_{1t} = \mu_1 dt + \sigma_1 dW_{1t}$ and $dS_{it}/S_{it} = \mu_i dt + \sigma_i \left(\rho(Y_t)\, dW_{1t} + \sqrt{1 - \rho^2(Y_t)} dW_{it}\right)$ for $i \in \{2, 3, 4\}$.

5 Conclusion

Estimation of joint default probabilities under the structural-form model with stochastic correlation is considered as a variance reduction problem under a random environment. We resolve this problem by proposing a homogenized importance sampling method. It comprises (1) derivation of an asymptotic result by means of the singular perturbation analysis given a fast mean-reverting correlation assumption, and (2) incorporating the efficient importance sampling method from solving the classical first passage time problem. Numerical results show the efficiency and robustness of this homogenized importance sampling method even when the time scale parameter is not in a small regime.

Acknowledgements Work Supported by NSC-99-2115-M007-006-MY2 and TIMS at National Taiwan University. We are grateful to an anonymous referee for helpful comments.

References

1. F. Black and J. Cox, "Valuing Corporate Securities: Some Effects of Bond Indenture Provisions," Journal of Finance, 31(2), 1976, 351–367.
2. T.R. Bielecki and M. Rutkowski, *Credit Risk: Modeling, Valuation and Hedging,* Springer 2002.
3. D. Brigo, A. Pallavicini and R. Torresetti, *Credit Models and the Crisis: A Journey into CDOs, Copulas, Correlations and Dynamic Models,* Wiley. 2010.
4. J. A. Bucklew, *Introduction to rare event simulation*, Springer, 2003.
5. U. Cherubini, E. Luciano, and W. Vecchiato, *Copula Methods in Finance*, Wiley, 2004.
6. R. Engle, *Anticipating Correlations. A New Paradigm for Risk Management,* Princeton University Press. 2009.
7. J.-P. Fouque, G. Papanicolaou, R. Sircar, *Derivatives in Financial Markets with Stochastic Volatility*, Cambridge University Press, 2000.
8. J. Feng, M. Forde, and J.-P. Fouque, "Short maturity asymptotics for a fast mean reverting Heston stochastic volatility model," SIAM Journal on Financial Mathematics, Vol. 1, 2010, 126–141.
9. J. Gatheral, *The volatility surface,* New Jersey: Wiley, 2006.
10. C.H. Han, "Efficient Importance Sampling Estimation for Joint Default Probability: the First Passage Time Problem," Stochastic Analysis with Financial Applications. Editors A. Kohatsu-Higa, N. Privault, and S.-J. Sheu. Progress in Probability, Vol. 65, Birkhauser, 2011.
11. C.H. Han, "Monte Carlo Calibration to Implied Volatility Surface," Working Paper. National Tsing-Hua University.
12. C.H. Han and Y. Lai, "A Smooth Estimator for MC/QMC Methods in Finance," Mathematics and Computers in Simulation, 81 (2010), pp. 536–550.
13. C.H. Han and C.-T. Wu, "Efficient importance sampling for estimating lower tail probabilities under Gaussian and Student's *t* distributions," Preprint. National Tsing-Hua University. 2012.
14. A. Ilhan, M. Jonsson, and R. Sircar, "Singular Perturbations for Boundary Value Problems arising from Exotic Options," SIAM J. Applied Math. 64 (4), 2004.
15. Hull, J., M. Presescu, and A. White, "The Valuation of Correlation-Dependent Credit Derivatives Using a Structural Model," Working Paper, University of Toronto, 2005.
16. P. Malliavin and M.E. Mancino, "A Fourier transform method for nonparametric estimation of multivariate volatilities," The Annals of Statistics, 37, 2009, 1983–2010.
17. C. Zhou, "An Analysis of Default Correlations and Multiple Defaults," The Review of Financial Studies, 14(2), 2001, 555–576.

Spatial/Angular Contributon Maps for Improved Adaptive Monte Carlo Algorithms

Carole Kay Hayakawa, Rong Kong, and Jerome Spanier

Abstract In the field of biomedical optics, use of light to detect cancerous tissue transformations often involves a low probability detector response because tissue is very turbid and scattering is highly forward-peaked. In these applications, we use a contributon map to extend the geometric learning of adaptive Monte Carlo algorithms. The contributon function provides a phase space map of the **lossless** flow of "contributon particles" that necessarily are transported from source to detector. This map is utilized within an adaptive sequential correlated sampling algorithm to lower the variance systematically and provide improved convergence rates over conventional Monte Carlo.

1 Introduction

Diagnostic optical probes often are comprised of a fiber source and detector placed on the surface of the tissue being investigated. Such probes are used to explore whether dysplastic transformations are taking place within the tissue. To obtain estimates of detected reflected light that can account for the complexity of biological tissue, Monte Carlo methods applied to the radiative transport equation (RTE) often provide the only viable solutions.

C.K. Hayakawa
University of California at Irvine, 916 Engineering Tower, Irvine, CA 92697, USA
e-mail: hayakawa@uci.edu

R. Kong
Claremont Graduate University, 150 E 10th Street, Claremont, CA 91711, USA

J. Spanier
Beckman Laser Institute and Medical Clinic, University of California, 1002 Health Sciences Road East, Irvine, CA 92612, USA

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_23, © Springer-Verlag Berlin Heidelberg 2012

However, due to the small size of the detector (in terms of probability), conventional Monte Carlo methods produce small signal to noise ratios there and can often therefore require large amounts of computation. A similar problem exists for adjoint simulations because the source presents a probabilistically small target for adjoint "photons" launched from the detector. In addition, tissue is very turbid and scattering is very forward-peaked. A typical mean scattering length ($1/\mu_s$, where μ_s = scattering coefficient) is 0.1 mm and source-detector separations can range up to 3 cm, so thousands of scattering events can occur between source and detector. These aspects of the problem create a small efficiency factor:

$$\mathrm{Eff}[\xi] = [\sigma_{rel}^2 t]^{-1}$$

where ξ is the random variable estimating detection, σ_{rel} is the relative standard deviation of the detected signal and t is the computer run time. Sometimes $\mathrm{Eff}[\xi]$ is referred to as the Figure of Merit [9].

Using conventional Monte Carlo methods, the only way to improve efficiency is to increase the sample size. However, this only reduces the error at the rate $\frac{1}{\sqrt{N}}$ forecast by the central limit theorem (CLT) [2], where N is the number of photons executed. In order to achieve a gain over CLT rates, sequential correlated sampling (SCS) methods, initially introduced to solve matrix problems by Halton [3] and later extended to continuous transport problems [5–7], can be used to reduce the statistical error **geometrically**. However, when the problem complexity requires a solution in 5, 6 or 7 independent variables, or when the physical problem being modeled is highly heterogeneous, the computational efficiency degrades. Ultimately such problems can be traced to the need for an efficient representation of the transport solution at **every** point and in **every** direction in the phase space. These limitations prompted the development of improved methods [4] that require only the efficient estimation of a small number of linear functionals of the solution (these correspond to estimates of the photons detected) rather than achieving arbitrarily high precision at **every** point in phase space.

In **averaged** sequential correlated sampling (ASCS), we relax the requirement of a pointwise global solution and instead acquire regionwise constant averages of the radiative transport equation (RTE) solution throughout phase space. Of course, the accuracy of the detected signal is then heavily dependent on the phase space decomposition imposed to obtain these averages. To address this issue, our strategy starts with a crude decomposition and then uses the contributon function [11] to refine the decomposition intelligently, adding refinement where transport from source to detector is important and leaving a coarse decomposition of regions less important.

These ideas were applied in [4] in which the combined ASCS and **spatial** contributon maps were used to increase the computational efficiency of 1-dimensional model problem solutions. Here we show that the **angular** contributon map plays an essential role in the phase space refinement.

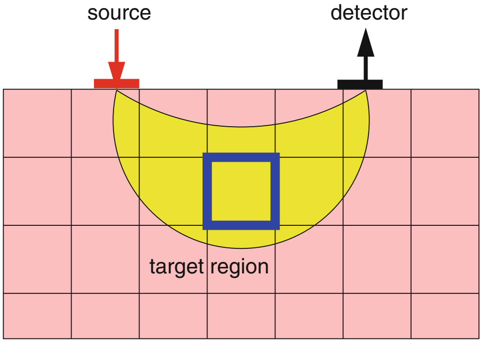


Fig. 1 Optical probe problem consists of a source, a detector and a target region.

2 Problem Description

To diagnose a region of interest in tissue an optical probe is placed on the surface of the tissue. The probe consists of a source of light and a detector measuring the amount of reflected or transmitted light. The goal is to identify tissue transformations that may be taking place below the tissue surface in some region of interest which we call the target region. The description of the probe along with the tissue geometry thus defines a problem with three components: (1) source, (2) detector, and (3) target region (Fig. 1).

The tissue model normally represents the tissue as decomposed into non-overlapping spatial/angular components, or elements, each element representing a tissue type with averaged optical properties. On this **physical** model we superimpose a crude subdivision of the entire phase space for the purpose of defining a **computational** model to produce an initial histogram representation of the RTE solution. For example, to start with, the computational model could be the same as the physical model, which would then define a minimal phase space decomposition that would capture the intrinsic heterogeneity of the tissue being probed. Our solution method aims to address the following questions:

- Is a global solution at **every** location and in **every** direction in phase space necessary for the accurate estimate of detection?
- How coarse/fine must the tissue definition be in various regions to ensure accuracy in the estimates of the detected signal?
- How much can we improve on conventional Monte Carlo estimates of detection in terms of computational efficiency by applying adaptive methods?

We answer these questions using an algorithm that combines averaged sequential correlated sampling with information from a contributon map.

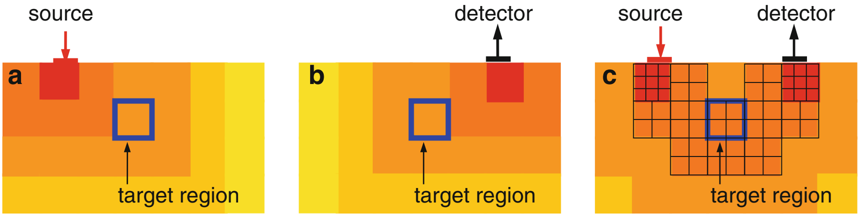


Fig. 2 Solution consists of Part I: (**a**) forward simulation applying ASCS to a crude spatial/angular mesh; Part II: (**b**) adjoint ASCS solution using same mesh, and (**c**) contributon map formed from forward and adjoint solutions to refine mesh.

3 Solution Method

The solution is segmented into two parts (Fig. 2). In Part I, a very crude spatial/angular mesh is imposed on the phase space $\varGamma$ and averaged sequential correlated sampling (ASCS) is used to obtain regionwise constant averages of the spatial/angular flux throughout the tissue. In Part II, the adjoint ASCS solution is then obtained using the same mesh. Using coupled forward/adjoint estimates of the averaged fluxes, a contributon map is formed and identifies those regions/angles needing more refinement. By coupling forward and adjoint RTE solutions, the contributon accounts for phase space regions that incorporate both significant light intensity and likelihood that the light will eventually be detected. A refined mesh based on relative values of the contributon map is then used to improve estimates of the detected signal by applying ASCS to the refined mesh. A detailed description of the estimators and computational strategy employed to implement these ideas can be found in [4].

3.1 Background

A brief description of Monte Carlo solutions to the transport equation is given and provides the background needed to understand our solution method. The integro-differential form of the radiative transport equation (RTE) is

$$\begin{aligned}\nabla \cdot \varOmega\, \varPsi(\mathbf{r}, \varOmega) &+ \mu_t(\mathbf{r})\varPsi(\mathbf{r}, \varOmega)\\ &= \mu_s(\mathbf{r}) \int_{4\pi} f(\varOmega' \to \varOmega)\varPsi(\mathbf{r}, \varOmega')\, d\varOmega' + Q(\mathbf{r}, \varOmega) \qquad (1)\end{aligned}$$

where $\varPsi(\mathbf{r}, \varOmega)$ is the flux (or radiance) as a function of position $\mathbf{r}$ and direction $\varOmega$, $\mu_t = \mu_s + \mu_a$ is the total attenuation, μ_a is the absorption coefficient, μ_s is the scattering coefficient, $f(\varOmega' \to \varOmega)$ is the scattering phase function, and Q

represents internal (volumetric) source(s). The integral on the right hand side of Eq. 1 is taken over 4π steradians of the unit sphere. Our interest is in estimating reflected (or transmitted) light at the detector location, which is determined by

$$R = \int_{\Gamma} Q^*(\mathbf{r}, \Omega)\Psi(\mathbf{r}, \Omega)\, d\mathbf{r}\, d\Omega$$

where Q^* is the detector (adjoint source) function, typically the characteristic function for physical detection, and Γ is the phase space.

An equivalent formulation of the integro-differential form of the RTE can be derived by integrating along characteristics [1] to obtain the integral form of the RTE. The integral form of the RTE for the collision density $\Phi = \mu_t \Psi$ is

$$\Phi(\mathbf{r}, \Omega) = \mathscr{K}\Phi(\mathbf{r}, \Omega) + S(\mathbf{r}, \Omega) \tag{2}$$

where $\mathscr{K}[\cdot] = \int K[\cdot]$ is the transport kernel, and S is the density of first collisions. The integral form of the RTE provides a more direct link between the physical model and the probabilistic model on which the Monte Carlo formulations are based. It is also useful because existence and uniqueness of the RTE solution can be established using the fact that the integral operator $\mathscr{K}$ is a contractive map [10]. The classical solution to Eq. 2 is given by the Neumann series

$$\Phi = (I + \mathscr{K} + \mathscr{K}^2 + \mathscr{K}^3 \cdots)S$$

provided the $\mathscr{L}_1$ norm of $\mathscr{K}$ is less than 1, $||\mathscr{K}|| < 1$, (or the weaker condition $||\mathscr{K}^n|| < 1$ for some $n \geq 1$ [10]) to ensure convergence of this series and existence of a unique solution. Making use of reciprocity [10], we can also represent reflected light at the detector location by

$$R = \int_{\Gamma} S^*(\mathbf{r}, \Omega)\Phi(\mathbf{r}, \Omega)\, d\mathbf{r}\, d\Omega$$

where S^* is the detector (adjoint source) function.

3.2 Part I: Averaged Sequential Correlated Sampling (ASCS)

Sequential correlated sampling (SCS) is a technique introduced by Halton in 1962 [3] to solve matrix problems. Generally, the technique subtracts an approximate solution from the governing equation at each stage and uses the random walks generated by Monte Carlo in the next stage to identify an additive correction to the solution obtained from the earlier stages. The SCS idea can be illustrated by subtracting an approximate solution $\tilde{\Phi}$ from the integral RTE:

$$\begin{aligned}\Phi - \tilde{\Phi} &= \mathscr{K}\Phi + S - \tilde{\Phi}\\ &= \mathscr{K}\Phi - \mathscr{K}\tilde{\Phi} + \mathscr{K}\tilde{\Phi} + S - \tilde{\Phi}\\ &= \mathscr{K}[\Phi - \tilde{\Phi}] + (\mathscr{K}\tilde{\Phi} + S - \tilde{\Phi}).\end{aligned}$$

The term in parenthesis, $(\mathscr{K}\tilde{\Phi} + S - \tilde{\Phi})$, forms a "reduced source" for an RTE whose solution is $[\Phi - \tilde{\Phi}]$. Random walks are initiated at each stage n using the reduced source determined from the previous stage $n - 1$

$$\begin{aligned}S^n(\mathbf{r}, \Omega) &= \mathscr{K}\tilde{\Phi}^{n-1}(\mathbf{r}, \Omega) + S^{n-1}(\mathbf{r}, \Omega) - \tilde{\Phi}^{n-1}(\mathbf{r}, \Omega)\\ S^0(\mathbf{r}, \Omega) &= S(\mathbf{r}, \Omega).\end{aligned}$$

This algorithm is self-correcting; that is, each stage produces an additive correction to the previous representation of Φ as a sum over previous stages. The additive term tends to 0 as the stages increase and the variance in the correction also tends to 0. The solution $\Phi = \Phi^0 + \Phi^1 + \cdots$ is then used to estimate detected light $R = \int S^*\Phi$.

The question that underlies the **averaged** sequential correlated sampling (ASCS) approach is: Can we estimate R with sufficient accuracy using only averaged values of Φ throughout phase space Γ and save a lot of time when compared with conventional Monte Carlo? Beginning with a crude initial decomposition of Γ we would like to understand how to improve this crude mesh so that the ASCS algorithm assigns computational resources in regions and directions that matter most in estimating detection. Note that we cannot avoid the need for an approximate RTE solution **everywhere** (because the RTE couples every two phase space states nontrivially through the transport kernel). However, we want our algorithm to take advantage of the possibility that the mesh over which the approximate solution of regionwise averages is defined can be adjusted recursively to account for varying relative accuracy needed throughout the phase space.

To implement the algorithm, we decompose Γ into space-angle bins $\Gamma_{ij} = \delta_i \times \Delta_{ij}$. Spatial mesh elements, δ_i, are defined and then for each of these spatial bins, angular mesh elements, Δ_{ij}, are defined so that

$$\Gamma = \bigcup_i \bigcup_j \Gamma_{ij}. \tag{3}$$

Then average values of Φ are determined for each bin by Monte Carlo

$$\bar{\Phi}_{ij} = \frac{1}{|\delta_i||\Delta_{ij}|} \int_{\Delta_{ij}} \int_{\delta_i} \Phi(\mathbf{r}, \Omega)\, d\mathbf{r}\, d\Omega.$$

To initiate the algorithm in stage 1, the first stage averages $\bar{\Phi}^1_{ij}$ are determined using conventional Monte Carlo based on the physical source $S(\mathbf{r}, \Omega)$ of initial collisions. A new "reduced source" is then found from the formula

$$\bar{S}_{ij}^{2} = \mathscr{K}\bar{\Phi}_{ij}^{1} + \bar{S}_{ij}^{1} - \bar{\Phi}_{ij}^{1}$$

where

$$\bar{S}_{ij}^{1} = \frac{1}{|\delta_i||\Delta_{ij}|}\int_{\Delta_{ij}}\int_{\delta_i} S(\mathbf{r}, \Omega)\, d\mathbf{r}\, d\Omega.$$

The ASCS algorithm then makes use of the adaptive process defined by

$$\bar{S}_{ij}^{n+1} = \mathscr{K}\bar{\Phi}_{ij}^{n} + \bar{S}_{ij}^{n} - \bar{\Phi}_{ij}^{n}$$

to identify a reduced source for stage $n + 1$ from the reduced source for stage n together with the histogram increments $\bar{\Phi}_{ij}^{n}$ that are generated by the source $\bar{S}_{ij}^{n}$. The sum over all adaptive stages, $\bar{\Phi}_{ij} = \bar{\Phi}^{1} + \bar{\Phi}^{2} + \cdots$ has been shown to converge geometrically to a histogram solution of Φ [8] that can then be used to compute the detected light signal $R = \int S^{*}\Phi$. However, the accuracy of the solution obtained in this way is limited by the selected mesh. This leads us to the idea of an intelligent mesh refinement determined using the contributon map.

3.3 Part II: Contributon Map and Mesh Refinement

To form the contributon map we need an adjoint RTE solution. The RTE that is adjoint to Eq. 1 in integro-differential form is:

$$\begin{aligned} -\nabla \cdot \Omega\Psi^{*}(\mathbf{r}, \Omega) + \mu_t(\mathbf{r})\Psi^{*}(\mathbf{r}, \Omega) = \\ \mu_s(\mathbf{r})\int_{4\pi} f(\Omega \to \Omega')\Psi^{*}(\mathbf{r}, \Omega')\, d\Omega' + Q^{*}(\mathbf{r}, \Omega) \end{aligned} \tag{4}$$

where Ψ^{*} is the adjoint flux (or radiance).

The contributon equation is formed by multiplying Eq. 1 by Ψ^{*} and Eq. 4 by Ψ and subtracting, which produces a new transport equation:

$$\begin{aligned} \nabla \cdot \Omega\Psi\Psi^{*} = \Psi^{*}(\mathbf{r}, \Omega)\mu_s(\mathbf{r})\int_{4\pi} f(\Omega' \to \Omega)\Psi(\mathbf{r}, \Omega')\, d\Omega' \\ - \Psi(\mathbf{r}, \Omega)\mu_s(\mathbf{r})\int_{4\pi} f(\Omega \to \Omega')\Psi^{*}(\mathbf{r}, \Omega')\, d\Omega' + Q\Psi^{*} - Q^{*}\Psi. \end{aligned} \tag{5}$$

When the new dependent variable, called the contributon function, is introduced

$$C(\mathbf{r}, \Omega) = \Psi(\mathbf{r}, \Omega)\Psi^{*}(\mathbf{r}, \Omega),$$

Eq. 5 becomes

$$\nabla \cdot \Omega C(\mathbf{r},\Omega) + \Sigma_s(\mathbf{r},\Omega)C(\mathbf{r},\Omega) = \int_{4\pi} \Sigma(\mathbf{r},\Omega' \to \Omega)C(\mathbf{r},\Omega')\,d\Omega' + Q\Psi^* - Q^*\Psi$$

where

$$\Sigma_s(\mathbf{r},\Omega) = \int_{4\pi} \Sigma(\mathbf{r},\Omega \to \Omega')\,d\Omega'$$

and

$$\Sigma(\mathbf{r},\Omega \to \Omega') = \mu_s(\mathbf{r}) f(\mathbf{r},\Omega \to \Omega') \frac{\Psi^*(\mathbf{r},\Omega')}{\Psi^*(\mathbf{r},\Omega)}.$$

Provided that the adjoint solution, $\Psi^*(\mathbf{r},\Omega)$, satisfies boundary conditions that are dual to those satisfied by $\Psi(\mathbf{r},\Omega)$ (which is a natural constraint for most RTE problems), this contributon transport equation describes an information density function, C, that captures the flow of "contributons" from the source through the tissue to the detector [11]. Notice that there is no absorption in the equation, only scattering. Thus, there is a **lossless** flow of "particles" (i.e., contributons) in this system that necessarily describe transport from source to detector.

Instead of solving the contributon equation directly, we estimate

$$\int\int C(\mathbf{r},\Omega)\,d\mathbf{r}\,d\Omega = \int\int \Psi(\mathbf{r},\Omega)\Psi^*(\mathbf{r},\Omega)\,d\mathbf{r}\,d\Omega$$

by matching Ψ and Ψ^* solutions, both spatially and angularly, over decompositions of the phase space Γ. Because the function C is the product of $\Psi(\mathbf{r},\Omega)$ (the photon intensity per unit source) and $\Psi^*(\mathbf{r},\Omega)$ (which can be regarded as the probability that a unit weight particle initiated at $(\mathbf{r},\Omega)$ will be detected), the value of $C(\mathbf{r},\Omega)$ provides a relative measure of the importance of the phase space vector $(\mathbf{r},\Omega)$ in transmitting light from source to detector in the given transport problem. Therefore we expect that a map of $C(\mathbf{r},\Omega)$ throughout the phase space Γ encodes critical information about the specific locations and directions that are most important in computing accurate estimates of detected light. Our algorithm uses this information in the way we now describe.

The theory of geometric convergence for ASCS [8] suggests that performing ASCS iterations on any fixed mesh converges geometrically to an estimate of detected light whose accuracy is limited by the maximal difference between the exact and approximate RTE solutions: $\sup_{(\mathbf{r},\Omega)} \left|\Psi(\mathbf{r},\Omega) - \bar{\Psi}(\mathbf{r},\Omega)\right|$, where $\bar{\Psi}(\mathbf{r},\Omega)$ is the regionwise constant approximation determined from our algorithm. From this we deduce that refining the mesh will produce more accurate ASCS estimates. Therefore, we have chosen to use the contributon map as the basis for deciding which subregions should be further subdivided, and by how much. In our implementation we subdivide subregions with large $\int C$ values until all subregions have roughly equal such values. That is, our algorithm goal is to uniformize $\int C$ over the remeshed phase space.

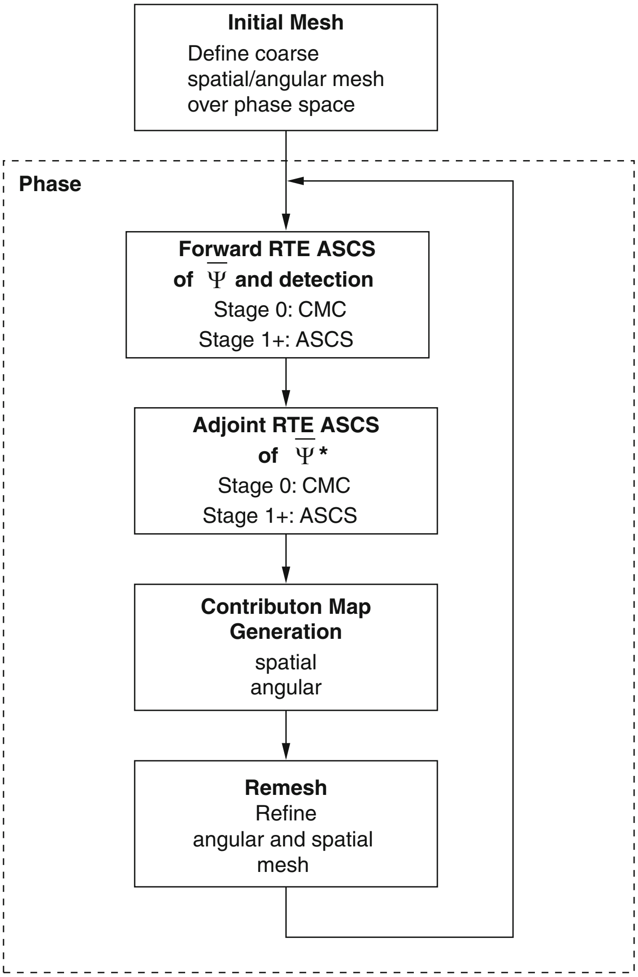

Initial Mesh
Define coarse spatial/angular mesh over phase space
Phase
Forward RTE ASCS of $\overline{\Psi}$ and detection
Stage 0: CMC
Stage 1+: ASCS
Adjoint RTE ASCS of $\overline{\Psi}$*
Stage 0: CMC
Stage 1+: ASCS
Contributon Map Generation
spatial
angular
Remesh
Refine angular and spatial mesh

Fig. 3 Flowchart of our algorithm.

The steps of our algorithm follow the flow chart shown in Fig. 3. We first define an initial coarse spatial/angular mesh over the phase space Γ as defined in Eq. 3. Using this mesh we execute ASCS and obtain regionwise constant values of the forward RTE solution, $\bar{\Psi}_{ij}$, over each space-angle bin. These are incorporated in an estimator that estimates detection at the optical probe detector site. Our initial stage of ASCS uses conventional Monte Carlo to obtain the initial estimates of $\bar{\Psi}_{ij}$. We next execute n adaptive stages of the ASCS algorithm without refining the mesh. Using the same mesh, an approximate adjoint solution $\bar{\Psi}^*_{ij}$ is obtained for each mesh element using n stages. The forward and adjoint solutions are then combined to form spatial contributon maps with elements

$$\bar{C}_i = \int_{4\pi} \int_{\delta_i} C(\mathbf{r}, \Omega) \, d\mathbf{r} \, d\Omega \tag{6}$$

and angular contributon maps with elements

$$\bar{C}_{ij} = \int_{\Delta_{ij}} \int_{\delta_i} C(\mathbf{r}, \Omega) \, d\mathbf{r} \, d\Omega. \tag{7}$$

Note that the spatial contributon for spatial mesh element δ_i, Eq. 6, is the sum of the $\bar{C}_{ij}$ [Eq. 7] over j. The spatial contributon map is considered first and the relative magnitudes of its values over the entire phase space are compared. This is accomplished by linearly ordering the $\{\bar{C}_i\}_{i=1}^{I} : \bar{C}_{i_1} \leq \bar{C}_{i_2} \leq \cdots \leq \bar{C}_{i_I}$ and then using the ratios $\bar{C}_{i_k}/\bar{C}_{i_1}$ to decompose the mesh element δ_{i_k} into $\lfloor \bar{C}_{i_k}/\bar{C}_{i_1} \rfloor$ equal subelements, where the notation $\lfloor X \rfloor$ means "greatest integer". Depending on the RTE problem and the geometry, there may be a variety of ways of performing this subdivision, but here we are only concerned with describing the general ideas that underlie our current mesh refinement strategy.

Once the spatial subdivision has been determined, the angular contributon maps are analyzed. Within a particular spatial bin δ_i, the angular contributon map over the unit sphere of directions "attached to" δ_i is considered. Again with the intent to uniformize the information density across all spatial/angular bins, in a fashion similar to that used for the spatial subdivision, the smallest angular magnitude within the unit sphere of directions is used to subdivide the angular bins. With a new spatial/angular mesh thus defined, the ASCS algorithm steps are repeated.

A "phase" consists of the steps contained within the dashed line in Fig. 3. Using the initial coarse mesh, Phase I is executed. After the new mesh is defined, Phase II is executed, with additional remeshing and phases employed if needed. These iterations cease when the resulting phase space refinement exhibits minimal contributon variation across the mesh elements.

Previous results have illustrated the benefit in using **spatial** contributon maps in 1-dimensional model problems [4]. Our current goal is to illustrate how the **angular** contributon map can further improve results (even in 1D!) and to move towards implementation in higher dimensional problems.

4 Application: Model Problem Analysis

We have developed a code that uses the algorithm strategy described in this paper to solve five dimensional (three spatial coordinates, two angular coordinates) tissue transport problems. For this initial test we applied the code to the simple slab represented in Fig. 4. For this test we estimated transmitted photons rather than reflected ones as our measurement of detection. A source injects photons into the top of a tissue slab normal to the surface. A detector is placed on the bottom of the slab and captures all photons transmitted through the slab. The slab is

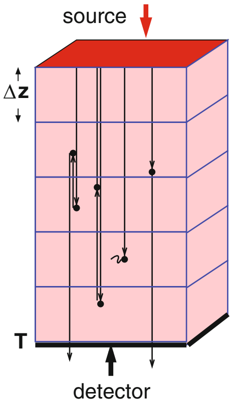


Fig. 4 Schematic of the 3-dimensional model transport problem.

modeled with optical properties characteristic of tissue, with absorption coefficient $\mu_a = 0.01$/mm, scattering coefficient $\mu_s = 0.99$/mm and anisotropy coefficient (average cosine of the scattering angle) $g = 0.9$. The slab thickness is 10 optical mean free paths, $T = 10$ mm. For this model problem, only bidirectional scattering is allowed so that we can compare our results to an analytic 1D solution which is available. Our initial coarse mesh divides the slab into 100 uniform spatial bins along the $z-$axis and 2 angular hemispheres defined by $-1 < \cos\theta < 0$ and $0 < \cos\theta < 1$ with $0 < \phi < 2\pi$, where θ = polar angle and ϕ = azimuthal angle. Thirty random walks were launched uniformly distributed in each mesh element to provide estimates of $\bar{\Psi}$, and an additional 10^5 random walks were used to estimate detection, $\int_\Gamma S^*\Psi$.

As described in Sect. 3.3 we evaluate integrals of the contributon function $C(\mathbf{r}, \Omega)$ using our approximations to the forward and adjoint solutions ($\bar{\Psi}$ and $\bar{\Psi}^*$) over the mesh elements. In this bidirectional model problem, photon movement within the the angular bins degenerates to movement solely up and down along the $z-$axis; i.e., along Ω and $-\Omega$ directions. The evaluation of the contributon integral over each spatial bin for this 1D model problem becomes

$$\int_{4\pi}\int_{\delta_i} C(\mathbf{r}, \Omega) d\mathbf{r} d\Omega = \int_{4\pi}\int_{\delta_i} \Psi(\mathbf{r}, \Omega)\Psi^*(\mathbf{r}, \Omega)\, d\mathbf{r}i\, d\Omega$$
$$\approx [\bar{\Psi}_{i+}\bar{\Psi}^*_{i+} + \bar{\Psi}_{i-}\bar{\Psi}^*_{i-}]|\delta_i| = \bar{C}_i|\delta_i|$$

where the + subscript designates downward-pointing directions and the − subscript designates the upward pointing ones. Figure 5 plots the spatial contributon values for each spatial bin (shown by the histogram) against the analytic solution (shown by the thick solid line). The spatial contributon values range from 0.058 to 0.066, on the basis of which no spatial subdivision is performed.

We next examine the angular contributon integrals. Because we have chosen to model scattering directly up or down, there are two angular maps describing

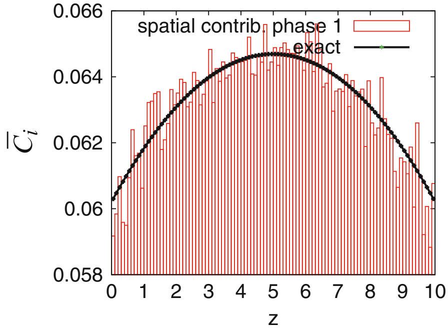


Fig. 5 Spatial contributon value ($\bar{C}_i$) as a function of the spatial bin.

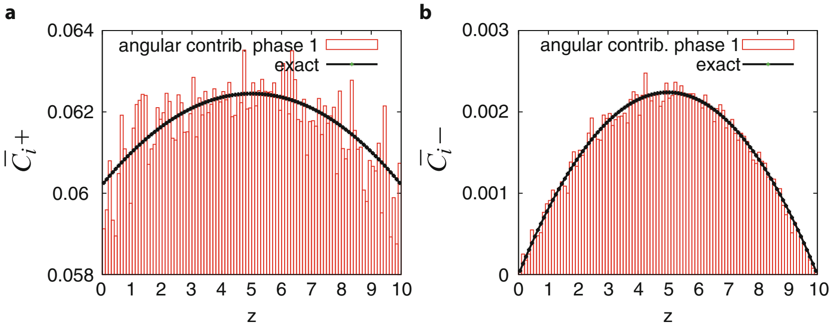


Fig. 6 Angular contributon value as a function of spatial bin for the (**a**) downward ($\bar{C}_{i+}$), and (**b**) upward ($\bar{C}_{i-}$) moving flux.

flow downward (exiting the lower hemisphere) and flow upward (exiting the upper hemisphere). Again, these are approximated using our coupled forward/adjoint estimates of $\bar{\Psi}$ and $\bar{\Psi}^*$. If we define $\bar{C}_{i+} = \bar{\Psi}_{i+}\bar{\Psi}^*_{i+}$ and $\bar{C}_{i-} = \bar{\Psi}_{i-}\bar{\Psi}^*_{i-}$, then we can write

$$\int_{4\pi}\int_{\delta_i} C(\mathbf{r},\Omega)\, d\mathbf{r}\, d\Omega = \int_{+}\int_{\delta_i} C(\mathbf{r},\Omega)\, d\mathbf{r}\, d\Omega + \int_{-}\int_{\delta_i} C(\mathbf{r},\Omega)\, d\mathbf{r}\, d\Omega$$
$$\approx [\bar{C}_{i+} + \bar{C}_{i-}]|\delta_i|$$

where the lower limit + on the integral designates the lower hemisphere of directions while − designates the upper hemisphere. Figure 6 plots the angular contributon values for each spatial bin against the analytic solution. Figure 6a, b display the downward and upward angular components, respectively. Although the variation of the magnitude of the angular contributon values across spatial bins is not large, comparison of the magnitudes between the downward and upward hemispheres in each spatial bin is large and indicates that downward (i.e., towards the detector) directions provide more “contributon information” than upward directions.

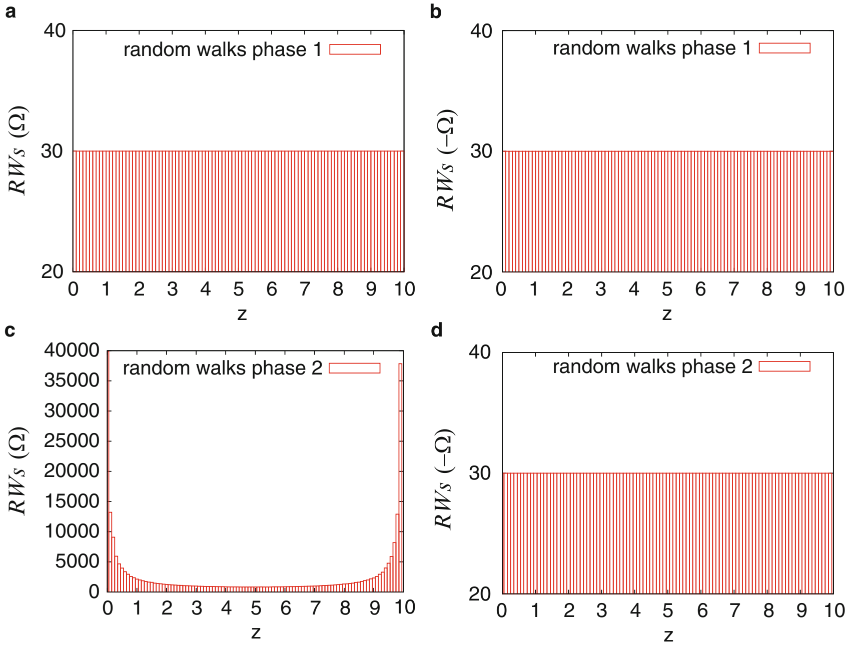


Fig. 7 Number of random walks executed in each spatial bin for Phase I in (**a**) downward (Ω), and (**b**) upward ($-\Omega$) directions, and for Phase II in (**c**) downward (Ω), and (**d**) upward ($-\Omega$) directions.

For this bidirectional model problem with only two discrete scattering directions, no angular subdivision is possible. Instead, we modify the number of random walks in each direction based on the ratios $[\bar{C}_{i+}/\bar{C}_{i-}]$. Figure 7a, b plot the number of random walks executed in each mesh bin using the initial coarse mesh (Phase I). For the initial mesh we generated 30 random walks per mesh element in both the downward (Fig. 7a) and upward (Fig. 7b) directions. Figure 7c, d plot the number of random walks to be executed for Phase II as designated by the angular contributon maps. Figure 7c shows that for the downward hemisphere (towards the detector), an increase in the number of random walks is needed throughout the slab, in particular close to the source and detector. For the upward hemisphere (Fig. 7d) no redistribution in the number of random walks is needed.

Table 1 shows our comparative efficiencies for this model problem. The exact analytic value for detection is shown on the first line of the table. Our estimate at various stages, standard deviation, time to execute in seconds and relative efficiency are displayed. The relative efficiency is the efficiency relative to that of conventional Monte Carlo (CMC). As stated in the introduction of this paper, our goal in devising this solution method is to produce estimates of detection with efficiency improved beyond central limit theorem rates. So the relative efficiency factor indicates how much our method improves upon CMC (i.e., by just running more photons).

Table 1 Detection estimates for the 3-dimensional model problem showing the exact value and estimates based on conventional Monte Carlo (CMC), and our solution method for an initial coarse mesh (Mesh Phase I) and from a refined mesh designated by the spatial/angular contributon maps (Mesh Phase II). The standard deviation (SD) of our estimates, time in seconds, and efficiency relative to CMC (Rel. Eff.) are shown in the final three columns.

Method	Stages	Estimate	SD	Time [sec]	Rel. Eff.
Exact	–	0.602096	–	–	∞
CMC	1	0.601018	0.002254	6.69	1
Mesh Phase I	4	0.602136	0.000145	26.70	60
Mesh Phase II	7	0.602103	0.000011	106.93	2,808

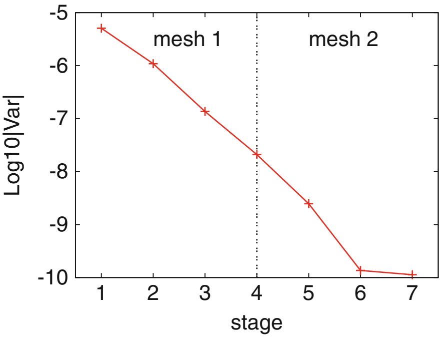


Fig. 8 Variance reduction as a function of stage.

Using the initial coarse mesh, we ran four stages of the forward ASCS solution to obtain convergence. With this coarse mesh alone we were able to obtain a relative efficiency gain by a factor of 60. Then four stages of the adjoint ASCS solution were executed and the spatial/angular contributon maps shown in Figs. 5 and 6 were generated. The angular contributon analysis produced a reallocation of random walks as shown in Fig. 7c, d and this produced a relative efficiency gain of 2,808 for the combined Phase I/Phase II strategy. Figure 8 plots the reduction of the base 10 log of the variance in the estimates of detection as a function of the number of adaptive stages using the two meshes. Because each phase employs an adaptive algorithm using a fixed mesh, the geometric convergence will cease when accuracy consistent with the mesh is reached. The break observed at stage 6 exhibits this behavior.

This model problem analysis utilized 100 spatial bins and 2 angular bins for the forward and adjoint simulations resulting in 400 double-precision numbers to be stored. If spatial subdivisions along the x- and y-axes and angular subdivisions azimuthally were added, the storage requirements could become unmanageable. To avoid an unruly memory utilization we have, in other problems, only refined the mesh in the region of interest (e.g., near the source/detector).

5 Summary

We have described our most recent work with adaptive, geometrically convergent Monte Carlo algorithms based on sequential correlated sampling error reduction ideas. Our algorithm constructs histogram estimates of general RTE solutions, including heterogeneity, as well as estimates of reflected/transmitted light. It employs contributon maps to refine an initial phase space decomposition to extend the geometric learning. We have presented results that reveal the importance of the angular dependence of the contributon map and how, in addition to the spatial information it contains, it can determine an appropriate remeshing of the phase space for improved computational efficiency.

The angular information contained in the contributon map is important not only for physical systems with high anisotropy (like tissue) but for isotropic problems (nuclear, global illumination) as well. In problems not described here, we generated results for model problems similar to the one discussed in Sect. 4 with isotropic scattering ($g = 0$) and found that the angular dependence is needed there too for optimal remeshing. This is because, although the particle scattering is isotropic, there are phase space regions in which the flux is not isotropic (near sources and boundaries), and also because the placement of the detector relative to the source will always have an effect on the directional importance of the flow of the radiation, whether the radiation is photons of light, neutrons or electrons.

An extension of our current ASCS solution method based on simple histogram estimates of averaged RTE solutions would be to replace the regionwise constant approximation by other locally defined approximations; e.g., ones that are regionwise polynomial. We also plan to examine the effect of alternate spatial and angular subdivisions, such as those useful in finite element method applications. While the phase space descriptions may change, we anticipate that contributon map information will provide the appropriate tool for deciding how crude phase space decompositions are to be refined to optimize computational efficiency in Monte Carlo simulations.

Acknowledgements We acknowledge support from the Laser Microbeam and Medical Program (LAMMP) a NIH Biomedical Technology Resource Center (P41-RR01192) and from the National Institutes of Health (NIH, K25-EB007309).

References

1. G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*. Krieger Pub. Co., 1970.
2. P. Billingsley. *Probability and Measure*. John Wiley and Sons, 1979.
3. J. Halton. Sequential Monte Carlo. *Proc. Camb. Phil. Soc.*, 58:57–78, 1962.
4. R. Kong, M. Ambrose, and J. Spanier. Efficient, automated Monte Carlo methods for radiation transport. *J. Comp. Physics*, 227(22):9463–9476, 2008.
5. R. Kong and J. Spanier. Error analysis of sequential Monte Carlo methods for transport problems. In *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 252–272. Springer, 1999.

6. R. Kong and J. Spanier. Sequential correlated sampling algorithms for some transport problems. In *Monte Carlo and Quasi-Monte Carlo Methods 1998*, pages 238–251. Springer, 1999.
7. R. Kong and J. Spanier. A new proof of geometric convergence for general transport problems based on sequential correlated sampling methods. *J. Comp. Physics*, 227(23):9762–9777, 2008.
8. R. Kong and J. Spanier. Geometric convergence of second generation adaptive Monte Carlo algorithms for general transport problems based on correlated sampling. *Intl. J. Pure & Appl. Math.*, 59(4):435–455, 2010.
9. MCNP, a Monte Carlo N-Particle Transport Code version 5, Report LA-UR-03-1987. Technical report, Los Alamos National Laboratory, 2003.
10. J. Spanier and E. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969, reprinted by Dover Publications, Inc. 2008.
11. M. L. Williams and W. W. Engle. The concept of spatial channel theory applied to reactor shielding analysis. *Nucl. Sci. Eng.*, 62:92–104, 1977.

Hybrid Function Systems in the Theory of Uniform Distribution of Sequences

Peter Hellekalek

Abstract A hybrid sequence in the multidimensional unit cube is a combination of two or more lower-dimensional sequences of different types. In this paper, we present tools to analyze the uniform distribution of such sequences. In particular, we introduce *hybrid* function systems, which are classes of functions that are composed of the trigonometric functions, the Walsh functions in base **b**, and the **p**-adic functions. The latter are related to the dual group of the p-adic integers, p a prime. We prove the Weyl criterion for hybrid function systems and define a new notion of diaphony, the *hybrid diaphony*. Our approach generalizes several known concepts and results.

1 Introduction

This work is motivated by recent advances of Niederreiter [15–17] in the analysis of certain *hybrid sequences*. In this series of papers, the first deterministic discrepancy bounds for such high-dimensional point sets were established.

Hybrid sequences are sequences of points in the multidimensional unit cube $[0,1[^s$ where certain coordinates of the points stem from one lower-dimensional sequence and the remaining coordinates from a second lower-dimensional sequence. Of course, this idea of "mixing" two sequences to obtain a new sequence in higher dimensions may be extended to more than two components.

While of considerable theoretical interest, there is also an applied aspect to this construction principle. As first proposed by Spanier [19], with a hybrid sequence composed of a low-discrepancy sequence and a pseudorandom number sequence,

P. Hellekalek
Department of Mathematics, University of Salzburg, Hellbrunner Strasse 34, 5020, Salzburg, Austria
e-mail: peter.hellekalek@sbg.ac.at

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_24, © Springer-Verlag Berlin Heidelberg 2012

one may combine the advantages of quasi-Monte Carlo methods and Monte Carlo methods for multidimensional numerical integration.

In this paper, we present new tools for the analysis of hybrid sequences. For this task, we introduce *hybrid function systems* on $[0,1[^s$, by which we denote orthonormal bases of $L^2([0,1[^s)$ that are obtained by mixing lower-dimensional bases, in analogy to the construction principle of hybrid sequences. Our components will be the trigonometric, the Walsh, and the **p**-adic function system (for this choice, see Remark 7).

As a qualitative result, we prove a hybrid version of the Weyl criterion. Further, we introduce the *hybrid diaphony* as a figure of merit that allows to measure the uniform distribution of hybrid sequences, and show its basic properties. This concept generalizes several of the current notions of diaphony (see Remark 6). In addition, we prove an inequality of the Erdös-Turán-Koksma type, i.e., an upper bound for the hybrid diaphony in terms of certain exponential sums.

2 Preliminaries

Throughout this paper, b denotes a positive integer, $b \geq 2$, and $\mathbf{b} = (b_1, \ldots, b_s)$ stands for a vector of not necessarily distinct integers $b_i \geq 2$, $1 \leq i \leq s$. Further, p denotes a prime, and $\mathbf{p} = (p_1, \ldots, p_s)$ represents a vector of not necessarily distinct primes p_i, $1 \leq i \leq s$. $\mathbb{N}$ stands for the positive integers, and we put $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

The underlying space is the s-dimensional torus $\mathbb{R}^s/\mathbb{Z}^s$, which will be identified with the half-open interval $[0,1[^s$. Haar measure on the s-torus $[0,1[^s$ will be denoted by λ_s. We put $\mathrm{e}(y) = \mathrm{e}^{2\pi \mathrm{i} y}$ for $y \in \mathbb{R}$, where i is the imaginary unit.

We will use the standard convention that empty sums have value 0 and empty products value 1.

Definition 1. Let $k \in \mathbb{Z}$. The *kth trigonometric function* e_k is defined as $\mathrm{e}_k : [0,1[\to \mathbb{C}$, $\mathrm{e}_k(x) = \mathrm{e}(kx)$. For $\mathbf{k} = (k_1, \ldots, k_s) \in \mathbb{Z}^s$, the **k***th trigonometric function* $\mathrm{e}_{\mathbf{k}}$ is defined as $\mathrm{e}_{\mathbf{k}} : [0,1[^s \to \mathbb{C}$, $\mathrm{e}_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{s} \mathrm{e}(k_i x_i)$, $\mathbf{x} = (x_1, \ldots, x_s) \in [0,1[^s$. The trigonometric function system in dimension s is denoted by $\mathscr{T}^{(s)} = \{\mathrm{e}_{\mathbf{k}} : \mathbf{k} \in \mathbb{Z}^s\}$.

For a nonnegative integer k, let $k = \sum_{j\geq 0} k_j\, b^j$, $k_j \in \{0,1,\ldots,b-1\}$, be the unique b-adic representation of k in base b. With the exception of at most finitely many indices j, the digits k_j are equal to 0.

Every real number $x \in [0,1[$ has a b-adic representation $x = \sum_{j\geq 0} x_j\, b^{-j-1}$, $x_j \in \{0,1,\ldots,b-1\}$. If x is a *b-adic rational*, which means that $x = ab^{-g}$, a and g integers, $0 \leq a < b^g$, $g \in \mathbb{N}$, and if $x \neq 0$, then there exist two such representations.

The b-adic representation of x is uniquely determined under the condition that $x_j \neq b-1$ for infinitely many j. In the following, we will call this particular representation the *regular* (b-adic) representation of x.

Definition 2. For $k \in \mathbb{N}_0$, $k = \sum_{j \geq 0} k_j b^j$, and $x \in [0,1[$, with regular b-adic representation $x = \sum_{j \geq 0} x_j b^{-j-1}$, the *kth Walsh function in base b* is defined by $w_k(x) = \mathrm{e}((\sum_{j \geq 0} k_j x_j)/b)$. For $\mathbf{k} \in \mathbb{N}_0^s$, $\mathbf{k} = (k_1, \ldots, k_s)$, and $\mathbf{x} \in [0,1[^s$, $\mathbf{x} = (x_1, \ldots, x_s)$, we define the **k***th Walsh function* $w_{\mathbf{k}}$ *in base* $\mathbf{b} = (b_1, \ldots, b_s)$ on $[0,1[^s$ as the following product: $w_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^{s} w_{k_i}(x_i)$, where w_{k_i} denotes the k_ith Walsh function in base b_i, $1 \leq i \leq s$. The Walsh function system in base $\mathbf{b}$, in dimension s, is denoted by $\mathscr{W}_{\mathbf{b}}^{(s)} = \{w_{\mathbf{k}} : \mathbf{k} \in \mathbb{N}_0^s\}$.

We refer the reader to [1, 4, 5] for elementary properties of the Walsh functions and to [18] for the background in harmonic analysis.

Let $\mathbb{Z}_b$ denote the compact group of the b-adic integers (see [9] and [13] for details). An element z of $\mathbb{Z}_b$ will be written as $z = \sum_{j \geq 0} z_j\, b^j$, with digits $z_j \in \{0, 1, \ldots, b-1\}$. Two such elements are added by addition with carry.

The set of integers $\mathbb{Z}$ is embedded in $\mathbb{Z}_b$. If $z \in \mathbb{N}_0$, then at most finitely many digits z_j are different from 0. If $z \in \mathbb{Z}$, $z < 0$, then at most finitely many digits z_j are different from $b-1$. In particular, $-1 = \sum_{j \geq 0} (b-1)\, b^j$.

We recall the following concepts from Hellekalek [7].

Definition 3. The map $\varphi_b : \mathbb{Z}_b \to [0,1[$, given by $\varphi_b(\sum_{j \geq 0} z_j\, b^j) = \sum_{j \geq 0} z_j\, b^{-j-1}$ (mod 1), will be called the *b-adic Monna map*.

The restriction of φ_b to $\mathbb{N}_0$ is often called the *radical-inverse function* in base b. The Monna map is surjective, but not injective. It may be inverted in the following sense.

Definition 4. We define the *pseudoinverse* φ_b^+ of the b-adic Monna map φ_b by

$$\varphi_b^+ : [0,1[\to \mathbb{Z}_b, \quad \varphi_b^+(\sum_{j \geq 0} x_j\, b^{-j-1}) = \sum_{j \geq 0} x_j\, b^j,$$

where $\sum_{j \geq 0} x_j\, b^{-j-1}$ stands for the regular b-adic representation of the element $x \in [0,1[$.

The image of $[0,1[$ under φ_b^+ is the set $\mathbb{Z}_b \setminus (-\mathbb{N})$. Furthermore, $\varphi_b \circ \varphi_b^+$ is the identity map on $[0,1[$, and $\varphi_b^+ \circ \varphi_b$ the identity on $\mathbb{N}_0 \subset \mathbb{Z}_b$. In general, $z \neq \varphi_b^+(\varphi_b(z))$, for $z \in \mathbb{Z}_b$. For example, if $z = -1$, then $\varphi_b^+(\varphi_b(-1)) = \varphi_b^+(0) = 0 \neq -1$.

If $b = p$ is a prime, φ_p gives a bijection between the subset $\mathbb{N}$ of $\mathbb{Z}_p$ of positive integers and the set $\{ap^{-g} : 0 < a < p^g, g \in \mathbb{N}, (a, p^g) = (a, p) = 1\}$ of all reduced p-adic fractions. It was shown in [6] that, as a consequence, the *dual group* $\hat{\mathbb{Z}}_p$ of $\mathbb{Z}_p$ (for this notion, see [9]) can be written in the form $\hat{\mathbb{Z}}_p = \{\chi_k : k \in \mathbb{N}_0\}$, where $\chi_k : \mathbb{Z}_p \to \{c \in \mathbb{C} : |c| = 1\}$, $\chi_k(\sum_{j \geq 0} z_j\, p^j) = \mathrm{e}(\varphi_p(k)(z_0 + z_1 p + \cdots))$.

We may "lift" the characters χ_k to the torus, as follows.

Definition 5. For a nonnegative integer k, let $\gamma_k : [0,1[\to \mathbb{C}$, $\gamma_k(x) = \chi_k(\varphi_p^+(x))$, denote the kth p-adic function. We put $\Gamma_p = \{\gamma_k : k \in \mathbb{N}_0\}$.

Remark 1. The functions γ_k are step functions on $[0,1[$. For details and also for the higher-dimensional case, we refer the reader to [7, Lemma 3.5].

There is an obvious generalization of the preceding notions to the higher-dimensional case. Let $\mathbf{b} = (b_1,\ldots,b_s)$ be a vector of not necessarily distinct integers $b_i \geq 2$, let $\mathbf{x} = (x_1,\ldots,x_s) \in [0,1[^s$, let $\mathbf{z} = (z_1,\ldots,z_s)$ denote an element of the compact product group $\mathbb{Z}_{\mathbf{b}} = \mathbb{Z}_{b_1} \times \cdots \times \mathbb{Z}_{b_s}$, and let $\mathbf{k} = (k_1,\ldots,k_s) \in \mathbb{N}_0^s$. We define $\varphi_{\mathbf{b}}(\mathbf{z}) = (\varphi_{b_1}(z_1),\ldots,\varphi_{b_s}(z_s))$, and $\varphi_{\mathbf{b}}^+(\mathbf{x}) = (\varphi_{b_1}^+(x_1),\ldots,\varphi_{b_s}^+(x_s))$.

If $\mathbf{p} = (p_1,\ldots,p_s)$ is a vector of not necessarily distinct primes p_i, then let $\chi_{\mathbf{k}}(\mathbf{z}) = \prod_{i=1}^s \chi_{k_i}(z_i)$, where $\chi_{k_i} \in \hat{\mathbb{Z}}_{p_i}$, and define $\gamma_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^s \gamma_{k_i}(x_i)$, where $\gamma_{k_i} \in \Gamma_{p_i}$, $1 \leq i \leq s$. That is, $\gamma_{\mathbf{k}} = \chi_{\mathbf{k}} \circ \varphi_{\mathbf{p}}^+$. Let $\Gamma_{\mathbf{p}}^{(s)} = \{\gamma_{\mathbf{k}} : \mathbf{k} \in \mathbb{N}_0^s\}$ denote the $\mathbf{p}$*-adic function system* in dimension s. It was shown in [7] that $\Gamma_{\mathbf{p}}^{(s)}$ is an orthonormal basis of $L^2([0,1[^s)$.

3 The Hybrid Weyl Criterion

In what follows, let $s = s_1 + s_2 + s_3$, $s_j \in \mathbb{N}_0$, and $s \geq 1$. We call the numbers s_j the subdimensions and we will consider sequences $\omega = (\mathbf{x}_n)_{n\geq 0}$ in $[0,1[^s$ where the first s_1 coordinates of the point $\mathbf{x}_n$ stem from the nth element $\mathbf{x}_n^{(1)}$ of a sequence $\omega^{(1)}$ on the s_1-torus, the following s_2 coordinates from the nth element $\mathbf{x}_n^{(2)}$ of a sequence $\omega^{(2)}$ on the s_2-torus, and so on. If one of the subdimensions s_j is 0, then only the other component sequences come into play.

Remark 2. Of course, a more general selection principle could have been used to construct the sequence ω from the component sequences $\omega^{(j)}$, by partitioning the set of coordinates $\{1,\ldots,s\}$ into disjoint sets M_j, with card $M_j = s_j$. One would then choose the ith coordinate of $\mathbf{x}_n$ according to which set M_j the index i belongs to, $1 \leq i \leq s$. The results below also hold in this more general setting but we have preferred not to enter this notational nightmare.

For $\mathbf{y} = (y_1,\ldots,y_s) \in \mathbb{R}^s$, let $\mathbf{y}^{(1)} = (y_1,\ldots,y_{s_1})$, $\mathbf{y}^{(2)} = (y_{s_1+1},\ldots,y_{s_1+s_2})$, and $\mathbf{y}^{(3)} = (y_{s_1+s_2+1},\ldots,y_s)$. We will concatenate these vectors and write the vector $\mathbf{y}$ in the form

$$\mathbf{y} = (\mathbf{y}^{(1)},\mathbf{y}^{(2)},\mathbf{y}^{(3)}),$$

by a slight abuse of notation. In the following, we will sometimes have to distinguish between zero vectors in different dimensions. With $\mathbf{0}^{(s)}$, we denote the s-dimensional zero vector, if necessary.

Let us fix the bases $\mathbf{b} = (b_1,\ldots,b_{s_2})$, and $\mathbf{p} = (p_1,\ldots,p_{s_3})$. Suppose that $\mathbf{k} = (\mathbf{k}^{(1)},\mathbf{k}^{(2)},\mathbf{k}^{(3)})$, with components $\mathbf{k}^{(1)} \in \mathbb{Z}^{s_1}$, $\mathbf{k}^{(2)} \in \mathbb{N}_0^{s_2}$, and $\mathbf{k}^{(3)} \in \mathbb{N}_0^{s_3}$. The tensor product $\xi_{\mathbf{k}} = \mathrm{e}_{\mathbf{k}^{(1)}} \otimes w_{\mathbf{k}^{(2)}} \otimes \gamma_{\mathbf{k}^{(3)}}$, where $\mathrm{e}_{\mathbf{k}^{(1)}} \in \mathcal{T}^{(s_1)}$, $w_{\mathbf{k}^{(2)}} \in \mathcal{W}_{\mathbf{b}}^{(s_2)}$, and $\gamma_{\mathbf{k}^{(3)}} \in \Gamma_{\mathbf{p}}^{(s_3)}$, defines a function $\xi_{\mathbf{k}}$ on the s-dimensional unit cube,

$$\xi_{\mathbf{k}} : [0,1[^s \to \mathbb{C}, \quad \xi_{\mathbf{k}}(\mathbf{x}) = \mathrm{e}_{\mathbf{k}^{(1)}}(\mathbf{x}^{(1)}) w_{\mathbf{k}^{(2)}}(\mathbf{x}^{(2)}) \gamma_{\mathbf{k}^{(3)}}(\mathbf{x}^{(3)}),$$

where $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) \in [0,1[^s$.

Definition 6. Let $s = s_1 + s_2 + s_3$, $s_i \in \mathbb{N}_0$, $s \geq 1$, and let $\mathbf{b} = (b_1, \ldots, b_{s_2})$, and $\mathbf{p} = (p_1, \ldots, p_{s_3})$ denote the bases of the associated representations of real numbers in subdimensions s_2 and s_3. We define the *hybrid function system* associated with this set of subdimensions s_j and this set of bases as the class of functions

$$\mathscr{F} = \{\xi_{\mathbf{k}} : \xi_{\mathbf{k}} = \mathrm{e}_{\mathbf{k}^{(1)}} \otimes w_{\mathbf{k}^{(2)}} \otimes \gamma_{\mathbf{k}^{(3)}},\ \mathbf{k}^{(1)} \in \mathbb{Z}^{s_1}, \mathbf{k}^{(2)} \in \mathbb{N}_0^{s_2}, \mathbf{k}^{(3)} \in \mathbb{N}_0^{s_3}\}.$$

We write $\mathscr{F}$ in the form $\mathscr{F} = \mathscr{T}^{(s_1)} \otimes \mathscr{W}_{\mathbf{b}}^{(s_2)} \otimes \Gamma_{\mathbf{p}}^{(s_3)}$.

Remark 3. All of the following results remain valid if we change the order of the factors in the hybrid function system $\mathscr{F}$, as it will become apparent from the proofs below.

For an integrable function f on $[0,1[^s$, the $\mathbf{k}$th Fourier coefficient of f with respect to the function system $\mathscr{F}$ is defined as

$$\hat{f}(\mathbf{k}) = \int_{[0,1[^s} f(\mathbf{x}) \overline{\xi_{\mathbf{k}}(\mathbf{x})}\, d\mathbf{x}.$$

The reader should notice that this definition encompasses the cases of Walsh and $\mathbf{p}$-adic Fourier coefficients, by putting $s = s_2$ or $s = s_3$.

Definition 7. Let $\mathbf{b} = (b_1, \ldots, b_s)$. A $\mathbf{b}$-*adic elementary interval*, or $\mathbf{b}$-*adic elint* for short, is a subinterval $I_{\mathbf{c},\mathbf{g}}$ of $[0,1[^s$ of the form

$$I_{\mathbf{c},\mathbf{g}} = \prod_{i=1}^{s} \left[\varphi_{b_i}(c_i), \varphi_{b_i}(c_i) + b_i^{-g_i}\right[,$$

where the parameters are subject to the conditions $\mathbf{g} = (g_1, \ldots, g_s) \in \mathbb{N}_0^s$, $\mathbf{c} = (c_1, \ldots, c_s) \in \mathbb{N}_0^s$, and $0 \leq c_i < b_i^{g_i}$, $1 \leq i \leq s$. We say that $I_{\mathbf{c},\mathbf{g}}$ belongs to the resolution class defined by $\mathbf{g}$ or that it has resolution $\mathbf{g}$.

A $\mathbf{b}$-*adic interval* in the resolution class defined by $\mathbf{g}$ (or with resolution $\mathbf{g}$) is a subinterval of $[0,1[^s$ of the form

$$\prod_{i=1}^{s} \left[a_i b_i^{-g_i}, d_i b_i^{-g_i}\right[, \quad 0 \leq a_i < d_i \leq b_i^{g_i},\ a_i, d_i, g_i \in \mathbb{N}_0,\ 1 \leq i \leq s.$$

For a given base $\mathbf{b}$ and for a given resolution $\mathbf{g} \in \mathbb{N}_0^s$, we define the following summation domains:

$$\Delta_{\mathbf{b}}(\mathbf{g}) = \left\{\mathbf{k} = (k_1, \ldots, k_s) \in \mathbb{N}_0^s : \; 0 \le k_i < b_i^{g_i}, 1 \le i \le s\right\},$$
$$\Delta_{\mathbf{b}}^*(\mathbf{g}) = \Delta_{\mathbf{b}}(\mathbf{g}) \setminus \{\mathbf{0}\}.$$

We note that $\Delta_{\mathbf{b}}(\mathbf{0}) = \{\mathbf{0}\}$.

In the following two lemmas, we recall the important fact that the Walsh and the **p**-adic Fourier series of the indicator functions $\mathbf{1}_I$ of elints I are finite and represent the function $\mathbf{1}_I$ pointwise (see Hellekalek [4, 6, 7]). In Lemma 2 below we have corrected the typo in the statement of Lemma 3.4 of [7, p. 277], where it should read '**p**'-adic elint.

Lemma 1. *Let* $I_{\mathbf{c},\mathbf{g}}$ *be an arbitrary* **b***-adic elint in* $[0,1[^s$ *and put* $f = \mathbf{1}_{I_{\mathbf{c},\mathbf{g}}} - \lambda_s(I_{\mathbf{c},\mathbf{g}})$. *Then, with respect to* $\mathcal{W}_{\mathbf{b}}^{(s)}$, *for all* $\mathbf{x} \in [0,1[^s$,

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \Delta_{\mathbf{b}}^*(\mathbf{g})} \hat{\mathbf{1}}_{I_{\mathbf{c},\mathbf{g}}}(\mathbf{k}) w_{\mathbf{k}}(\mathbf{x}).$$

Lemma 2. *Let* $I_{\mathbf{d},\mathbf{h}}$ *be an arbitrary* **p***-adic elint in* $[0,1[^s$ *and put* $f = \mathbf{1}_{I_{\mathbf{d},\mathbf{h}}} - \lambda_s(I_{\mathbf{d},\mathbf{h}})$. *Then, with respect to* $\Gamma_{\mathbf{p}}^{(s)}$, *for all* $\mathbf{x} \in [0,1[^s$,

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \Delta_{\mathbf{p}}^*(\mathbf{h})} \hat{\mathbf{1}}_{I_{\mathbf{d},\mathbf{h}}}(\mathbf{k}) \gamma_{\mathbf{k}}(\mathbf{x}).$$

For $\mathbf{k} = (k_1, \ldots, k_s) \in \mathbb{Z}^s$, let

$$M(\mathbf{k}) = \max_{1 \le i \le s} \mid k_i \mid .$$

For a positive integer t, we define the following weight function on $\mathbb{Z}^s$:

$$r_t(k_i) = \begin{cases} 1 & \text{if } k_i = 0, \\ \mid k_i \mid^{-t} & \text{if } k_i \ne 0, \end{cases} \quad r_t(\mathbf{k}) = \prod_{i=1}^{s} r_t(k_i). \tag{1}$$

Let $H \in \mathbb{N}$ be arbitrary and define, for parameters $t > 1$,

$$R_t = \sum_{\mathbf{k} \in \mathbb{Z}^s} r_t(\mathbf{k}), \quad R_t(H) = \sum_{\mathbf{k} \in \mathbb{Z}^s :\; 0 \le M(\mathbf{k}) \le H} r_t(\mathbf{k}).$$

In the definition of the hybrid diaphony in Sect. 4 below, we will make use of the fact that $R_2 = (1 + 2\zeta(2))^s = (1 + \pi^2/3)^s$.

For the Fourier series of indicator functions, Niederreiter [17, Lemma 2] has established the following result for the trigonometric function system.

Lemma 3. *Let* J *be an arbitrary subinterval of* $[0,1[$. *For every* $H \in \mathbb{N}$, *there exists a trigonometric polynomial*

$$P_J(x) = \sum_{k=-H}^{H} c_J(k)\mathrm{e}_k(x), \quad x \in [0,1[,$$

with complex coefficients $c_J(k)$, where $c_J(0) = \lambda_1(J)$ and $\mid c_J(k) \mid < r_1(k)$ for $k \neq 0$, such that, for all $x \in [0,1[$,

$$|\mathbf{1}_J(x) - P_J(x)| \leq \frac{1}{H+1} \sum_{k=-H}^{H} u_J(k)\mathrm{e}_k(x),$$

with complex numbers $u_J(k)$ satisfying $\mid u_J(k) \mid \leq 1$ for all k and $u_J(0) = 1$.

Corollary 1. *Let $s \geq 1$ and let J be an arbitrary subinterval of $[0,1[^s$. For every positive integer H, there exists a trigonometric polynomial P_J,*

$$P_J(\mathbf{x}) = \sum_{\substack{\mathbf{k}\in\mathbb{Z}^s:\\ 0\leq M(\mathbf{k})\leq H}} c_J(\mathbf{k})\mathrm{e}_{\mathbf{k}}(\mathbf{x}), \quad \mathbf{x} \in [0,1[^s,$$

with complex coefficients $c_J(\mathbf{k})$, where $c_J(\mathbf{0}) = \lambda_s(J)$ and $\mid c_J(\mathbf{k}) \mid < r_1(\mathbf{k})$ for $\mathbf{k} \neq \mathbf{0}$, such that, for all points $\mathbf{x} \in [0,1[^s$,

$$\mid \mathbf{1}_J(\mathbf{x}) - P_J(\mathbf{x}) \mid \leq \\ -1 + \sum_{\substack{\mathbf{k}\in\mathbb{Z}^s:\\ 0\leq M(\mathbf{k})\leq H}} \left(1 + \frac{1}{H+1}\right)^{s-\mathrm{wt}(\mathbf{k})} \frac{1}{(H+1)^{\mathrm{wt}(\mathbf{k})}} u_J(\mathbf{k})\mathrm{e}_{\mathbf{k}}(\mathbf{x}), \qquad (2)$$

with complex numbers $u_J(\mathbf{k})$ satisfying $\mid u_J(\mathbf{k}) \mid \leq 1$ for all $\mathbf{k}$ and $u_J(\mathbf{0}) = 1$. Here, $\mathrm{wt}(\mathbf{k})$ *denotes the Hamming weight of the vector* $\mathbf{k}$*, which is to say, the number of nonzero coordinates of* $\mathbf{k}$*.*

Proof. Let J_i, $1 \leq i \leq s$, denote the one-dimensional intervals such that $J = J_1 \times \cdots \times J_s$, and put $P_J(\mathbf{x}) = \prod_{i=1}^{s} P_{J_i}(x_i)$, where the trigonometric polynomials P_{J_i} are given by Lemma 3. We then proceed in the very same manner as in the proof of Theorem 1 in [17]. This yields (2). □

If $\omega = (\mathbf{x}_n)_{n\geq 0}$ is a (possibly finite) sequence in $[0,1[^s$ with at least N elements, and if $f : [0,1[^s \to \mathbb{C}$, we define

$$S_N(f,\omega) = \frac{1}{N}\sum_{n=0}^{N-1} f(\mathbf{x}_n).$$

Note that $S_N(\cdot,\omega)$ is a linear operator in the following sense: $S_N(f+g,\omega) = S_N(f,\omega) + S_N(g,\omega)$, and $S_N(cf,\omega) = cS_N(f,\omega)$, for all $c \in \mathbb{C}$.

We recall that a sequence ω is uniformly distributed in $[0,1[^s$ if and only if $\lim_{N\to\infty} S_N(\mathbf{1}_J,\omega) = \lambda_s(J)$, for all subintervals J of $[0,1[^s$, and that this limit relation extends to all Riemann integrable functions (see the monographs [12, 14]).

Theorem 1 (Hybrid Weyl Criterion). *Let $s \geq 1$, $s = s_1 + s_2 + s_3$, $s_j \in \mathbb{N}_0$, let $\mathbf{b} = (b_1,\dots,b_{s_2})$ be a vector of s_2 not necessarily distinct integers $b_i \geq 2$, and let $\mathbf{p} = (p_1,\dots,p_{s_3})$ be a vector of s_3 not necessarily distinct primes p_i. Put $\mathscr{F} = \mathscr{T}^{(s_1)} \otimes \mathscr{W}_{\mathbf{b}}^{(s_2)} \otimes \Gamma_{\mathbf{p}}^{(s_3)}$. Then, a sequence $\omega = (\mathbf{x}_n)_{n\geq 0}$ is uniformly distributed in $[0,1[^s$ if and only if for all functions $\xi_{\mathbf{k}} \in \mathscr{F}$, $\mathbf{k} \neq \mathbf{0}$,*

$$\lim_{N\to\infty} S_N(\xi_{\mathbf{k}},\omega) = 0. \tag{3}$$

Proof. Suppose first that ω is uniformly distributed in $[0,1[^s$. Each function $\xi_{\mathbf{k}}$ is Riemann-integrable. Further, for $\mathbf{k} \neq \mathbf{0}$, the integral of $\xi_{\mathbf{k}}$ is 0. Hence, in this case. the uniform distribution of ω implies that the sum $S_N(\xi_{\mathbf{k}},\omega)$ tends to 0 as N tends to infinity. This implies (3).

For the reverse direction, assume that (3) holds. In order to prove the uniform distribution of the sequence ω, it is enough to show $\lim_{N\to\infty} S_N(\mathbf{1}_J - \lambda_s(J),\omega) = 0$ for subintervals J of $[0,1[^s$ of the special form $J = J^{(1)} \times J^{(2)} \times J^{(3)}$, where $J^{(1)}$ is an arbitrary subinterval of $[0,1[^{s_1}$, $J^{(2)}$ is an arbitrary $\mathbf{b}$-adic subinterval of $[0,1[^{s_2}$ with resolution $\mathbf{g} \in \mathbb{N}^{s_2}$, and $J^{(3)}$ is an arbitrary $\mathbf{p}$-adic subinterval of $[0,1[^{s_3}$ with resolution $\mathbf{h} \in \mathbb{N}^{s_3}$. This follows easily from Lemma 3.9 and its proof in [7], when we apply the technique presented there to approximate arbitrary subintervals of $[0,1[^s$ by subintervals J of the special form above.

Hence, choose an arbitrary $H \in \mathbb{N}$ and arbitrary vectors $\mathbf{g} \in \mathbb{N}^{s_2}$, and $\mathbf{h} \in \mathbb{N}^{s_3}$, and let J be a subinterval of $[0,1[^s$ of the special form $J = J^{(1)} \times J^{(2)} \times J^{(3)}$ introduced above. We have

$$\begin{aligned} S_N(\mathbf{1}_J - \lambda_s(J),\omega) &= \frac{1}{N}\sum_{n=0}^{N-1} \mathbf{1}_{J^{(1)}}(\mathbf{x}_n^{(1)})\mathbf{1}_{J^{(2)}}(\mathbf{x}_n^{(2)})\mathbf{1}_{J^{(3)}}(\mathbf{x}_n^{(3)}) - \lambda_s(J) \\ &= \frac{1}{N}\sum_{n=0}^{N-1} \left(\mathbf{1}_{J^{(1)}}(\mathbf{x}_n^{(1)}) - P_{J^{(1)}}(\mathbf{x}_n^{(1)})\right)\mathbf{1}_{J^{(2)}}(\mathbf{x}_n^{(2)})\mathbf{1}_{J^{(3)}}(\mathbf{x}_n^{(3)}) \\ &\quad + \frac{1}{N}\sum_{n=0}^{N-1} P_{J^{(1)}}(\mathbf{x}_n^{(1)})\mathbf{1}_{J^{(2)}}(\mathbf{x}_n^{(2)})\mathbf{1}_{J^{(3)}}(\mathbf{x}_n^{(3)}) - \lambda_s(J) \qquad (4) \\ &= \Sigma_1 + \Sigma_2, \end{aligned}$$

where $P_{J^{(1)}}$ is a trigonometric polynomial which is given by Corollary 1, and Σ_1 and Σ_2 denote the two sums in (4).

In the estimate of Σ_1 below, we use the convention that the parameters on which the implied constant in a Landau symbol O depends are written in the subscript of O. Let $\omega^{(j)}$ denote the component sequence $(\mathbf{x}_n^{(j)})_{n\geq 0}$, $j = 1, 2, 3$. From (2) and from the proof of Theorem 1 in [17] we obtain the following bound:

$$\begin{aligned} |\ \Sigma_1\ | &\leq \frac{1}{N}\sum_{n=0}^{N-1}\left|\mathbf{1}_{J^{(1)}}(\mathbf{x}_n^{(1)}) - P_{J^{(1)}}(\mathbf{x}_n^{(1)})\right| \\ &= O_{s_1}\left(\frac{1}{H} + \sum_{\substack{\mathbf{k}^{(1)}\in\mathbb{Z}^{s_1}:\\ 0<M(\mathbf{k}^{(1)})\leq H}} r_1(\mathbf{k}^{(1)})\left|S_N(\mathrm{e}_{\mathbf{k}^{(1)}},\omega^{(1)})\right|\right). \end{aligned}$$

Condition (3) implies that Σ_1 tends to 0 if N tends to infinity.

In order to estimate Σ_2, we observe that Lemmas 1 and 2 and Corollary 1 imply the following pointwise Fourier series representations:

$$P_{J^{(1)}}(\mathbf{x}^{(1)}) = \sum_{0\leq M(\mathbf{k}^{(1)})\leq H} c_{J^{(1)}}(\mathbf{k}^{(1)})\mathrm{e}_{\mathbf{k}^{(1)}}(\mathbf{x}^{(1)}), \tag{5}$$

$$\mathbf{1}_{J^{(2)}}(\mathbf{x}^{(2)}) = \sum_{\mathbf{k}^{(2)}\in\Delta_{\mathbf{b}}(\mathbf{g})} \hat{\mathbf{1}}_{J^{(2)}}(\mathbf{k}^{(2)})w_{\mathbf{k}^{(2)}}(\mathbf{x}^{(2)}), \tag{6}$$

$$\mathbf{1}_{J^{(3)}}(\mathbf{x}^{(3)}) = \sum_{\mathbf{k}^{(3)}\in\Delta_{\mathbf{p}}(\mathbf{h})} \hat{\mathbf{1}}_{J^{(3)}}(\mathbf{k}^{(3)})\gamma_{\mathbf{k}^{(3)}}(\mathbf{x}^{(3)}). \tag{7}$$

We note that $c_{J^{(1)}}(\mathbf{0}^{(s_1)}) = \lambda_{s_1}(J^{(1)})$, $\hat{\mathbf{1}}_{J^{(2)}}(\mathbf{0}^{(s_2)}) = \lambda_{s_2}(J^{(2)})$, and $\hat{\mathbf{1}}_{J^{(3)}}(\mathbf{0}^{(s_3)}) = \lambda_{s_3}(J^{(3)})$. The linearity of the operator $S_N(\cdot,\omega)$ and identities (5)–(7) give

$$\begin{aligned} S_N(P_{J^{(1)}}\mathbf{1}_{J^{(2)}}\mathbf{1}_{J^{(3)}},\omega) = &\sum_{0\leq M(\mathbf{k}^{(1)})\leq H}\ \sum_{\mathbf{k}^{(2)}\in\Delta_{\mathbf{b}}(\mathbf{g})}\ \sum_{\mathbf{k}^{(3)}\in\Delta_{\mathbf{p}}(\mathbf{h})} \\ &\qquad c_{J^{(1)}}(\mathbf{k}^{(1)})\hat{\mathbf{1}}_{J^{(2)}}(\mathbf{k}^{(2)})\hat{\mathbf{1}}_{J^{(3)}}(\mathbf{k}^{(3)})S_N(\xi_{\mathbf{k}},\omega). \end{aligned} \tag{8}$$

Relation (3) implies

$$\lim_{N\to\infty} S_N(P_{J^{(1)}}\mathbf{1}_{J^{(2)}}\mathbf{1}_{J^{(3)}},\omega) = c_{J^{(1)}}(\mathbf{0}^{(s_1)})\hat{\mathbf{1}}_{J^{(2)}}(\mathbf{0}^{(s_2)})\hat{\mathbf{1}}_{J^{(3)}}(\mathbf{0}^{(s_3)}) = \lambda_s(J).$$

Hence, Σ_2 tends to 0 if N increases to infinity. This finishes the proof. □

Corollary 2. *Theorem 1 implies the classical Weyl criterion (see [12, Chap. 1.6, Theorem 6.2]), its Walsh version (see [5, Theorem 4.2]), and the* $\mathbf{p}$*-adic Weyl criterion (see [7, Theorem 3.11]).*

Corollary 3. *Let $s = s_1 + s_2$, $s_1, s_2 \geq 1$, let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{s_1}) \in \mathbb{R}^{s_1}$, and let $\mathbf{p} = (p_1, \ldots, p_{s_2})$, p_i prime, $1 \leq i \leq s_2$. For $n \in \mathbb{N}_0$, let $\mathbf{x}_n^{(1)} = n\boldsymbol{\alpha} \pmod 1$, and $\mathbf{x}_n^{(2)} = \varphi_{\mathbf{p}}(n)$, where we write $\varphi_{\mathbf{p}}(n) = (\varphi_{p_1}(n), \ldots, \varphi_{p_{s_2}}(n))$.*

If $\omega = \left((\mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)})\right)_{n\geq 0}$*, then this hybrid sequence is uniformly distributed in* $[0,1[^s$ *if and only if the following two conditions are satisfied:*

(i) $1, \alpha_1, \ldots, \alpha_{s_1}$ *are linearly independent over* $\mathbb{Q}$*, and*
(ii) *The primes* p_i *are distinct,* $1 \leq i \leq s_2$.

Proof. In the hybrid Weyl criterion, put $\mathscr{F} = \mathscr{T}^{(s_1)} \otimes \Gamma_{\mathbf{p}}^{(s_2)}$.

If ω is uniformly distributed in $[0,1[^s$, then the two projection sequences $(\mathbf{x}_n^{(1)})_{n\geq 0}$ and $(\mathbf{x}_n^{(2)})_{n\geq 0}$ will be uniformly distributed in $[0,1[^{s_1}$ and $[0,1[^{s_2}$, respectively. This implies conditions (i), as is well known (see [12, Chap. 1.6]), and (ii) above. The latter is elementary to verify.

If we assume conditions (i) and (ii), then, for arbitrary $\mathbf{k} \neq \mathbf{0}$, $S_N(\xi_{\mathbf{k}}, \omega) = (1/N)(C^N - 1)/(C-1)$, where $C = \mathrm{e}(k_1\alpha_1 + \cdots + k_{s_1}\alpha_{s_1} + \varphi_{p_1}(k_{s_1+1}) + \cdots + \varphi_{p_{s_2}}(k_{s_1+s_2}))$. We have $C \neq 1$, because otherwise a contradiction to condition (i) would arise. This implies $\lim_{N\to\infty} S_N(\xi_{\mathbf{k}}, \omega) = 0$. From the hybrid Weyl criterion, the uniform distribution of ω in $[0,1[^s$ follows. □

Remark 4. We note that Corollary 3 can also be derived from the proofs of Theorems 1 and 2 in Niederreiter [15], thus by a different approach. For related results, we refer the reader to Hofer and Kritzer [10] and Hofer and Larcher [11].

Corollary 4. *The hybrid function system* $\mathscr{F}$ *is an orthonormal basis of the space* $L^2([0,1[^s)$.

Proof. It is elementary to see that $\mathscr{F}$ is an orthonormal system in $L^2([0,1[^s)$. The idea of the proof is to show that the set of finite linear combinations of elements of $\mathscr{F}$ is dense in the set of functions $\mathbf{1}_J$, J an arbitrary subinterval of $[0,1[^s$, in the Hilbert space $L^2([0,1[^s)$. From this, it follows by a standard argument that $\mathscr{F}$ is an orthonormal basis.

Hence, let J be an arbitrary subinterval of $[0,1[^s$. We have $J = J^{(1)} \times J^{(2)} \times J^{(3)}$, where $J^{(j)}$ is a subinterval of $[0,1[^{s_j}$, $j = 1,2,3$. As $\mathscr{T}^{(s_1)}$ is an orthonormal basis of $L^2([0,1[^{s_1})$, we may approximate $\mathbf{1}_{J^{(1)}}$ arbitrarily closely in $L^2([0,1[^{s_1})$ by trigonometric polynomials. From the proof of Lemma 3.9 in [7] it follows that we may approximate $\mathbf{1}_{J^{(2)}}$ arbitrarily closely in $L^2([0,1[^{s_2})$ by functions of the form $\mathbf{1}_I$, where I is a $\mathbf{b}$-adic interval in $[0,1[^{s_2}$. As Identity (6) above shows, every function $\mathbf{1}_I$ is a Walsh polynomial, that is to say, a finite linear combination of elements of $\mathscr{W}_{\mathbf{b}}^{(s_2)}$. The same reasoning may be applied to $\mathbf{1}_{J^{(3)}}$, with respect to the function system $\Gamma_{\mathbf{p}}^{(s_3)}$, see Identity (7). Altogether, this implies that $\mathbf{1}_J$ can be approximated arbitrarily closely in $L^2([0,1[^s)$ by finite linear combinations of elements of $\mathscr{F}$. □

Remark 5. Corollary 4 generalizes Theorem A.11 in Dick and Pillichshammer [1, p. 562] and, in addition, presents a different method of proof.

4 Hybrid Diaphony

For a given base $\mathbf{b}$, and a vector $\mathbf{k} = (k_1, \dots, k_s) \in \mathbb{N}_0^s$, let

$$\rho_{b_i}(k_i) = \begin{cases} 1 & \text{if } k = 0, \\ b_i^{-2(t_i-1)} & \text{if } b_i^{t_i-1} \le k_i < b_i^{t_i},\ t_i \in \mathbb{N}, \end{cases} \qquad \rho_{\mathbf{b}}(\mathbf{k}) = \prod_{i=1}^{s} \rho_{b_i}(k_i).$$

It is elementary to prove that, for $\mathbf{g} = (g_1, \dots, g_s) \in \mathbb{N}_0^s$, the sum $\sigma_{\mathbf{b}}$ of all weights $\rho_{\mathbf{b}}(\mathbf{k})$ and the truncated sum $\sigma_{\mathbf{b}}(\mathbf{g})$ are given by the formulas

$$\sigma_{\mathbf{b}} = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \rho_{\mathbf{b}}(\mathbf{k}) = \prod_{i=1}^{s} (b_i + 1), \tag{9}$$

$$\sigma_{\mathbf{b}}(\mathbf{g}) = \sum_{\mathbf{k} \in \Delta_{\mathbf{b}}(\mathbf{g})} \rho_{\mathbf{b}}(\mathbf{k}) = \prod_{i=1}^{s} \left(b_i + 1 - b_i^{-g_i+1} \right). \tag{10}$$

Definition 8. Let $s \ge 1$, $s = s_1 + s_2 + s_3$, $s_j \in \mathbb{N}_0$, $1 \le j \le 3$, let $\mathbf{b} = (b_1, \dots, b_{s_2})$ be a vector of s_2 not necessarily distinct integers $b_i \ge 2$, and let $\mathbf{p} = (p_1, \dots, p_{s_3})$ be a vector of s_3 not necessarily distinct primes p_i. Put $\mathscr{F} = \mathscr{T}^{(s_1)} \otimes \mathscr{W}_{\mathbf{b}}^{(s_2)} \otimes \Gamma_{\mathbf{p}}^{(s_3)}$.

The *hybrid diaphony* $F_N(\omega)$ of the first N elements of a sequence $\omega = (\mathbf{x}_n)_{n \ge 0}$ in $[0,1[^s$ with respect to the function system $\mathscr{F}$ and the weight function ρ is defined by

$$F_N(\omega) = \left(\frac{1}{\sigma - 1} \sum_{\mathbf{k} \ne \mathbf{0}} \rho(\mathbf{k}) \, |S_N(\xi_{\mathbf{k}}, \omega)|^2 \right)^{1/2},$$

where ρ is given by the product of weights

$$\rho(\mathbf{k}) = r_2(\mathbf{k}^{(1)}) \rho_{\mathbf{b}}(\mathbf{k}^{(2)}) \rho_{\mathbf{p}}(\mathbf{k}^{(3)}),$$

$\mathbf{k} = (\mathbf{k}^{(1)}, \mathbf{k}^{(2)}, \mathbf{k}^{(3)}) \in \mathbb{Z}^{s_1} \times \mathbb{N}_0^{s_2} \times \mathbb{N}_0^{s_3}$. The normalizing constant σ is defined as $\sigma = R_2 \sigma_{\mathbf{b}} \sigma_{\mathbf{p}}$, where $R_2 = (1 + \pi^2/3)^{s_1}$ (see (1)), and $\sigma_{\mathbf{b}}$ and $\sigma_{\mathbf{p}}$ are given by (9).

Remark 6. Definition 8 generalizes the classical diaphony of Zinterhof [20], see also Kuipers and Niederreiter [12, Exercise 5.27, p. 162], the dyadic diaphony of Hellekalek and Leeb [8], its generalizations to the b-adic case by Grozdanov et al. [2,3], and also the recent version of diaphony based on p-adic arithmetic introduced by Hellekalek [7].

In the following theorem, we prove that F_N is a measure of uniform distribution of sequences in $[0,1[^s$.

Theorem 2. *Let ω be a sequence in $[0,1[^s$. Then the hybrid diaphony F_N defined by the hybrid function system $\mathcal{F} = \mathcal{T}^{(s_1)} \otimes \mathcal{W}_{\mathbf{b}}^{(s_2)} \otimes \Gamma_{\mathbf{p}}^{(s_3)}$ and the weight function ρ has the following properties:*

(i) *F_N is normalized:* $0 \leq F_N(\omega) \leq 1$,
(ii) *ω is uniformly distributed in* $[0,1[^s$ *if and only if* $\lim_{N\to\infty} F_N(\omega) = 0$.

Proof. For every $\mathbf{k}$, $|S_N(\xi_{\mathbf{k}},\omega)| \leq 1$. We have the identity $\sigma - 1 = \sum_{\mathbf{k}\neq\mathbf{0}} \rho(\mathbf{k})$, which implies (i).

In (ii), let $\lim_{N\to\infty} F_N(\omega) = 0$. As a consequence, $\lim_{N\to\infty} S_N(\xi_{\mathbf{k}},\omega) = 0$ for all $\mathbf{k} \neq \mathbf{0}$. The hybrid Weyl criterion implies the uniform distribution of ω.

For the reverse direction, let ω be uniformly distributed in $[0,1[^s$. Let $H \in \mathbb{N}$, $\mathbf{g} \in \mathbb{N}^{s_2}$, and $\mathbf{h} \in \mathbb{N}^{s_3}$ be arbitrary and define the summation domains

$$\begin{aligned}
\Delta(H,\mathbf{g},\mathbf{h}) &= \big\{\mathbf{k} = (\mathbf{k}^{(1)},\mathbf{k}^{(2)},\mathbf{k}^{(3)}) \in \mathbb{Z}^{s_1} \times \mathbb{N}_0^{s_2} \times \mathbb{N}_0^{s_3} : \\
&\qquad 0 \leq M(\mathbf{k}^{(1)}) \leq H, \mathbf{k}^{(2)} \in \Delta_{\mathbf{b}}(\mathbf{g}), \mathbf{k}^{(3)} \in \Delta_{\mathbf{p}}(\mathbf{h})\big\}, \\
\Delta^*(H,\mathbf{g},\mathbf{h}) &= \Delta(H,\mathbf{g},\mathbf{h}) \setminus \{\mathbf{0}\}, \\
\Delta(H,\mathbf{g},\mathbf{h})^c &= \mathbb{Z}^{s_1} \times \mathbb{N}_0^{s_2} \times \mathbb{N}_0^{s_3} \setminus \Delta(H,\mathbf{g},\mathbf{h}).
\end{aligned}$$

Then we have the following upper bound:

$$F_N^2(\omega) \leq \frac{1}{\sigma-1} \sum_{\mathbf{k}\in\Delta^*(H,\mathbf{g},\mathbf{h})} \rho(\mathbf{k})\,|S_N(\xi_{\mathbf{k}},\omega)|^2 + \frac{1}{\sigma-1} \sum_{\mathbf{k}\in\Delta(H,\mathbf{g},\mathbf{h})^c} \rho(\mathbf{k}).$$

If we put

$$\sigma(H,\mathbf{g},\mathbf{h}) = \sum_{\mathbf{k}\in\Delta(H,\mathbf{g},\mathbf{h})} \rho(\mathbf{k}),$$

then we obtain the inequality

$$F_N^2(\omega) \leq \frac{1}{\sigma-1} \sum_{\mathbf{k}\in\Delta^*(H,\mathbf{g},\mathbf{h})} \rho(\mathbf{k})\,|S_N(\xi_{\mathbf{k}},\omega)|^2 + \frac{\sigma - \sigma(H,\mathbf{g},\mathbf{h})}{\sigma-1}. \tag{11}$$

From the uniform distribution of ω it follows, by an application of the hybrid Weyl criterion, that $\lim_{N\to\infty} S_N(\xi_{\mathbf{k}},\omega) = 0$, for all $\mathbf{k} \neq \mathbf{0}$. The summation domain $\Delta^*(H,\mathbf{g},\mathbf{h})$ is finite, hence

$$\limsup_{N\to\infty} F_N^2(\omega) \leq \frac{\sigma - \sigma(H,\mathbf{g},\mathbf{h})}{\sigma-1}.$$

The difference $\sigma - \sigma(H,\mathbf{g},\mathbf{h})$ can be made arbitrarily small, by increasing H and every component of the vectors $\mathbf{g}$ and $\mathbf{h}$. This implies the existence of $\lim_{N\to\infty} F_N^2(\omega)$ and yields $\lim_{N\to\infty} F_N^2(\omega) = 0$. □

Inequalities of the Erdös-Turán-Koksma type provide an upper bound for the given measure of uniform distribution, like discrepancy or diaphony, in terms of certain exponential sums. For the hybrid diaphony, we obtain the following result.

Corollary 5. *Let $H \in \mathbb{N}$, $\mathbf{g} = (g_1, \ldots, g_{s_2}) \in \mathbb{N}^{s_2}$, and $\mathbf{h} = (h_1, \ldots, h_{s_3}) \in \mathbb{N}^{s_3}$ be arbitrary. Then the inequality of Erdös-Turán-Koksma for the hybrid diaphony is given by*

$$F_N^2(\omega) \leq \frac{\sigma}{\sigma - 1} s\delta + \frac{1}{\sigma - 1} \sum_{\mathbf{k} \in \Delta^*(H, \mathbf{g}, \mathbf{h})} \rho(\mathbf{k}) \left|S_N(\xi_{\mathbf{k}}, \omega)\right|^2, \tag{12}$$

where

$$\delta = \max\left\{ \frac{2}{(1 + \pi^2/3)H}, \max_{1 \leq i \leq s_2} (b_i + 1)^{-1} b_i^{-g_i+1}, \max_{1 \leq i \leq s_3} (p_i + 1)^{-1} p_i^{-h_i+1} \right\}.$$

Proof. We estimate the error term in (11) as follows. We have

$$\frac{\sigma - \sigma(H, \mathbf{g}, \mathbf{h})}{\sigma - 1} = \frac{\sigma}{\sigma - 1}\left(1 - \frac{\sigma(H, \mathbf{g}, \mathbf{h})}{\sigma}\right),$$

and

$$\frac{\sigma(H, \mathbf{g}, \mathbf{h})}{\sigma} = \frac{R_2(H)}{R_2} \frac{\sigma_{\mathbf{b}}(\mathbf{g})}{\sigma_{\mathbf{b}}} \frac{\sigma_{\mathbf{p}}(\mathbf{h})}{\sigma_{\mathbf{p}}}.$$

Hence, by an elementary estimate for the quotient $R_2(H)/R_2$ and by applying Identity (10),

$$\frac{\sigma(H, \mathbf{g}, \mathbf{h})}{\sigma} > \left(1 - \frac{2}{R_2 H}\right)^{s_1} \prod_{i=1}^{s_2}(1 - (b_i + 1)^{-1} b_i^{-g_i+1}) \times \prod_{i=1}^{s_3}(1 - (p_i + 1)^{-1} p_i^{-h_i+1}).$$

An application of Lemma 3.9 of [14] yields the estimate

$$1 - \frac{\sigma(H, \mathbf{g}, \mathbf{h})}{\sigma} \leq 1 - (1 - \delta)^s,$$

which is easily seen to be bounded by $s\delta$. □

Remark 7. Our choice of components in a hybrid function system is motivated as follows. Concerning the first component, the trigonometric function system is the system of choice when it comes to analyzing the uniform distribution of sequences that are based on addition modulo one, like $(n\alpha)_{n\geq 0}$ sequences or good lattice points. Concerning the remaining components, we observe that important construction methods for sequences with good uniform distribution on the s-torus employ the representation of real numbers in some integer base b. The resulting (finite and infinite) *digital sequences* in base b are constructed by arithmetic operations applied to digit vectors. We refer the reader to Niederreiter [14] and to the recent monograph Dick and Pillichshammer [1] for details.

In the analysis of these digital sequences, addition of digit vectors plays a central role. This leads to the following question: what are the different possibilities to add digit vectors and which are the function systems associated with different types of addition? We will address this question in a forthcoming paper, where we show that, essentially, Walsh and p-adic function systems cover all possible cases.

As a consequence, if a hybrid sequence employs construction principles like addition modulo one or if digital sequences come into play, the appropriate function systems for analysis will be of the form introduced in Sect. 3.

Remark 8. It is a natural question to study how to extend the p-adic concepts introduced in this paper and in Hellekalek [7] from the case of a prime base p to a general integer base $b \geq 2$. This will be a theme of future research, in which some technical problems that arise in this context will have to be overcome.

Acknowledgements The author would like to thank Markus Neuhauser, NUHAG, University of Vienna, Austria, and RWTH Aachen, Germany, and Harald Niederreiter, University of Salzburg, and RICAM, Austrian Academy of Sciences, Linz, for several helpful comments.

References

1. Dick, J., Pillichshammer, F.: Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press, Cambridge (2010)
2. Grozdanov, V., Nikolova, E., Stoilova, S.: Generalized b-adic diaphony. C. R. Acad. Bulgare Sci. **56**(4), 23–30 (2003)
3. Grozdanov, V.S., Stoilova, S.S.: On the theory of b-adic diaphony. C. R. Acad. Bulgare Sci. **54**(3), 31–34 (2001)
4. Hellekalek, P.: General discrepancy estimates: the Walsh function system. Acta Arith. **67**, 209–218 (1994)
5. Hellekalek, P.: On the assessment of random and quasi-random point sets. In: P. Hellekalek, G. Larcher (eds.) Pseudo and Quasi-Random Point Sets, *Lecture Notes in Statistics*, vol. 138, pp. 49–108. Springer, New York (1998)
6. Hellekalek, P.: A general discrepancy estimate based on p-adic arithmetics. Acta Arith. **139**, 117–129 (2009)
7. Hellekalek, P.: A notion of diaphony based on p-adic arithmetic. Acta Arith. **145**, 273–284 (2010)
8. Hellekalek, P., Leeb, H.: Dyadic diaphony. Acta Arith. **80**, 187–196 (1997)
9. Hewitt, E., Ross, K.A.: Abstract harmonic analysis. Vol. I, *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, vol. 115, second edn. Springer-Verlag, Berlin (1979)
10. Hofer, R., Kritzer, P.: On hybrid sequences built of Niederreiter-Halton sequences and Kronecker sequences. Bull. Austral. Math. Soc. (2011). To appear

11. Hofer, R., Larcher, G.: Metrical results on the discrepancy of Halton-Kronecker sequences. Mathematische Zeitschrift (2011). To appear
12. Kuipers, L., Niederreiter, H.: Uniform Distribution of Sequences. John Wiley, New York (1974). Reprint, Dover Publications, Mineola, NY, 2006
13. Mahler, K.: Lectures on diophantine approximations. Part I: g-adic numbers and Roth's theorem. University of Notre Dame Press, Notre Dame, Ind (1961)
14. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
15. Niederreiter, H.: On the discrepancy of some hybrid sequences. Acta Arith. **138**(4), 373–398 (2009)
16. Niederreiter, H.: A discrepancy bound for hybrid sequences involving digital explicit inversive pseudorandom numbers. Unif. Distrib. Theory **5**(1), 53–63 (2010)
17. Niederreiter, H.: Further discrepancy bounds and an Erdös-Turán-Koksma inequality for hybrid sequences. Monatsh. Math. **161**, 193–222 (2010)
18. Schipp, F., Wade, W., Simon, P.: Walsh Series. An Introduction to Dyadic Harmonic Analysis. With the collaboration of J. Pál. Adam Hilger, Bristol and New York (1990)
19. 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 (Las Vegas, NV, 1994), *Lecture Notes in Statist.*, vol. 106, pp. 121–148. Springer, New York (1995)
20. Zinterhof, P.: Über einige Abschätzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden. Sitzungsber. Österr. Akad. Wiss. Math.-Natur. Kl. II **185**, 121–132 (1976)

An Intermediate Bound on the Star Discrepancy

Stephen Joe

Abstract Let $P_n(z)$ denote the point set of an n-point rank-1 lattice rule with generating vector z. A criterion used to assess the 'goodness' of the point set $P_n(z)$ is the star discrepancy, $D^*(P_n(z))$. As calculating the star discrepancy is an NP-hard problem, then it is usual to work with bounds on it. In particular, it is known that the following two bounds hold:

$$D^*(P_n(z)) \le 1 - (1 - 1/n)^d + T(z,n) \le 1 - (1 - 1/n)^d + R(z,n)/2,$$

where d is the dimension and the quantities $T(z,n)$ and $R(z,n)$ are defined in the paper. Here we provide an intermediate bound on the star discrepancy by introducing a new quantity $W(z,n)$ which satisfies

$$T(z,n) \le W(z,n) \le R(z,n)/2.$$

Like $R(z,n)$, the quantity $W(z,n)$ may be calculated to a fixed precision in $O(nd)$ operations. A component-by-component construction based on $W(z,n)$ is analysed. We present the results of numerical calculations which indicate that values of $W(z,n)$ are much closer to $T(z,n)$ than to $R(z,n)/2$.

1 Introduction

We wish to approximate the d-dimensional integral given by

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

S. Joe
Department of Mathematics, The University of Waikato, Private Bag 3105, Hamilton, 3240, New Zealand
e-mail: stephenj@math.waikato.ac.nz

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_25, © Springer-Verlag Berlin Heidelberg 2012

A well-known method is to use an n-point rank-1 lattice rule given by

$$Q_{n,d}(f) = \frac{1}{n}\sum_{k=0}^{n-1} f\left(\left\{\frac{kz}{n}\right\}\right),$$

where $z \in \mathbb{Z}^d$ has no factor in common with n and the braces around a vector indicate that we take the fractional part of each component.

A criterion used to assess the 'goodness' of the point set $P_n(z) := \{\{kz/n\},\ 0 \leq k \leq n-1\}$ is the star discrepancy defined by

$$D^*(P_n(z)) := \sup_{\boldsymbol{x}\in[0,1)^d} |\mathrm{discr}(\boldsymbol{x}, P_n)|,$$

where discr$(\boldsymbol{x}, P_n)$ is the 'local discrepancy' defined by

$$\mathrm{discr}(\boldsymbol{x}, P_n) := \frac{|P_n(z) \cap [\mathbf{0}, \boldsymbol{x})|}{n} - \mathrm{Vol}([\mathbf{0}, \boldsymbol{x})).$$

The star discrepancy occurs in the well-known Koksma-Hlawka inequality given by

$$|I_d(f) - Q_{n,d}(f)| \leq D^*(P_n(z))V(f),$$

where $V(f)$ is the variation of f in the sense of Hardy and Krause. Further details may be found in [5] and [11] or in more general works such as [9]. For simplicity, we shall work with the star discrepancy defined above although the results presented here can be generalized to the weighted star discrepancy.

Though there exist algorithms which calculate the star discrepancy or approximate it to a user-specified error, these have running times which are exponential in d as pointed out in [3]. In fact, the paper [3] shows that calculation of the star discrepancy is an NP-hard problem. These computational difficulties make it hard to work with the star discrepancy directly in computations. Instead, we work with bounds on the star discrepancy such as those given by the quantities $R(z,n)$ and $T(z,n)$, which we define shortly. So, for example, the component-by-component construction given in [6] to find rank-1 lattice rules with $O(n^{-1}(\ln(n))^d)$ star discrepancy is based on a search using $R(z,n)$.

Following [9], suppose $n \geq 2$ and let $C(n) = (-n/2, n/2] \cap \mathbb{Z}$ with $C^*(n) = C(n)\backslash\{0\}$. Moreover, let $C_d(n)$ be the Cartesian product of d copies of $C(n)$ with $C_d^*(n) = C_d(n)\backslash\{\mathbf{0}\}$. For $h \in C(n)$, set

$$r(h) = \max(1, |h|) \quad \text{and} \quad t(h,n) = \begin{cases} n\sin(\pi|h|/n) & \text{for } h \in C^*(n), \\ 1 & \text{for } h = 0. \end{cases} \tag{1}$$

For $\mathbf{h} = (h_1, \ldots, h_d) \in C_d(n)$, we then set

$$r(\mathbf{h}) = \prod_{i=1}^{d} r(h_i) \quad \text{and} \quad t(\mathbf{h}, n) = \prod_{i=1}^{d} t(h_i, n).$$

With

$$R(z, n) = \sum_{\mathrm{h}} \frac{1}{r(\mathbf{h})} \quad \text{and} \quad T(z, n) = \sum_{\mathbf{h}} \frac{1}{t(\mathbf{h}, n)},$$

where the sums are over all $\mathbf{h} \in C_d^*(n)$ satisfying $\mathbf{h} \cdot z \equiv 0 \mod n$, Theorem 5.6 of [9] yields

$$D^*(P_n(z)) \leq 1 - \left(1 - \frac{1}{n}\right)^d + T(z, n) \leq 1 - \left(1 - \frac{1}{n}\right)^d + \frac{1}{2} R(z, n). \tag{2}$$

Previous papers such as [6] have used the bound on the star discrepancy based on the quantity $R(z, n)/2$ (see (2)). By writing $R(z, n)$ as a quadrature error (see (10) in Sect. 3), one observes that the calculation of $R(z, n)$ for a given z requires $O(n^2 d)$ operations. However, the asymptotic expansion in [7] allows this quantity to be calculated to a fixed precision in $O(nd)$ operations.

In the next section, we introduce a quantity $W(z, n)$ such that

$$T(z, n) \leq W(z, n) \leq \frac{1}{2} R(z, n). \tag{3}$$

In Sect. 5 we give numerical values of these three quantities. These numerical results suggest that the bounds on the star discrepancy obtained from $W(z, n)$ can be significantly better than those obtained from $R(z, n)/2$. Of course, the bound on the star discrepancy based on $T(z, n)$ would be even better. However, we shall see in Sect. 3 that, as for $R(z, n)$, $W(z, n)$ may be calculated to a fixed precision for a given z in $O(nd)$ operations. Attempts to calculate $T(z, n)$ to a fixed precision also in $O(nd)$ operations did not prove fruitful.

A component-by-component construction based on $W(z, n)$ is analysed in Sect. 4. As in [6], the construction yields a z for which $D^*(P_n(z)) = O(n^{-1}(\ln(n))^d)$. However, the implied constant is smaller.

The definition of the star discrepancy means that it is bounded above by one. It should be pointed out that the numerical results in Sect. 5 and [10] indicate that the values of the bounds on the star discrepancy given in (2) or based on $W(z, n)$ can be much greater than one, even in moderate dimensions. Hence there is a large gap between the true values of the star discrepancy and the bounds. This is not too surprising since the bounds obtained are essentially $O(n^{-1}(\ln(n))^d)$. However, the function $n^{-1}(\ln(n))^d$ considered as a function of n is increasing for $n \leq e^d$ as discussed, for example, in [2, p. 88].

2 An Intermediate Bound on the Star Discrepancy

Since $1/\sin(\pi x) \le 1/(2x)$ for $x \in (0, 1/2]$, it follows from the definition of $r(h)$ and $t(h, n)$ in (1) that $1/t(h, n) \le 1/(2r(h))$ for $h \in C^*(n)$. Moreover, $1/t(0, n) = 1/r(0) = 1$. Since $\boldsymbol{h} \in C_d^*(n)$ has at least one non-zero component, this then leads to the inequality $T(z, n) \le R(z, n)/2$ seen in (2).

In order to find a quantity $W(z, n)$ such that $T(z, n) \le W(z, n) \le R(z, n)/2$, we will find a quantity $w(h, n)$ such that $1/w(0, n) = 1$ and $1/t(h, n) \le 1/w(h, n) \le 1/(2r(h))$ for $h \in C^*(n)$. This last requirement is equivalent to

$$\frac{1}{\sin(\pi|h|/n)} \le \frac{1}{w(h, n)/n} \le \frac{1}{2|h|/n}. \tag{4}$$

For $h \in C^*(n)$, we have $0 < |h|/n \le 1/2$. So if we can find a function G such that $1/\sin(\pi x) \le G(x) \le 1/(2x)$ for $x \in (0, 1/2]$ and take $w(h, n) = n/G(|h|/n)$, we then see that (4) is satisfied.

We shall construct such a function G. The function that is constructed is piecewise on $(0, 1/2]$ and consists of two 'pieces'. In particular, let $\kappa = 0.46$, $\kappa_1 \approx 1.102449$, and $\kappa_2 \approx -0.204898$. Then G is defined by

$$G(x) := \begin{cases} 1/(\pi x) + \pi x/6 + 7\pi^3/2880 & \text{for } x \in (0, \kappa], \\ \kappa_1 + \kappa_2 x & \text{for } x \in (\kappa, 1/2]. \end{cases} \tag{5}$$

Exact expressions for κ_1 and κ_2 are given later in (8) and these values are such that G is continuous at $x = \kappa$ and $G(1/2) = 1$.

We now prove that this G satisfies the required bounds. We start by showing that the first 'piece' on $(0, \kappa]$ is bounded below by $1/\sin(\pi x)$ for x in this interval.

Lemma 1. *Let $g_1(x) = 1/\sin(\pi x) - 1/(\pi x) - \pi x/6 - 7\pi^3/2880$ and let $\kappa = 0.46$. Then $g_1(x) < 0$ for $x \in (0, \kappa]$.*

Proof. We first show that the function g_1 is an increasing function on the interval $(0, 1/2]$. Straight-forward differentiation leads to

$$g_1'(x) = \frac{-\pi\cos(\pi x)}{\sin^2(\pi x)} + \frac{1}{\pi x^2} - \frac{\pi}{6} = \frac{(1 - \pi^2 x^2/6)\sin^2(\pi x) - \pi^2 x^2 \cos(\pi x)}{\pi x^2 \sin^2(\pi x)}.$$

This derivative clearly exists for $x \in (0, 1/2]$ and we now show that $g_1'(x) > 0$ on this interval.

Elementary calculus shows that

$$\sin(\pi x) \ge \pi x - \frac{(\pi x)^3}{6}, \quad x \in (0, 1/2],$$

and

$$\cos(\pi x) \le 1 - \frac{(\pi x)^2}{2} + \frac{(\pi x)^4}{24}, \quad x \in (0, 1/2].$$

For $x \in (0, 1/2]$, we have $1 - \pi^2 x^2/6 > 0$. It then follows that

$$\begin{aligned} g_1'(x) &= \frac{(1 - \pi^2 x^2/6)\sin^2(\pi x) - \pi^2 x^2 \cos(\pi x)}{\pi x^2 \sin^2(\pi x)} \\ &\ge \frac{(1 - \pi^2 x^2/6)(\pi x - (\pi x)^3/6)^2 - \pi^2 x^2 (1 - (\pi x)^2/2 + (\pi x)^4/24)}{\pi x^2 \sin^2(\pi x)} \\ &= \frac{(\pi x)^6 [9 - (\pi x)^2]}{216 \pi x^2 \sin^2(\pi x)}. \end{aligned}$$

We have $0 < (\pi x)^2 < 3 < 9$ for $x \in (0, 1/2]$. Hence $g_1'(x) > 0$ for x in this interval and so g_1 is an increasing function on the interval.

Now $g_1(x) = 0$ for $x = 0.4604264347$ (to ten decimal places). For our purposes, it is enough to use the approximation $\kappa = 0.46$ which is just slightly less than this value. As expected, a direct calculation shows that $g_1(\kappa) < 0$. Since $g_1'(x) > 0$ for $x \in (0, 1/2]$, we must have $g_1(x) \le g_1(\kappa) < 0$ for $x \in (0, \kappa]$. This then completes the proof. □

Corollary 1. *The G defined in (5) satisfies*

$$\frac{1}{\sin(\pi x)} < G(x) = \frac{1}{\pi x} + \frac{\pi x}{6} + \frac{7\pi^3}{2880}, \quad x \in (0, \kappa]. \tag{6}$$

Remark 1. The approximation $G(x)$ to $1/\sin(\pi x)$ for $x \in (0, \kappa]$ arises from the Laurent series of $1/\sin(\pi x)$ given by (for example, see [4, p. 43])

$$\frac{1}{\sin(\pi x)} = \frac{1}{\pi x} + \frac{\pi x}{6} + \frac{7\pi^3 x^3}{360} + \sum_{i=3}^{\infty} \frac{2(2^{2i-1} - 1)|B_{2i}|}{(2i)!} (\pi x)^{2i-1}, \ 0 < |x| < 1,$$

where B_{2i} is the $(2i)$-th Bernoulli number. The function G that we construct is piecewise. It would be possible to use the function $\widetilde{G}$ for the whole interval $(0, 1/2]$, where

$$\widetilde{G}(x) := \frac{1}{\pi x} + \frac{\pi x}{6} + 1 - \frac{2}{\pi} - \frac{\pi}{12}. \tag{7}$$

This function satisfies $\widetilde{G}(1/2) = 1$. However, the bounds obtained on the star discrepancy with the G we construct are slightly better since $G(x) \le \widetilde{G}(x)$ for all $x \in (0, 1/2]$. A proof of this inequality is given later in Sect. 4.

To obtain an appropriate $G(x)$ for $x \in (\kappa, 1/2]$, let $p(x) = \kappa_1 + \kappa_2 x$ be the linear interpolating function on $[\kappa, 1/2]$ such that $p(\kappa) = G(\kappa) = 1/(\pi\kappa) + \pi\kappa/6 + 7\pi^3/2880 \approx 1.008196$ and $p(1/2) = 1$. (This choice of $p(\kappa)$ ensures that G is continuous at $x = \kappa$ and hence continuous on the whole interval $(0, 1/2]$.) Then setting up the linear equations and solving, we find that

$$\kappa_1 = \frac{p(\kappa) - 2\kappa}{1 - 2\kappa} \approx 1.102449 \quad \text{and} \quad \kappa_2 = \frac{2(1 - p(\kappa))}{1 - 2\kappa} \approx -0.204898. \tag{8}$$

Lemma 2. *With $\kappa = 0.46$ and κ_1 and κ_2 given in (8), let $p(x) = \kappa_1 + \kappa_2 x$ and $g_2(x) = 1/\sin(\pi x) - p(x)$ for $x \in [\kappa, 1/2]$. Then $g_2(x) \le 0$ for x in this interval.*

Proof. We have

$$g_2'(x) = \frac{-\pi\cos(\pi x)}{\sin^2(\pi x)} - \kappa_2 = \frac{-\kappa_2 \sin^2(\pi x) - \pi\cos(\pi x)}{\sin^2(\pi x)}.$$

By substituting $\sin^2(\pi x) = 1 - \cos^2(\pi x)$ into the numerator of this last expression, we see that with $v(x) = \cos(\pi x)$, then $g_2'(x) = 0$ when $\kappa_2(v(x))^2 - \pi v(x) - \kappa_2 = 0$. The quadratic formula yields

$$v(x) = \frac{\pi \pm \sqrt{\pi^2 + 4\kappa_2^2}}{2\kappa_2} \approx -15.397402,\ 0.064946.$$

Since $v(x) = \cos(\pi x)$, there is just one value of $x \in [\kappa, 1/2]$ such that $g_2'(x) = 0$, namely $x \approx \cos^{-1}(0.064946)/\pi \approx 0.479312$. We denote the exact value of this x by μ. Also, the quotient rule yields

$$g_2''(x) = \frac{\pi^2 \sin^3(\pi x) + 2\pi^2 \sin(\pi x)\cos^2(\pi x)}{\sin^4(\pi x)} = \frac{\pi^2}{\sin(\pi x)} + \frac{2\pi^2\cos^2(\pi x)}{\sin^3(\pi x)}.$$

Then $g_2''(x)$ is clearly positive for $x \in (\kappa, 1/2)$, so that g_2' is an increasing function on $[\kappa, 1/2]$. As a result, $g_2'(x)$ is negative for $x \in [\kappa, \mu)$, zero at $x = \mu$, and positive for $x \in (\mu, 1/2]$.

By taking $x = \kappa$ in (6), we have

$$\frac{1}{\sin(\pi\kappa)} < \frac{1}{\pi\kappa} + \frac{\pi\kappa}{6} + \frac{7\pi^3}{2880} = p(\kappa),$$

and hence $g_2(\kappa) < 0$. Because of the behavior of $g_2'(x)$ described above, we see that $g_2(x)$ decreases from $x = \kappa$ onwards until x reaches the turning point at $x = \mu$. Then $g_2(x)$ increases from $x = \mu$ onwards until $x = 1/2$ is reached. Since $g_2(1/2) = 1 - p(1/2) = 0$, we then conclude that $g_2(x) \le 0$ for $x \in [\kappa, 1/2]$. □

The previous corollary and lemma then lead to the following theorem.

Theorem 1. *Let $\kappa = 0.46$ and let G be defined by (5) with κ_1 and κ_2 given in (8). Then G satisfies $1/\sin(\pi x) \le G(x)$ for $x \in (0, 1/2]$.*

The previous theorem gives the required lower bound on G. We now give the upper bound that we require.

Theorem 2. *Let G be defined by* (5). *Then $G(x) \leq 1/(2x)$ for $x \in (0, 1/2]$.*

Proof. Let $g_3(x) = G(x) - 1/(2x)$. Then for $x \in (0, \kappa]$ we have

$$g_3(x) = \frac{1}{\pi x} + \frac{\pi x}{6} + \frac{7\pi^3}{2880} - \frac{1}{2x} = \left(\frac{1}{\pi} - \frac{1}{2}\right)\frac{1}{x} + \frac{\pi x}{6} + \frac{7\pi^3}{2880}.$$

The derivative of g_3 is given by

$$g_3'(x) = -\left(\frac{1}{\pi} - \frac{1}{2}\right)\frac{1}{x^2} + \frac{\pi}{6}.$$

Since $1/\pi - 1/2 < 0$, we conclude that $g_3'(x)$ is positive on $(0, \kappa)$ and hence g_3 is an increasing function on this interval. A direct calculation shows that $g_3(\kappa) \approx -0.078761$. So for $x \in (0, \kappa]$ we have $g_3(x) \leq g_3(\kappa) < 0$, that is, $G(x) < 1/(2x)$ on this interval.

For $x \in (\kappa, 1/2]$, we have $g_3(x) = \kappa_1 + \kappa_2 x - 1/(2x)$. Then for $x \in (\kappa, 1/2)$, $g_3'(x)$ is given by

$$g_3'(x) = \kappa_2 + \frac{1}{2x^2} > \kappa_2 + \frac{1}{2(1/2)^2} = \kappa_2 + 2 \approx 1.795102 > 0.$$

Hence g_3 is an increasing function on $(\kappa, 1/2]$. By the construction of the linear function $p(x) = \kappa_1 + \kappa_2 x$, we had $p(1/2) = 1$. So $\kappa_1 + \kappa_2/2 = 1$ and hence $g_3(1/2) = 0$. This means that $G(x) \leq 1/(2x)$ for $x \in (\kappa, 1/2]$. So, overall, we conclude that $G(x) \leq 1/(2x)$ for $x \in (0, 1/2]$. □

For $h \in \mathbb{Z}$, let

$$w(h, n) = \begin{cases} n/G(|h|/n) & \text{for } h \in C^*(n), \\ 1 & \text{for } h = 0, \end{cases}$$

and for $\boldsymbol{h} \in \mathbb{Z}^d$, set

$$w(\boldsymbol{h}, n) = \prod_{i=1}^{d} w(h_i, n).$$

Since, by construction, we have $1/t(h, n) \leq 1/w(h, n) \leq 1/(2r(h)$ for $h \in C^*(n)$ while $1/t(0, n) = 1/w(0, n) = 1/r(0) = 1$, it follows that

$$\frac{1}{t(\boldsymbol{h}, n)} \leq \frac{1}{w(\boldsymbol{h}, n)} \leq \frac{1}{2r(\boldsymbol{h})} \text{ for } \boldsymbol{h} \in C_d^*(n).$$

Setting

$$W(z,n) = \sum_{\boldsymbol{h}} \frac{1}{w(\boldsymbol{h},n)},$$

where the sum is over all $\boldsymbol{h} \in C_d^*(n)$ satisfying $\boldsymbol{h} \cdot z \equiv 0 \mod n$, we then have (3) holding, that is, $T(z,n) \leq W(z,n) \leq R(z,n)/2$. This leads to the intermediate bound on the star discrepancy given by

$$D^*(P_n(z)) \leq 1 - (1 - 1/n)^d + W(z,n). \tag{9}$$

3 Calculating $W(z, n)$

From the error expression for lattice rules, one may write

$$\begin{aligned} R(z,n) &= -1 + \frac{1}{n}\sum_{k=0}^{n-1} \sum_{\boldsymbol{h}\in C_d(n)} \frac{e^{2\pi \mathrm{i} k \boldsymbol{h}\cdot z/n}}{r(\boldsymbol{h})} \\ &= -1 + \frac{1}{n}\sum_{k=0}^{n-1} \prod_{j=1}^{d} \left(1 + \sum_{h\in C^*(n)} \frac{e^{2\pi \mathrm{i} k h z_j/n}}{|h|}\right) \end{aligned} \tag{10}$$

and

$$\begin{aligned} W(z,n) &= -1 + \frac{1}{n}\sum_{k=0}^{n-1} \sum_{\boldsymbol{h}\in C_d(n)} \frac{1}{w(\boldsymbol{h},n)} e^{2\pi \mathrm{i} k \boldsymbol{h}\cdot z/n} \\ &= -1 + \frac{1}{n}\sum_{k=0}^{n-1} \prod_{j=1}^{d} \left(1 + \frac{1}{n}\sum_{h\in C^*(n)} G(|h|/n) e^{2\pi \mathrm{i} k h z_j/n}\right). \end{aligned} \tag{11}$$

Then we see that the calculation of $W(z,n)$ for a given z by using this last formula would require $O(n^2 d)$ operations.

This formula may be written as $W(z,n) = Q_{n,d}(f_n) - 1$, where

$$f_n(\boldsymbol{x}) = \prod_{i=1}^{d} F_n(x_i)$$

and

$$F_n(x) = 1 + \frac{1}{n}\sum_{h\in C^*(n)} G(|h|/n) e^{2\pi \mathrm{i} h x}, \qquad x \in [0,1).$$

With the notation

$$\eta(n) = \begin{cases} (n+1)/2 \text{ for } n \text{ odd,} \\ n/2, \qquad\quad \text{for } n \text{ even,} \end{cases}$$

and

$$S(x,\eta(n)) = \frac{1}{n}\sum_{h=1}^{\eta(n)-1} G(h/n)\cos(2\pi hx),$$

we have

$$F_n(x) = \begin{cases} 1 + 2S(x,\eta(n)) & \text{for } n \text{ odd,} \\ 1 + 2S(x,\eta(n)) + \dfrac{e^{\pi i n x}}{n} & \text{for } n \text{ even,} \end{cases}$$

where we have used $G(1/2) = 1$ in the case when n is even.

For $1 \le h \le \eta(n) - 1$, we have

$$\frac{G(h/n)}{n} = \begin{cases} 1/(\pi h) + \pi h/(6n^2) + 7\pi^3/(2880n) & \text{for } 0 < h/n \le \kappa, \\ \kappa_1/n + \kappa_2 h/n^2 & \text{for } \kappa < h/n \le 1/2. \end{cases}$$

Now let $\alpha(n) = \lfloor \kappa n \rfloor + 1$. Then it follows that $0 < h/n \le \kappa$ for $1 \le h \le \alpha(n) - 1$ and $\kappa < h/n \le 1/2$ for $\alpha(n) \le h \le \eta(n) - 1$. Hence

$$\begin{aligned} S(x,\eta(n)) = {} & \frac{1}{\pi}\sum_{h=1}^{\alpha(n)-1}\frac{\cos(2\pi hx)}{h} + \frac{\pi}{6n^2}\sum_{h=1}^{\alpha(n)-1} h\cos(2\pi hx) \\ & + \frac{7\pi^3}{2880n}\sum_{h=1}^{\alpha(n)-1}\cos(2\pi hx) \\ & + \sum_{h=\alpha(n)}^{\eta(n)-1}\left(\frac{\kappa_1}{n} + \frac{\kappa_2 h}{n^2}\right)\cos(2\pi hx). \end{aligned}$$

(This last sum is taken to be an empty sum of zero when n is odd and less than 13 or when n is even and less than 26 as then $\alpha(n) > \eta(n) - 1$.)

For integer $m \ge 2$ and $x \in (0,1)$, it follows from [4, p. 37] that

$$\sum_{h=1}^{m-1}\cos(2\pi hx) = \frac{\sin(m\pi x)\cos((m-1)\pi x)}{\sin(\pi x)} - 1 := \sigma_1(x,m).$$

For the case $x = 0$, we set $\sigma_1(0,m) = m - 1$. Moreover, [4, p. 38] yields for integer $m \ge 2$ and $x \in (0,1)$,

$$\sum_{h=1}^{m-1} h\cos(2\pi hx) = \frac{m\sin((2m-1)\pi x)}{2\sin(\pi x)} - \frac{1-\cos(2m\pi x)}{4\sin^2(\pi x)} := \sigma_2(x,m).$$

For the case $x = 0$, we set $\sigma_2(0,m) = (m-1)m/2$. We may then write

$$\begin{aligned} S(x,\eta(n)) = \frac{1}{\pi}\sum_{h=1}^{\alpha(n)-1} \frac{\cos(2\pi hx)}{h} &+ \frac{\pi}{6n^2}\sigma_2(x,\alpha(n)) + \frac{7\pi^3}{2880n}\sigma_1(x,\alpha(n)) \\ &+ \frac{\kappa_1}{n}\big[\sigma_1(x,\eta(n)) - \sigma_1(x,\alpha(n))\big] \\ &+ \frac{\kappa_2}{n^2}\big[\sigma_2(x,\eta(n)) - \sigma_2(x,\alpha(n))\big]. \end{aligned} \tag{12}$$

As all the components of the points in an n-point rank-1 lattice are of the form j/n for $0 \le j \le n-1$ and since $\cos(2\pi h(1-x)) = \cos(2\pi hx)$ for $x \in [0,1]$, we see that calculation of $W(z,n)$ requires the values of $F(j/n)$ and hence the values of $S(j/n,\eta(n))$ for j satisfying $0 \le j \le \lfloor n/2 \rfloor$.

It is clear from (12) that the time-consuming part of the calculation of $S(j/n,\eta(n))$ is in calculating the values

$$\sum_{h=1}^{\alpha(n)-1} \frac{\cos(2\pi hj/n)}{h}, \quad 0 \le j \le \lfloor n/2 \rfloor.$$

The Appendix gives details of how the results in [7] may be used to approximate all these values accurately enough in $O(n)$ operations so that the values of $F(j/n)$ have absolute error no more than some specified $\varepsilon > 0$. These $\lfloor n/2 \rfloor + 1$ values of $F(j/n)$ are then stored and allow, for a given z, $W(z,n)$ to be calculated to a fixed precision in $O(nd)$ operations.

4 Component-by-Component Construction

In this section, we show that for n prime we can construct z component-by-component (CBC) such that the bound given in (13) below holds. The result and proof is very similar to Theorem 1 and its proof found in [6].

Theorem 3. *Let n be a prime number and let $\mathscr{Z}_n = \{z : 1 \le z \le n-1\}$. Suppose there exists a $z \in \mathscr{Z}_n^d$ such that*

$$W(z,n) \le \frac{1}{n-1}\left(1+U_n\right)^d, \tag{13}$$

where

$$U_n = \frac{1}{n} \sum_{h \in C^*(n)} G(|h|/n).$$

Then there exists $z_{d+1} \in \mathscr{Z}_n$ *such that*

$$W((\boldsymbol{z}, z_{d+1}), n) \leq \frac{1}{n-1} \left(1 + U_n\right)^{d+1}.$$

Such a z_{d+1} *can be found by minimizing* $W((\boldsymbol{z}, z_{d+1}), n)$ *over the set* $\mathscr{Z}_n$.

Proof. For any $z_{d+1} \in \mathscr{Z}_n$, we have from (11) that $W((\boldsymbol{z}, z_{d+1}), n)$ may be expressed as

$$\begin{aligned} &W((\boldsymbol{z}, z_{d+1}), n) \\ &= W(\boldsymbol{z}, n) + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^{d} \left(1 + \frac{1}{n} \sum_{h \in C^*(n)} G(|h|/n) e^{2\pi \mathrm{i} k h z_j / n}\right) \\ &\quad \times \left(\frac{1}{n} \sum_{h \in C^*(n)} G(|h|/n) e^{2\pi \mathrm{i} k h z_{d+1} / n}\right). \end{aligned} \tag{14}$$

Next, we average over the possible $n-1$ values of z_{d+1} in the last term to form for $0 \leq k \leq n-1$,

$$\begin{aligned} V_n(k) &= \frac{1}{n-1} \sum_{z_{d+1}=1}^{n-1} \frac{1}{n} \sum_{h \in C^*(n)} G(|h|/n) e^{2\pi \mathrm{i} k h z_{d+1} / n} \\ &= \frac{1}{(n-1)n} \sum_{h \in C^*(n)} \sum_{z=1}^{n-1} G(|h|/n) e^{2\pi \mathrm{i} k h z / n} \\ &= \frac{1}{(n-1)n} \sum_{h \in C^*(n)} G(|h|/n) \left(\sum_{z=0}^{n-1} \left(e^{2\pi \mathrm{i} k h / n}\right)^z - 1\right). \end{aligned}$$

When $k = 0$, $V_n(0)$ is simply U_n. For $1 \leq k \leq n-1$ and $h \in C^*(n)$, it is clear that k and h are relatively prime with n. It then follows that $kh \not\equiv 0 \pmod{n}$ so that

$$\sum_{z=0}^{n-1} \left(e^{2\pi \mathrm{i} k h / n}\right)^z - 1 = -1.$$

Hence for $1 \leq k \leq n-1$, we have

$$V_n(k) = \frac{-U_n}{n-1}. \tag{15}$$

From the expression for $W((z, z_{d+1}), n)$ given in (14), it follows by separating out the $k = 0$ term that there exists a $z_{d+1} \in \mathscr{Z}_n$ such that

$$\begin{aligned} & W((z, z_{d+1}), n) \\ & \le W(z, n) + \frac{1}{n}(1 + U_n)^d U_n \\ & \qquad + \frac{1}{n}\sum_{k=1}^{n-1}\prod_{j=1}^{d}\left(1 + \frac{1}{n}\sum_{h \in C^*(n)} G(|h|/n) e^{2\pi i k h z_j / n}\right) V_n(k) \\ & = W(z, n) + \frac{1}{n}(1 + U_n)^d U_n \\ & \qquad + \frac{1}{n}\sum_{k=1}^{n-1}\prod_{j=1}^{d}\left(1 + \frac{1}{n}\sum_{h \in C^*(n)} G(|h|/n) e^{2\pi i k h z_j / n}\right)\left(\frac{-U_n}{n-1}\right), \end{aligned} \tag{16}$$

where we have made use of (15). By subtracting and adding in the $k = 0$ term, we see that the last term in (16) may be written as

$$\frac{U_n}{n-1}\left(-\frac{1}{n}\sum_{k=0}^{n-1}\prod_{j=1}^{d}\left(1 + \frac{1}{n}\sum_{h \in C^*(n)} G(|h|/n) e^{2\pi i k h z_j / n}\right) + \frac{(1 + U_n)^d}{n}\right).$$

Equation 11 shows that this last expression is simply

$$\frac{U_n}{n-1}\left(-W(z, n) - 1 + \frac{(1 + U_n)^d}{n}\right).$$

Hence it follows from (16) that there exists a $z_{d+1} \in \mathscr{Z}_n$ such that

$$\begin{aligned} & W((z, z_{d+1}), n) \\ & \le W(z, n) + \frac{1}{n}(1 + U_n)^d U_n + \frac{U_n}{n-1}\left(-W(z, n) - 1 + \frac{(1 + U_n)^d}{n}\right) \\ & \le W(z, n) + \frac{1}{n}(1 + U_n)^d U_n \left(1 + \frac{1}{n-1}\right) \\ & = W(z, n) + \frac{1}{n-1}(1 + U_n)^d U_n \\ & \le \frac{1}{n-1}(1 + U_n)^d + \frac{1}{n-1}(1 + U_n)^d U_n = \frac{1}{n-1}(1 + U_n)^{d+1}, \end{aligned}$$

where we have made use of the fact that $W(z, n)$ satisfies the assumed bound. This completes the proof. □

In the case when $d = 1$, we can set $z_1 = 1$. Then we have from (11) that

$$\begin{aligned} W(z_1, n) &= -1 + \frac{1}{n}\sum_{k=0}^{n-1}\left(1 + \frac{1}{n}\sum_{h \in C^*(n)} G(|h|/n)e^{2\pi \mathrm{i} kh/n}\right) \\ &= \frac{1}{n^2}\sum_{k=0}^{n-1}\sum_{h \in C^*(n)} G(|h|/n)e^{2\pi \mathrm{i} kh/n} \\ &= \frac{1}{n^2}\sum_{h \in C^*(n)} G(|h|/n)\sum_{k=0}^{n-1}\left(e^{2\pi \mathrm{i} h/n}\right)^k = 0. \end{aligned}$$

This result together with the previous theorem leads to the following corollary.

Corollary 2. *Let n be a prime number. We can construct $\boldsymbol{z} \in \mathscr{Z}_n^d$ component-by-component such that for all $s = 1, \ldots, d$,*

$$W((z_1, \ldots, z_s), n) \le \frac{1}{n-1}\left(1 + U_n\right)^s.$$

We can set $z_1 = 1$, and for $2 \le s \le d$, each z_s can be found by minimizing $W((z_1, \ldots, z_s), n)$ over the set $\mathscr{Z}_n$.

To obtain bounds on the star discrepancy resulting from the CBC construction based on $W(\boldsymbol{z}, n)$, we now consider

$$U_n = \frac{1}{n}\sum_{h \in C^*(n)} G(|h|/n)$$

in more detail. By construction, we have $G(|h|/n) \le 1/(2|h|/n)$ and hence

$$U_n \le \frac{1}{2}\sum_{h \in C^*(n)} \frac{1}{|h|} =: \frac{1}{2}S_n < \ln(n) + \gamma - \ln(2) + \frac{1}{2n^2}, \tag{17}$$

where $\gamma \approx 0.57722$ is Euler's constant and the last step follows from [8, Lemmas 1 and 2]. For n an odd prime, we have $1/n^2 \le 1/9$. Then the previous corollary, the intermediate bound from (9), and calculation of the constant in (17) show that the $\boldsymbol{z}$ from the CBC construction results in a point set for which

$$\begin{aligned} D^*(P_n(\boldsymbol{z})) &\le 1 - \left(1 - \frac{1}{n}\right)^d + \frac{(0.9397 + \ln(n))^d}{n-1} \\ &\le \frac{d}{n} + \frac{(0.9397 + \ln(n))^d}{n-1}. \end{aligned} \tag{18}$$

When $d \geq 2$ and $n \geq 3$, this bound is an improvement on the bound of

$$D^*(P_n(z)) \leq \frac{d}{n} + \frac{(0.8793 + 2\ln(n))^d}{2(n-1)} \tag{19}$$

found in [6, Sect. 3].

To get a bound on the star discrepancy better than the one in (18), we could work with U_n directly. However, the piecewise nature of G complicates the analysis. For simplicity, we shall work with the function $\widetilde{G}$ defined in (7).

Since a direct calculation shows that $7\pi^3/2880 < 1 - 2/\pi - \pi/12$, then we have $G(x) \leq \widetilde{G}(x)$ for $x \in (0, \kappa]$. For $x \in (\kappa, 1/2]$, the function $g_4(x) = \widetilde{G}(x) - (\kappa_1 + \kappa_2 x)$ has derivative given by

$$g_4'(x) = -\frac{1}{\pi x^2} + \frac{\pi}{6} - \kappa_2 \leq -\frac{1}{\pi(1/2)^2} + \frac{\pi}{6} - \kappa_2 \approx -0.544743 < 0.$$

So g_4 is a decreasing function on $(\kappa, 1/2)$ meaning that on this interval, $g_4(x) \geq g_4(1/2) = 0$. Hence we conclude overall that

$$G(x) \leq \widetilde{G}(x), \quad x \in (0, 1/2].$$

Though not needed here, the first part of the proof of Theorem 2 showing that $G(x) \leq 1/(2x)$ for $x \in (0, \kappa]$ may be modified to show that $\widetilde{G}(x) \leq 1/(2x)$ for $x \in (0, 1/2]$.

We then have

$$\begin{aligned} U_n = \frac{1}{n} \sum_{h \in C^*(n)} G(|h|/n) &\leq \frac{1}{n} \sum_{h \in C^*(n)} \widetilde{G}(|h|/n) \\ &= \frac{1}{n} \sum_{h \in C^*(n)} \left(\frac{n}{\pi|h|} + \frac{\pi|h|}{6n} + 1 - \frac{2}{\pi} - \frac{\pi}{12} \right) \\ &= \frac{S_n}{\pi} + \frac{\pi}{6n^2} \sum_{h \in C^*(n)} |h| + \frac{n-1}{n} \left(1 - \frac{2}{\pi} - \frac{\pi}{12} \right). \end{aligned}$$

In the case when n is odd, the sum in this last equation is simply $(n-1)(n+1)/4$. So for n odd, we obtain

$$\begin{aligned} U_n &\leq \frac{S_n}{\pi} + \frac{\pi(n^2-1)}{24n^2} + \frac{n-1}{n} \left(1 - \frac{2}{\pi} - \frac{\pi}{12} \right) \\ &\leq \frac{S_n}{\pi} + \frac{\pi}{24} + 1 - \frac{2}{\pi} - \frac{\pi}{12} \\ &\leq \frac{1}{\pi} \left(2\ln(n) + 2\gamma - \ln(4) + \frac{1}{n^2} \right) + 1 - \frac{2}{\pi} - \frac{\pi}{24}, \end{aligned}$$

where we have made use of (17). So for n an odd prime, we use the previous corollary, (9), $1/n^2 \leq 1/9$, and calculation of the constant in this last expression to conclude that the CBC construction leads to a $\boldsymbol{z}$ for which

$$D^*(P_n(\boldsymbol{z})) \leq \frac{d}{n} + \frac{(1.1941 + 2\ln(n)/\pi)^d}{n-1}. \tag{20}$$

Like the bounds given in (18) and (19), this bound shows that the CBC construction leads to a $\boldsymbol{z}$ for which $D^*(P_n(\boldsymbol{z})) = O(n^{-1}(\ln(n))^d)$. However, this bound has a smaller implied constant than in the two earlier bounds.

5 Numerical Results and Summary

Here we present numerical values of the three quantities $T(\boldsymbol{z}, n)$, $W(\boldsymbol{z}, n)$, and $R(\boldsymbol{z}, n)/2$ to see how they compare against each other. The $\boldsymbol{z}$ that were used in the calculations came from the CBC algorithm given in Corollary 2. We present results for $d = 2$, $d = 3$, $d = 10$, and $d = 20$. In the case when $d = 2$ and $d = 3$, we provide (when it was computationally feasible to do so) the values of $E(\boldsymbol{z}, n)$, where

$$E(\boldsymbol{z}, n) := D^*(P_n(\boldsymbol{z})) - \left[1 - (1 - 1/n)^d\right].$$

Then

$$E(\boldsymbol{z}, n) \leq T(\boldsymbol{z}, n) \leq W(\boldsymbol{z}, n) \leq \frac{1}{2} R(\boldsymbol{z}, n).$$

The calculation of the star discrepancy required for $E(\boldsymbol{z}, n)$ (Tables 1 and 2) was done using the formulas given in [1].

Also presented are upper bounds on $W(\boldsymbol{z}, n)$ that arise from Corollary 2 and (20), namely,

$$\beta_1(n, d) := \frac{(1 + U_n)^d}{n-1} \quad \text{and} \quad \beta_2(n, d) := \frac{(1.1941 + 2\ln(n)/\pi)^d}{n-1}.$$

Table 1 Results for $d = 2$.

n	$E(\boldsymbol{z}, n)$	$T(\boldsymbol{z}, n)$	$W(\boldsymbol{z}, n)$	$R(\boldsymbol{z}, n)/2$	$\beta_1(n, 2)$	$\beta_2(n, 2)$
157	5.92(−3)	5.10(−2)	5.32(−2)	2.18(−1)	1.21(−1)	1.25(−1)
313	3.88(−3)	3.14(−2)	3.27(−2)	1.36(−1)	7.36(−2)	7.55(−2)
619	2.33(−3)	1.93(−2)	2.00(−2)	8.44(−2)	4.42(−2)	4.52(−2)
1,249	1.30(−3)	1.15(−2)	1.18(−2)	5.06(−2)	2.58(−2)	2.63(−2)
2,503	6.66(−4)	6.70(−3)	6.92(−3)	2.99(−2)	1.49(−2)	1.52(−2)
5,003	3.86(−4)	3.87(−3)	3.98(−3)	1.73(−2)	8.59(−3)	8.75(−3)
10,007	2.15(−4)	2.23(−3)	2.29(−3)	1.00(−2)	4.89(−3)	4.98(−3)
20,011		1.27(−3)	1.30(−3)	5.76(−3)	2.77(−3)	2.81(−3)

Table 2 Results for $d = 3$.

n	$E(z,n)$	$T(z,n)$	$W(z,n)$	$R(z,n)/2$	$\beta_1(n,3)$	$\beta_2(n,3)$
157	1.54(−2)	3.86(−1)	4.03(−1)	3.79(0)	5.28(−1)	5.51(−1)
313	1.37(−2)	2.63(−1)	2.74(−1)	2.68(0)	3.53(−1)	3.66(−1)
619	9.34(−3)	1.75(−1)	1.81(−1)	1.85(0)	2.31(−1)	2.39(−1)
1,249		1.12(−1)	1.16(−1)	1.22(0)	1.46(−1)	1.51(−1)
2,503		7.05(−2)	7.28(−2)	7.95(−1)	9.14(−2)	9.41(−2)
5,003		4.37(−2)	4.50(−2)	5.02(−1)	5.63(−2)	5.79(−2)
10,007		2.68(−2)	2.75(−2)	3.16(−1)	3.42(−2)	3.51(−2)
20,011		1.62(−2)	1.66(−2)	1.94(−1)	2.06(−2)	2.11(−2)

Table 3 Results for $d = 10$.

n	$T(z,n)$	$W(z,n)$	$R(z,n)/2$	$\beta_1(n,10)$	$\beta_2(n,10)$
10,007	2.60(4)	2.81(4)	3.38(8)	2.81(4)	3.07(4)
20,011	2.41(4)	2.59(4)	3.40(8)	2.59(4)	2.81(4)
40,009	2.15(4)	2.31(4)	3.26(8)	2.31(4)	2.49(4)
80,021	1.86(4)	1.99(4)	3.01(8)	1.99(4)	2.14(4)
160,001	1.57(4)	1.67(4)	2.68(8)	1.67(4)	1.79(4)
320,009	1.29(4)	1.36(4)	2.31(8)	1.36(4)	1.45(4)

Table 4 Results for $d = 20$.

n	$T(z,n)$	$W(z,n)$	$R(z,n)/2$	$\beta_1(n,20)$	$\beta_2(n,20)$
10,007	6.79(12)	7.92(12)	2.29(21)	7.92(12)	9.41(12)
20,011	1.16(13)	1.34(13)	4.62(21)	1.34(13)	1.58(13)
40,009	1.86(13)	2.13(13)	8.52(21)	2.13(13)	2.48(13)
80,021	2.78(13)	3.16(13)	1.45(22)	3.16(13)	3.66(13)
160,001	3.93(13)	4.45(13)	2.29(22)	4.45(13)	5.10(13)
320,009	5.29(13)	5.94(13)	3.41(22)	5.94(13)	6.77(13)

Obviously, the upper bounds on $W(z,n)$ from $\beta_1(n,d)$ will be better than the ones from $\beta_2(n,d)$. The numerical results suggest that the values of $\beta_2(n,d)$ still provide reasonable bounds. For given d and n, this quantity requires $O(1)$ operations to calculate compared to $O(n)$ operations for $\beta_1(n,d)$.

For $d = 10$ and $d = 20$, the results in Tables 3 and 4 show that the quantities $W(z,n)$ and $\beta_1(n,d)$ are close, though not equal.

From these numerical results and the work described in the previous sections, we summarize this paper as follows:

1. A quantity $W(z,n)$ has been introduced which leads to an intermediate bound on the star discrepancy.
2. The values of $W(z,n)$ are closer to $T(z,n)$ than to $R(z,n)/2$.
3. Even for moderate dimensions, the values of $W(z,n)$ are magnitudes of order smaller than $R(z,n)/2$. Nevertheless, since the star discrepancy is less than one, there is a large gap between the true values and the $O(n^{-1}(\ln(n))^d)$ bounds on the star discrepancy obtained from $W(z,n)$.

4. For a given z, $W(z,n)$ may be calculated to a fixed precision in $O(nd)$ operations. The author was not able to reduce the $O(n^2 d)$ operations required to calculate $T(z,n)$ to $O(nd)$ operations.
5. A CBC construction of z based on $W(z,n)$ has been analyzed and an $O(n^{-1}(\ln(n))^d)$ bound on the star discrepancy obtained with a smaller implied constant than the bound found in [6].

Appendix: Calculation of $F_n(x)$

We recall from Sect. 3 that the calculation of $W(z,n)$ requires the values of $F_n(j/n)$ for j satisfying $0 \le j \le \lfloor n/2 \rfloor$, where

$$F_n(x) = \begin{cases} 1 + 2S(x,\eta(n)) & \text{for } n \text{ odd}, \\ 1 + 2S(x,\eta(n)) + \dfrac{e^{\pi i n x}}{n} & \text{for } n \text{ even}, \end{cases}$$

with $S(x,\eta(n))$ given by (12). This last equation shows that with $\alpha(n) = \lfloor \kappa n \rfloor + 1 = \lfloor 0.46n \rfloor + 1$, we need the values

$$Y(j,\alpha(n)) := \sum_{h=1}^{\alpha(n)-1} \frac{\cos(2\pi hj/n)}{h}, \; 0 \le j \le \lfloor n/2 \rfloor. \tag{21}$$

Now suppose we want approximations to the values $F_n(j/n)$, $0 \le j \le \lfloor n/2 \rfloor$, such that they have absolute error no more than ε and that they may be calculated in $O(n)$ operations. This may be done by making use of the results in [7]. In particular, to apply those results here, the parameter $\eta(N)$ in that paper should be taken to be $\alpha(n)$. Moreover, (3.4) in Theorem 4 of that paper given by

$$\frac{4(T+1)!}{(\gamma-1)^{T+2}\pi^{T+2}} \le \varepsilon$$

should be replaced by

$$\frac{4(L+1)!}{(2\kappa)^{L+2}(\ell-1)^{L+2}\pi^{L+3}} \le \varepsilon,$$

where we have used ℓ and L here instead of γ and T, respectively, to avoid confusion with the notation used earlier. (This change in (3.4) of [7] arises because the proof of Theorem 4 there makes use of $\eta(N) \ge N/2 = 0.5N$ while here the corresponding inequality is $\alpha(n) > \kappa n = 0.46n$. Moreover, the values $F_n(j/n)$ here require $Y(j,\alpha(n))/\pi$.)

With the changes described in the previous paragraph, the results in [7] show that if ℓ and L are positive integers satisfying

$$2 \leq \ell \leq \left(\frac{6n^2}{\pi^2}\right)^{1/3} \quad \text{and} \quad \frac{4(L+1)!}{(2\kappa)^{L+2}(\ell-1)^{L+2}\pi^{L+3}} \leq \varepsilon, \tag{22}$$

then to approximate $F(j/n)$ to the required accuracy, $Y(j,\alpha(n))$ should be calculated directly using (21) for $0 \leq j < \ell$. When $\ell \leq j \leq \lfloor n/2 \rfloor$, $Y(j,\alpha(n))$ should be approximated by $K(j/n)$, where

$$K(x) = -\ln(2|\sin(\pi x)|) - \sum_{i=0}^{L} b_i(x)\cos(\pi[(2\alpha(n)+i-1)x + (i+1)/2]).$$

In this expression, $b_0(x) = 1/\ (2\alpha(n)|\sin(\pi x)|)$ and

$$b_{i+1}(x) = \frac{-(i+1)}{2(\alpha(n)+i+1)|\sin(\pi x)|} b_i(x).$$

As an example of a possible choice for ℓ, the first equation in (22) is satisfied with $\ell = 20$ when $n \geq 115$. Then the second equation in (22) is satisfied for $\varepsilon = 10^{-16}$ when $L = 15$. If $\varepsilon = 10^{-18}$, then we can take $L = 19$. So we see that approximations to all the values $F(j/n)$, $0 \leq j \leq \lfloor n/2 \rfloor$, may be obtained with an absolute error of at most ε using $O(\ell n) + O(L) \times (\lfloor n/2 \rfloor + 1 - \ell) = O(n)$ operations. This means that even if n is large, $W(z,n)$ may be calculated to a fixed precision in $O(nd)$ operations.

Acknowledgements This work was carried out when the author was a Visiting Fellow in the School of Mathematics and Statistics at the University of New South Wales. The author acknowledges the hospitality received and gives particular thanks to Dr Frances Kuo and Professor Ian Sloan. The author also thanks Dr Vasile Sinescu for his useful comments on an earlier version of this paper.

References

1. Bundschuh, P., Zhu, Y.: A method for exact calculation of the discrepancy of low-dimensional finite point sets I. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg **63**, 115–133 (1993)
2. Dick, J., Pillichshammer, F.: Digital Nets and Sequences. Cambridge University Press, New York (2010)
3. Gnewuch, M., Srivastav, A., Winzen, C.: Finding optimal subintervals with k points and calculating the star discrepancy are NP-hard problems. Journal of Complexity **25**, 115–127 (2009)
4. Gradshteyn, I.S., Ryzhik, I.M., Jeffrey, A., Zwillinger, D.: Tables of Integrals, Series and Products (7th edition). Academic Press, San Diego (2007)

5. Hlawka, E.: Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. Annali di Matematica Pura ed Applicata **54**, 325–333 (1961)
6. Joe, S.: Component by component construction of rank-1 lattice rules having $O(n^{-1}(\ln(n))^d)$ star discrepancy. In: Niederreiter, H. (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2002, pp. 293–298. Springer, Berlin (2004)
7. Joe, S., Sloan, I.H.: On computing the lattice rule criterion R. Mathematics of Computation **59**, 557–568 (1992)
8. Niederreiter, H.: Existence of good lattice points in the sense of Hlawka. Monatshefte für Mathematik **86**, 203–219 (1978)
9. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
10. Sinescu, V., L'Ecuyer, P.: On the behavior of the weighted star discrepancy bounds for shifted lattice rules. In: L'Ecuyer, P., Owen, A.B. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2008, pp. 603–616. Springer, Berlin (2009)
11. Zaremba, S.K.: Some applications of multidimensional integration by parts. Annales Polonici Mathematici **21**, 85–96 (1968)

On Monte Carlo and Quasi-Monte Carlo Methods for Series Representation of Infinitely Divisible Laws

Reiichiro Kawai and Junichi Imai

Abstract Infinitely divisible random vectors and Lévy processes without Gaussian component admit representations with shot noise series. To enhance efficiency of the series representation in Monte Carlo simulations, we discuss variance reduction methods, such as stratified sampling, control variates and importance sampling, applied to exponential interarrival times forming the shot noise series. We also investigate the applicability of the generalized linear transformation method in the quasi-Monte Carlo framework to random elements of the series representation. Although implementation of the proposed techniques requires a small amount of initial work, the techniques have the potential to yield substantial improvements in estimator efficiency, as the plain use of the series representation in those frameworks is often expensive. Numerical results are provided to illustrate the effectiveness of our approaches.

1 Introduction

An infinitely divisible random vector without Gaussian component admits representations of shot noise series. Such series representations have played an important role in theories such as the tail probability of stable laws and have also been studied in the applied literature, known as "shot noise." Series representation provides perfect and often easy simulation of infinitely divisible laws and associated Lévy processes. Series representations involving Poisson arrival times are given

R. Kawai
Department of Mathematics, University of Leicester, Leicester LE1 7RH, United Kingdom
e-mail: reiichiro.kawai@le.ac.uk; http://sites.google.com/site/reiichirokawai/

J. Imai
Faculty of Science and Technology, Keio University, Yokohama, 223–8522, Japan
e-mail: jimai@ae.keio.ac.jp

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_26, © Springer-Verlag Berlin Heidelberg 2012

for the first time by Ferguson and Klass [5] for real independent increment processes without Gaussian component and with positive jumps. The theory of stable processes and their applications are expanded, due to LePage [19] on series representation of stable random vectors. The simulation of nonnegative infinitely divisible random variables is considered and their series representations as a special form of generalized shot noise is developed in Bondesson [3]. The same approach is used in Rosiński [24] as a general pattern for series representations of Banach space valued infinitely divisible random vectors.

A disadvantage in simulation comes from the fact that the series representation for infinite Lévy measure is necessarily infinite as well. If the series converges at an extremely slow rate, a huge number of terms will be required to achieve a desired accuracy of the approximation. (We refer the reader to [11, 21] for examples of simulation use.) With ever increasing computational speed, however, a slow convergence may no longer cause a serious practical issue. The authors have recently achieved in [6–8] various improvements in implementation of the series representation. In this paper, we discuss some other possibilities of improvement in Monte Carlo and quasi-Monte Carlo methods. Although implementation of the proposed techniques requires a small amount of initial work, the techniques have the potential to yield substantial improvements in estimator efficiency, in particular as the plain use of the series representation in those frameworks is often expensive. In order to illustrate the effectiveness of the proposed techniques, we provide some numerical results for test examples (and do not present an exhaustive numerical study to avoid overloading the paper).

Let us begin with generalities on the series representation of infinitely divisible laws and Lévy processes. Consider a Lévy process $\{X_t : t \geq 0\}$ in $\mathbb{R}^d$, without Gaussian component, that is, its characteristic function is given by

$$\mathbb{E}\left[e^{i\langle y,X_t\rangle}\right] = \exp\left[t\left(i\langle y,\gamma\rangle + \int_{\mathbb{R}_0^d}\left(e^{i\langle y,z\rangle} - 1 - i\langle y,z\rangle \quad {}_{(0,1]}(\|z\|)\right)\nu(dz)\right)\right], \tag{1}$$

where $\gamma \in \mathbb{R}^d$ and ν is a Lévy measure on $\mathbb{R}_0^d (:= \mathbb{R}^d \setminus \{0\})$, that is, a σ-finite measure satisfying $\int_{\mathbb{R}_0^d}(\|z\|^2 \wedge 1)\nu(dz) < +\infty$. Let us start with construction of the series representation, based on the simulation of an inhomogeneous Poisson process. (See Asmussen and Glynn [1].) For the sake of simplicity, we restrict to the unilateral and univariate marginal (at unit time), that is, the infinitely divisible distribution on $\mathbb{R}_+$, rather than the multivariate Lévy process in $\mathbb{R}^d$. Denote by $\{\Gamma_k\}_{k\in\mathbb{N}}$ arrival times of a standard Poisson process, and let $\{E_k\}_{k\in\mathbb{N}}$ be a sequence of iid exponential random variables with unit mean. Notice first that the random variable $\sum_{k=1}^{+\infty}\Gamma_k \;\; (\Gamma_k \in [0,T])$ is infinitely divisible with Lévy measure $\nu(dz) = dz$ defined on $(0,T]$. Recall also that the epochs of an inhomogeneous Poisson process on $[0,T]$ with intensity $h(t)$ can be generated by $H(\Gamma_1)$, $H(\Gamma_2)$, $\ldots$, where $H(t) := \inf\{u \in [0,T] : \int_0^u h(s)ds > t\}$, provided that $\int_0^T h(s)ds < +\infty$. Therefore, by regarding the intensity $h(t)$ as a Lévy measure ("on state space" rather than on time), we deduce that $\sum_{k=1}^{+\infty} H(\Gamma_k) \;\; (\Gamma_k \in [0,T])$ is an infinitely divisible random variable with Lévy measure $\nu(dz) = h(z)dz$ defined on $(0, T]$. The definition of $H(t)$ implicitly assumes that the Lévy measure ν has a compact support. Moreover, the condition $\int_0^T h(s)ds < +\infty$ implies that Lévy measure is finite. The above argument can be extended to an infinite Lévy measure on $\mathbb{R}_+$, simply by redefining the inverse function H as running down from the infinity rather than up the other way, that is,

$$H(r) := \inf\left\{u > 0 : \int_u^{+\infty} h(s)ds > r\right\},$$

and compute $\sum_{k=1}^{+\infty} H(\Gamma_k)$, where $\{\Gamma_k\}_{k\in\mathbb{N}}$ is no longer restricted to lie in $[0, T]$. This formulation is the so-called inverse Lévy measure method [5, 19].

In most cases, however, the above tail inverse $H(r)$ of the Lévy measure is not available in closed form even in the one-dimensional setting. (See [7] for a numerical approach to the inverse Lévy measure method.) To obtain a closed form in general multi-dimensional settings, some alternative methods have been proposed, for example, the thinning method and the rejection method of [24], while each of those methods can be considered as a special case of the so-called generalized shot noise method of [3, 24], which we describe in brief as follows. Suppose that the Lévy measure ν in (1) can be decomposed as

$$\nu(B) = \int_0^{+\infty} \mathbb{P}\left(H(r, U) \in B\right) dr, \quad B \in \mathscr{B}(\mathbb{R}_0^d), \tag{2}$$

where U is a random variable taking values in a suitable space $\mathscr{U}$, and where $H : \mathbb{R}_+ \times \mathscr{U} \mapsto \mathbb{R}_0^d$ here is such that for each $u \in \mathscr{U}$, $r \mapsto \|H(r, u)\|$ is non-increasing. Then, the Lévy process $\{X_t : t \in [0, 1]\}$ in (1) admits the shot noise series representation

$$\{X_t : t \in [0, 1]\} \stackrel{\mathscr{L}}{=} \left\{t\gamma + \sum_{k=1}^{+\infty} \left[H\left(\Gamma_k, U_k\right) \quad (T_k \in [0, t]) - tc_k\right] : t \in [0, 1]\right\}, \tag{3}$$

where $\{U_k\}_{k\in\mathbb{N}}$ is a sequence of iid copies of the random variable U, $\{T_k\}_{k\in\mathbb{N}}$ is a sequence of iid uniform random variables on $[0, 1]$, and $\{c_k\}_{k\in\mathbb{N}}$ is a sequence of constants defined by $c_k := \mathbb{E}[H(\Gamma_k, U) \quad (\|H(\Gamma_k, U)\| \leq 1)]$. The random sequences $\{\Gamma_k\}_{k\in\mathbb{N}}$, $\{U_k\}_{k\in\mathbb{N}}$ and $\{T_k\}_{k\in\mathbb{N}}$ are mutually independent. Here, regardless of the structure of the function H, the common key building block is the epochs $\{\Gamma_k\}_{k\in\mathbb{N}}$ of a standard Poisson process. They can be generated iteratively as a successive summation of iid exponential random variables

$$\{\Gamma_1, \Gamma_2, \Gamma_3, \ldots\} \stackrel{\mathscr{L}}{=} \left\{\sum_{k=1}^{1} E_k, \sum_{k=1}^{2} E_k, \sum_{k=1}^{3} E_k, \ldots\right\}, \tag{4}$$

where the exponential random variables $\{E_k\}_{k\in\mathbb{N}}$ act as interarrival times of a standard Poisson process and can be generated through the standard inversion method, namely, $E_k \leftarrow -\ln(1-J_k)$ (or $-\ln J_k$, identically in law), where $\{J_k\}_{k\in\mathbb{N}}$ is a sequence of iid uniform random variables on $[0,1]$. It is worth emphasizing here that the argument r in $H(r,u)$ in (3), corresponding to the sequence $\{\Gamma_k\}_{k\in\mathbb{N}}$, is univariate, no matter what dimension the Lévy process $\{X_t : t \in [0,1]\}$ is defined in.

2 Variance Reduction Methods to Exponential Interarrival Times

In this section, we discuss variance reduction methods applied to exponential interarrival times $\{E_k\}_{k\in\mathbb{N}}$ in (4). To illustrate our methods, suppose we are interested in estimation of the η-th moment of an one-sided stable random variable, that is,

$$F := \left[\sum_{k=1}^{+\infty}\left(\alpha\sum_{l=1}^{k}E_l\right)^{-1/\alpha}\right]^{\eta} \stackrel{\mathscr{L}}{=} \left[\sum_{k=1}^{+\infty}(\alpha\Gamma_k)^{-1/\alpha}\right]^{\eta}, \tag{5}$$

for $\alpha \in (0,1)$ and $\eta \in (-\infty,\alpha)$. This is a simple yet very good example for our purpose, as the moment of arbitrary order is known in closed form;

$$\mathbb{E}_{\mathbb{P}}[F] = \left(\frac{\Gamma(1-\alpha)}{\alpha}\right)^{\eta/\alpha}\frac{\Gamma(1-\eta/\alpha)}{\Gamma(1-\eta)}, \quad \eta \in (-\infty,\alpha).$$

(See Examples 25.10 and 24.12 of Sato [25].) To guarantee that the variance $\mathrm{Var}_{\mathbb{P}}(F)$ is well defined, we need to impose $\eta \in (-\infty,\alpha/2)$. Throughout this section, we truncate the infinite sum to 100 terms, with which we have confirmed a sufficient convergence of the series.

We first consider stratified sampling. For simplicity, we apply the method only to the first (inter)arrival exponential time E_1. We divide the support $(0,+\infty)$ of the standard exponential distribution into M disjoint strata $\{B_m\}_{m\in\mathbb{M}}$, where $\mathbb{M} := \{1,\ldots,M\}$ and $B_1 = (0,b_1]$, $B_2 = (b_1,b_2],\ldots, B_M = (b_{M-1},+\infty)$ for $0 < b_1 < b_2 < \cdots$ in such a way that all the strata have the equal probability $p_m := \mathbb{P}(E_1 \in B_m) = 1/M$, for $m \in \mathbb{M}$. (We will use the general notation p_m below, while they are independent of m in our setting.) Define the stratum mean $\mu_m := \mathbb{E}_{\mathbb{P}}[F|E_1 \in B_m]$ and the stratum variance $\sigma_m^2 := \mathrm{Var}_{\mathbb{P}}(F|E_1 \in B_m)$. For each stratum m, let $\{F_{m,k}\}_{k\in\mathbb{N}}$ be a sequence of iid random variables such that each $F_{m,k}$ has the distribution of F conditional on the event $\{G \in B_m\}$, and let $(n_1,\ldots,n_M)'$ be the number of samples allocated to strata such that $n_m \geq 1$ and $\sum_{m\in\mathbb{M}} n_m = n$. Then, the random variable

$$\sum_{m\in\mathbb{M}} p_m \frac{1}{n_m}\sum_{k=1}^{n_m} F_{m,k}$$

is an unbiased estimator of $\mathbb{E}[F]$. Its variance is given by

$$\mathrm{Var}_{\mathbb{P}}\left(\sum_{m\in\mathbb{M}} p_m \frac{1}{n_m}\sum_{k=1}^{n_m} F_{m,k}\right) = \sum_{m\in\mathbb{M}} p_m^2 \frac{\alpha_m^2}{n_m} = \frac{1}{n}\sum_{m\in\mathbb{M}} p_m^2 \frac{\alpha_m^2}{q_m},$$

where $q_m := n_m/n$ indicates the fraction of observations drawn from the stratum m. This Monte Carlo variance is controllable through the allocation ratio $\{q_m\}_{m\in\mathbb{M}}$. For example, the proportional allocation, that is $q_m = p_m$, yields the variance

$$\sum_{m\in\mathbb{M}} p_m \sigma_m^2, \tag{6}$$

which, by the Jensen inequality, is smaller than, or at most equal to, the variance of the plain Monte Carlo method ($M = 1$),

$$\mathrm{Var}_{\mathbb{P}}(F) = \sum_{m\in\mathbb{M}} p_m\sigma_m^2 + \sum_{m\in\mathbb{M}} p_m\mu_m^2 - \left(\sum_{m\in\mathbb{M}} p_m\mu_m\right)^2. \tag{7}$$

Moreover, the allocation $q_m = p_m\sigma_m/(\sum_{m\in\mathbb{M}} p_m\sigma_m)$ achieves the minimal variance

$$\left(\sum_{m\in\mathbb{M}} p_m\sigma_m\right)^2, \tag{8}$$

which is further smaller than, or at most equal to, the variance (6), again due to the Jensen inequality. We report in Table 1 variance ratios achieved through stratified sampling for $\alpha = \{0.3, 0.5, 0.7\}$ and $\eta = -0.2$. The displayed quantities (vratio1) and (vratio2) indicate ratio of variances "(7)/(6)" and "(7)/(8)", respectively. The observed reductions in variance are remarkable. This fact confirms the importance of the first (inter)arrival exponential time E_1 in series representations of infinitely divisible laws for estimation purpose. Further reduction in variance through the optimal allocation $q_m = p_m\sigma_m/(\sum_{m\in\mathbb{M}} p_m\sigma_m)$ varies for different settings. Note that this further improvement requires a pilot run for estimation of the stratum variance σ_m^2 to find the optimal allocation, while stratified sampling with the proportional allocation $q_m = p_m$ is free of a pilot run.

Before proceeding to different variance reduction methods, let us remark that the method of stratified sampling can be in principle multi-dimensional, that is, Latin hypercube sampling. In our context, the method is also applied to exponential times E_2 and so on, not only to E_1. This extension is however usually not computationally effective. First, as discussed in [6], a first few exponential times often account for most of variation. Moreover, in a d-dimensional Latin hypercube sampling problem, the total number of strata increases to the product $M_1 \cdots M_d$, where M_k denotes the number of strata of the k-th coordinate. Note that for successful variance reduction, each M_k must be fairly large. (We discuss the quasi-Monte Carlo method that are

Table 1 Variance ratios achieved through stratified sampling for $\eta = -0.2$.

α	M	2	5	10	50	200
0.3	vratio1	2.7	9.1	20.1	63.1	83.9
	vratio2	3.0	13.3	35.9	117.7	143.6
0.5	vratio1	2.9	10.0	20.5	39.6	42.7
	vratio2	2.9	11.0	23.0	46.3	51.6
0.7	vratio1	2.4	6.8	12.4	23.2	25.0
	vratio2	2.6	8.0	14.1	24.4	26.8

better suited to higher dimensional problems in Sect. 3. See, for example, Owen [22] for details.)

Let us turn to variance reduction methods of control variates and importance sampling applied again to moment estimation of the one-sided stable random variable (5). We begin with some notations. Fix $n \in \mathbb{N}$ and define $E^{(n)} = [E_1, \ldots, E_n]^\top$ and $\lambda := [\lambda_1, \ldots, \lambda_n]^\top \in (-\infty, +1)^n$. Here, we add a parametrization to the first n exponential random variables $\{E_k\}_{k=1,\ldots,n}$ in (5) as $\{E_k/(1-\lambda_k)\}_{k=1,\ldots,n}$, and also parametrize the random variable F of interest as $F(\lambda)$ accordingly. Clearly, $F(0)$ reduces to the original form (5). Define a family $\{\mathbb{Q}_\lambda\}_{\lambda\in(-\infty,+1)^n}$ of probability measures by

$$\frac{d\mathbb{Q}_\lambda}{d\mathbb{P}}\Big|_{\sigma(E^{(n)})} := \frac{e^{\langle\lambda, E^{(n)}\rangle}}{\mathbb{E}_{\mathbb{P}}[e^{\langle\lambda, E^{(n)}\rangle}]} = \prod_{k=1}^{n} \frac{e^{\lambda_k E_k}}{\mathbb{E}_{\mathbb{P}}[e^{\lambda_k E_k}]} = \prod_{k=1}^{n} (1-\lambda_k)\, e^{\lambda_k E_k}, \quad \mathbb{P}\text{-}a.s.$$

It holds that

$$\frac{d\mathbb{P}}{d\mathbb{Q}_\lambda}\Big|_{\sigma(E^{(n)})} = \left(\frac{d\mathbb{Q}_\lambda}{d\mathbb{P}}\Big|_{\sigma(E^{(n)})}\right)^{-1} = \prod_{k=1}^{n} \frac{e^{-\lambda_k E_k}}{1-\lambda_k}, \quad \mathbb{Q}_\lambda\text{-}a.s.$$

We can show that under $\mathbb{Q}_\lambda$, $\{E_k\}_{k\in\mathbb{N}}$ is a sequence of independent (not necessarily identically distributed) exponential random variables. Note also that $\mathbb{Q}_\lambda(E_k \in B) = \mathbb{P}(E_k/(1-\lambda_k) \in B)$ and $\mathbb{Q}_\lambda(F(0) \in B) = \mathbb{P}(F(\lambda) \in B)$ for $B \in \mathscr{B}(\mathbb{R})$. We are now in a position to formulate variance reduction methods as; for each $\lambda \in (-\infty, +1)^n$ and $\theta := [\theta_1, \ldots, \theta_n]^\top \in \mathbb{R}^n$,

$$\mathbb{E}_{\mathbb{P}}[F(0)] = \mathbb{E}_{\mathbb{P}}\left[F(0) - \left\langle\theta, E^{(n)} - \mathbb{E}_{\mathbb{P}}\left[E^{(n)}\right]\right\rangle\right] \tag{9}$$

$$= \mathbb{E}_{\mathbb{Q}_\lambda}\left[\frac{d\mathbb{P}}{d\mathbb{Q}_\lambda}\Big|_{\sigma(E^{(n)})}\left(F(0) - \left\langle\theta, E^{(n)} - \mathbb{E}_{\mathbb{P}}\left[E^{(n)}\right]\right\rangle\right)\right] \tag{10}$$

$$= \mathbb{E}_{\mathbb{P}}\left[\left(\prod_{k=1}^{n} \frac{e^{-\frac{\lambda_k}{1-\lambda_k}E_k}}{1-\lambda_k}\right)\left(F(\lambda) - \sum_{k=1}^{n}\theta_k\left(\frac{E_k}{1-\lambda_k} - 1\right)\right)\right]. \tag{11}$$

The subtraction term inside the expectation (9) corresponds to the method of control variates, while the change of measure in (10) acts as the method of

Table 2 Variance ratio (vratio) for $\eta = -0.2$ when either control variates alone $(\theta^*, 0)$ or importance sampling alone $(0, \lambda^*)$ is applied.

α	0.3		0.5		0.7	
(θ, λ)	(0.223, 0)	(0, 0.368)	(0.180, 0)	(0, 0.232)	(0.117, 0)	(0, 0.147)
vratio	12.7	3.5	4.0	2.5	2.6	2.1

importance sampling. The equality (11) holds by the so-called scaling property of the exponential (more generally, gamma) distribution. (See [13, 16] for its applications.) Within this framework, it is most ideal to find the joint parameter (λ, θ) minimizing the variance

$$V(\lambda, \theta) := \mathrm{Var}_{\mathbb{Q}_\lambda}\left(\frac{d\mathbb{P}}{d\mathbb{Q}_\lambda}|_{\sigma(E^{(n)})}\left(F(0) - \left\langle\theta, E^{(n)} - \mathbb{E}_{\mathbb{P}}\left[E^{(n)}\right]\right\rangle\right)\right)$$

$$= \mathbb{E}_{\mathbb{P}}\left[\left(\prod_{k=1}^{n}\frac{e^{-\lambda_k E_k}}{1-\lambda_k}\right)\left(F(0) - \left\langle\theta, E^{(n)} - \quad {}_n\right\rangle\right)^2\right] - \left(\mathbb{E}_{\mathbb{P}}\left[F(0)\right]\right)^2,$$

where ${}_n := [1, \dots, 1]^\top \in \mathbb{R}^n$. We however do not discuss this joint framework, as it entails various complex techniques, such as adaptive variance reduction and stochastic approximation. (See, for example, [12, 13] for details.) For the sake of simplicity, we instead apply either control variates alone ($\lambda = 0$) or importance sampling alone ($\theta = 0$) to the first arrival time E_1, that is, $n = 1$ in (10). In particular, as the function $V(\lambda, \theta)$ is quadratic in θ, the optimal parameter θ^* for control variates (with $\lambda = 0$) can be easily derived as

$$\theta^* = \mathrm{Cov}_{\mathbb{P}}\left(F(0),\ E^{(n)}\right),$$

which is to be estimated through a pilot run. It is known that there exists a unique λ^* which attains the global minimum of the function $V(0, \lambda)$, while searching λ^* is not a trivial problem and requires some techniques such as stochastic approximation algorithm. (See [14] for details.) In order to illustrate the effectiveness of the above variance reduction methods, we present in Table 2 optimal variance ratios for the same parameter set as in Table 1. The variance ratio should be read as either $V(0,0)/V(\theta^*, 0)$ or $V(0,0)/V(0, \lambda^*)$. The formulation (11) has achieved a reduction in variance by roughly factors between 2 and 13. (In this specific example, unfortunately, it is not quite competitive against the aforementioned method of stratified sampling.) It is worth emphasizing that as can be seen in (11), it requires little amount of initial work to apply the variance reduction methods. There remains an issue of pilot run for finding optimal parameter θ^* or λ^*. Taking into account the gained variance ratios, a pilot run seems worthwhile, as Monte Carlo methods with series representation is computationally demanding by nature even with ever increasing computing speed.

We close this section with discussion on some extensions. First, as mentioned in the introductory, the exponential interarrival times $\{E_k\}_{k\in\mathbb{N}}$ are univariate, regardless of dimension. Hence, the proposed techniques on $\{E_k\}_{k\in\mathbb{N}}$ are directly applicable to general multi-dimensional setting. It might also be beneficial

1. To apply variance reduction methods to the further interarrival times $\{E_k\}_{k=2,\dots}$,
2. To combine the discussed variance reduction methods, such as [13, 15],
3. To employ yet different variance reduction methods,
4. To apply variance reduction methods to the random element $\{U_k\}_{k\in\mathbb{N}}$ in (3),

to mention just a few. Such different approaches would certainly entail different types of initial work and yield different improvements in estimator efficiency. In any case, we will need to look closely at both gained efficiency and additional computing effort required for the pilot run and implementation.

3 Generalized Linear Transformation

As we mentioned in the previous section, the stratified sampling is not effective in applying high-dimensional problems due to a rapid growth of strata as the Monte Carlo dimension increases. We examine in this section the effectiveness of quasi-Monte Carlo (QMC) method on the series representation. As mentioned in Sect. 2, latin hypercube sampling may avoid increasing the number of strata, while we do not compare this method as the QMC method is known to have superior properties in dealing with multiple dimensions.

By relying on a specially constructed sequence known as the low discrepancy sequence, QMC achieves a convergence rate of $O(N^{-1}\log^d N)$, in dimension d and sample size N. Asymptotically, this rate is far more superior than that of the classical MC and Latin hypercube sampling. Furthermore, it is known that the success of QMC is intricately related to the notion of effective dimension. Therefore, in practical applications, the superior rate of QMC could be attained when the effective dimension of the integrand is small, even if its nominal dimension is of several hundreds or thousands. This suggests that an effective way of enhancing the efficiency of QMC is performed via dimension reduction techniques. In this section, we investigate one of the dimension reduction techniques, called the generalized linear transformation (GLT, for short) method, proposed in Imai and Tan [10], to enhance the QMC method applied to the series representations.

Let $g:\mathbb{R}^d\to\mathbb{R}$ and let $X:=[X_1,\dots,X_d]^\top$ be a random vector in $\mathbb{R}^d$ with independent components where each X_k has the (univariate) law F_k on $\mathbb{R}$. We assume that for every $k=1,\dots,d$, the inverse F_k^{-1} is well defined and each law F_k admits a probability density function f_k. We are concerned with evaluation of the expectation

$$\mathbb{E}\left[g\left(X\right)\right]=\int_{\mathbb{R}^d} g(x)f_1\left(x_1\right)\cdots f_d\left(x_d\right)dx=:\mathscr{I}_d\left(\{F_k\};g\right), \tag{12}$$

where $x := [x_1, \ldots, x_d]^\top$. Using standard transformation, this can be reformulated as

$$\begin{aligned}\mathscr{I}_d\left(\{F_k\}; g\right) &= \int_{[0,1]^d} g\left(F_1^{-1}\left(u_1\right), \ldots, F_d^{-1}\left(u_d\right)\right) du, \\ &= \mathbb{E}\left[g\left(F_1^{-1}(U_1), \ldots, F_d^{-1}(U_d)\right)\right],\end{aligned}$$

where $u := [u_1, \ldots, u_d]^\top$ and $\{U_k\}_{k=1,\ldots,d}$ is a sequence of iid uniform random variables on [0, 1]. Next, let Φ and ϕ be cumulative distribution and density of the standard normal respectively. Note that $\Phi(Z) \sim U(0,1)$ with $Z \sim \mathscr{N}(0,1)$ due to the standard inversion method $\Phi^{-1}(U) \sim \mathscr{N}(0,1)$ with $U \sim U(0,1)$. Also, recall that the normal law is closed under linear transformation, that is, $\mathscr{L}(AZ) = \mathscr{L}(Z)$ whenever A is an orthogonal matrix in $\mathbb{R}^{d\times d}$, with $Z := [Z_1, \ldots, Z_d]^\top$ being a standard d-dimensional normal random vector. To sum up, it holds that for each orthogonal matrix A in $\mathbb{R}^{d\times d}$ with $A_{k,\cdot}$ denoting its k-th row,

$$\begin{aligned}\mathscr{I}_d\left(\{F_k\}; g\right) &= \mathbb{E}\left[g\left(F_1^{-1}\left(\Phi\left(Z_1\right)\right), \ldots, F_d^{-1}\left(\Phi\left(Z_d\right)\right)\right)\right] \\ &= \mathbb{E}\left[g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}Z\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}Z\right)\right)\right)\right].\end{aligned}$$

The GLT method provides us a systematic way to determine the optimal matrix A^* to enhance the computational efficiency of the QMC method. Suppose that $g \in C^2(\mathbb{R}^d; \mathbb{R})$ and $F_k^{-1} \in C^1([0,1]; \mathbb{R})$, $k = 1, \ldots, d$. It holds by the Taylor theorem that for $z \in \mathbb{R}^d$ and $\epsilon \in \mathbb{R}^d$,

$$\begin{aligned}&g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}(z+\epsilon)\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}(z+\epsilon)\right)\right)\right) \\ &\quad = g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right) + \langle G(z), \epsilon\rangle + O(\|\epsilon\|^2), \qquad (13)\end{aligned}$$

where

$$G(z) := \nabla_z \left(g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right)\right),$$

and the asymptotics holds as $\|\epsilon\| \downarrow 0$. The n-th component of $G(z)$ is given by

$$\begin{aligned}&\frac{\partial}{\partial z_n} g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right) \\ &\quad = \sum_{k=1}^{d} \left[\partial_k g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right), \ldots, F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right)\right] \frac{\phi(A_{k,\cdot}z)}{f_k(F_k^{-1}(\Phi(A_{k,\cdot}z)))} a_{k,n},\end{aligned}$$

where a_{k_1,k_2} being the (k_1, k_2)-element of the orthogonal matrix A and where we have used $(d/dx)F_k^{-1}(x) = 1/f_k(F_k^{-1}(x))$. In particular, we have

$$\frac{\partial}{\partial z_n} g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right),\ldots,F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right)\Big|_{z=0}$$

$$= \sum_{k=1}^{d} \frac{\partial_k g(F_1^{-1}(1/2),\ldots,F_d^{-1}(1/2))}{\sqrt{2\pi}\, f_k(F_k^{-1}(1/2))} a_{k,n},$$

In order to extract large variance contribution from the first dimension of the low discrepancy sequence, we first maximize the first component of the coefficient $G(z)$ at the origin $z = 0$ based upon the following optimization problem

$$(P_1)\quad \left|\begin{array}{ll} \max\limits_{A_{\cdot,1}\in\mathbb{R}^d} & \left(\frac{\partial}{\partial z_1} g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right),\ldots,F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right)\Big|_{z=0}\right)^2 \\ \text{s.t.} & \|A_{\cdot,1}\| = 1. \end{array}\right.$$

We can show that the optimal vector $A^*_{\cdot,1}$ is given by

$$A^*_{\cdot,1} = \frac{\nabla g(F_1^{-1}(1/2),\ldots,F_d^{-1}(1/2))}{\|\nabla g(F_1^{-1}(1/2),\ldots,F_d^{-1}(1/2))\|}.$$

The optimal column vectors $A^*_{\cdot,k}$ are determined iteratively for $k = 2,\ldots,d$. One possible approach is to go further into higher order terms of the above Taylor expansion. This implies that finding optimal $A^*_{\cdot,k}$ requires the k-th order Taylor expansion, which can be very complex and time-consuming. A more logical and yet simpler solution is to rely on the Taylor approximation (13) except with expansion at different points. In the k-th optimization step, we derive the k-th column $A^*_{\cdot,k}$ to ensure the orthogonality against the earlier columns $\{A^*_{\cdot,l}\}_{l=1,\ldots,k-1}$, which have already been determined in the previous iterations. Formally, the optimization problem (P_k) is formulated as

$$(P_k)\quad \left|\begin{array}{ll} \max\limits_{A_{\cdot,k}\in\mathbb{R}^d} & \left(\frac{\partial}{\partial z_k} g\left(F_1^{-1}\left(\Phi\left(A_{1,\cdot}z\right)\right),\ldots,F_d^{-1}\left(\Phi\left(A_{d,\cdot}z\right)\right)\right)\Big|_{z=0}\right)^2 \\ \text{s.t.} & \|A_{\cdot,k}\| = 1, \\ & \langle A_{\cdot,k}, A_{\cdot,l}\rangle = 0, \quad l = 1,\ldots,k-1. \end{array}\right.$$

This problem (P_k) can be derived by first solving it without the orthogonality constraint, and then orthogonalize the rustling solution using the algorithms, such as the Gram-Schmidt method. (See [9] for the theoretical and technical details.) This complete iterative procedure, however, can be time-consuming, in particular for problems with large nominal dimensions d. In practice, the computational burden can be reduced by only seeking a sub-optimal orthogonal matrix with optimum columns up to some dimension $m(<d)$. The remaining columns are then randomly assigned as long as the orthogonality conditions are satisfied. This translates to a significant reduction in computational time when $m \ll d$.

The effectiveness of the QMC method can be practically assessed by the effective dimension. The effective dimension of function f in the truncation sense of Caflisch et al. [4] is defined as the smallest integer d such that

$$\sum_{u\in\{1,2,\ldots,d\}} \sigma^2(f_u) \geq p\sigma^2(f), \tag{14}$$

where $\sigma^2(f)$ denotes the total variance of the function f, where $\sigma^2(f_u)$ represents the variance attributes to the set u and where p is a quantile of the variance which is closely to 1, such as $p = 0.95$. The inequality (14) reads that the first d dimensions capture more than p of the total variance, even though its nominal dimension of f can be very large. In short, a problem with small effective dimension can use the greater uniformity in the lower-dimensional structure of the low discrepancy sequence. Accordingly, it is expected that the QMC method can estimate more accurate expectations with small number of iterations than the MC method. To investigate the effect of GLT method to decrease the effective dimension, we introduce a cumulative explanatory ratio (CER, for short), which is defined by

$$\text{CER}\,(d) = \frac{\sum_{u\in\{1,2,\ldots,d\}} \sigma^2(f_u)}{\sigma^2(f)}.$$

In short, this quantity represents the proportion of the variance captured by the first d dimensions. (See [6] for a further explanation of CER.) In view of the inequality (14), it is clear that the effective dimension is given by the smallest integer d such that the cumulative explanatory ratio exceeds p. Although it is in general difficult to provide an explicit expression for $\sigma^2(f_u)$, this can be estimated numerically, as shown in Sobol' [26] and Wang and Fang [27].

For a numerical example, let us take option pricing problems under the exponential variance gamma Lévy process of Madan and Seneta [20]. The variance gamma process $\{X_t : t \geq 0\}$ can be characterized by its characteristic function

$$\mathbb{E}\left[e^{iyX_t}\right] = e^{iy\mu t}\left(1 - iy\theta\nu + \frac{1}{2}\sigma^2\nu y^2\right)^{-t/\nu}, \quad y \in \mathbb{R},$$

with $(\mu, \theta, \sigma, \nu) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}_+$. It is well known that the variance gamma process can be expressed in two ways as

$$X_t \overset{\mathscr{L}}{=} \mu t + \theta Y_t + \sigma W_{Y_t} \overset{\mathscr{L}}{=} \mu t + Y_{t,p} - Y_{t,n}.$$

In the first expression, $\{W_t : t \geq 0\}$ is a standard Brownian motion in $\mathbb{R}$ and $\{Y_t : t \geq 0\}$ is a gamma process, independent of $\{W_t : t \geq 0\}$, with marginal density

$$f_{Y_t}(x) = \frac{b^{at}}{\Gamma(at)} x^{at-1} e^{-bx}, \quad x > 0, \tag{15}$$

with $a = b =: \nu^{-1} > 0$. In the second expression, $\{Y_{t,p} : t \geq 0\}$ and $\{Y_{t,n} : t \geq 0\}$ are independent gamma processes, each of which can be characterized based on (15) with $(a_p, b_p) = (\nu^{-1}, (\mu_p \nu)^{-1})$ and $(a_n, b_n) = (\nu^{-1}, (\mu_n \nu)^{-1})$, where $\mu_p = \frac{1}{2}\sqrt{\theta^2 + 2\sigma^2/\nu} + \theta/2$ and $\mu_n = \frac{1}{2}\sqrt{\theta^2 + 2\sigma^2/\nu} - \theta/2$, respectively. The QMC methods are applied in Ribeiro and Webber [23] and Avramidis and L'Ecuyer [2] to the gamma processes, respectively, based on the above two different expressions. In this paper, we examine the GLT method applied to the second expression. Due to [3], the two independent gamma processes admit series representations

$$Y_{t,p} \overset{\mathscr{L}}{=} \sum_{k=1}^{+\infty} e^{-\frac{\Gamma_{k,p}}{Ta_p}} \frac{V_{k,p}}{b_p} \quad \left(T_{k,p} \in [0,t]\right), \quad Y_{t,n} \overset{\mathscr{L}}{=} \sum_{k=1}^{+\infty} e^{-\frac{\Gamma_{k,n}}{Ta_n}} \frac{V_{k,n}}{b_n} \quad (T_{k,n} \in [0,t]),$$

where $\{\Gamma_{k,p}\}_{k\in\mathbb{N}}$ and $\{\Gamma_{k,n}\}_{k\in\mathbb{N}}$ are arrival times of a standard Poisson process, $\{V_{k,p}\}_{k\in\mathbb{N}}$ and $\{V_{k,n}\}_{k\in\mathbb{N}}$ are sequences of iid exponential random variables with unit mean, $\{T_{k,p}\}_{k\in\mathbb{N}}$ and $\{T_{k,n}\}_{k\in\mathbb{N}}$ are sequences of iid uniform random variables on $[0,T]$. Here, all the six random sequences are mutually independent and can easily be generated from the uniform distribution due to (4). In the both expressions, we truncate the infinite sum to $N = 100$ terms, as in Sect. 2. Now, we define the discrete time average of asset price dynamics

$$B_M := \frac{1}{M} \sum_{m=1}^{M} S_0 \exp\left[X_{\frac{m}{M}T}\right], \tag{16}$$

and its payoff $\max(B_M - K, 0)$, where M indicates the number of equidistant monitoring points in time and K denotes the strike price. In our framework, it is difficult to determine the orthogonal matrix A with respect to the entire function, as $\max(\cdot - K, 0)$ is not strictly differentiable and the series representations here involve the non-differentiable function $(T_k \in [0,t])$. Instead, we determine the orthogonal matrix A with respect to the terminal value B_1 where B_1 represents a terminal value of the underlying asset price. In other words, we solve the optimization problems (P_k) as though we wished to optimize estimation of $\mathbb{E}[B_1]$, no matter what the averaging frequency M is. (Hence, the dimension d in (12) is $4N$.)

Although this choice of the target function is not optimal for evaluating Asian options, we will show in the numerical example that this dimension reduction method can increase the CER even in pricing Asian options. In addition, it is worth noting that other dimension reduction techniques such as Brownian bridge construction and principal component construction to generate B_1 do not improve the accuracy of the QMC because X_T is expressed as a difference between $Y_{T,p}$ and $Y_{T,n}$, which are nonlinear functions of $V_{k,p}$, $V_{k,n}$, $\Gamma_{k,p}$ and $\Gamma_{k,n}$. In other words, we cannot generate B_1 with a single random number, hence Brownian bridge does not work to increase the CER. In fact, we confirm that the naive application of Brownian bridge construction decreases CERs, although we do not report those to avoid overloading the paper. We can expect the obtained orthogonal matrix A to enhance estimator efficiency for $\mathbb{E}[\max(B_1 - K, 0)]$, as $M = 1$ and the function $\max(x-K,0)$ has a structure fairly close to x itself. We will shortly provide numerical results to illustrate whether or not this approach is sensible for the complex settings with $M \geq 2$ as well. Next, it is well known that the assignment of the low discrepancy sequence is very important because earlier coordinates of the low discrepancy point set is more evenly (uniformly) scattered in the unit hypercube. We examine the GLT method with two ways of the allocation to reflect [6].

Scheme I (Separate Assignment) Every six coordinate of the $6N$-dimensional low discrepancy sequence $\{\mathrm{LD}_k\}_{k=1,\ldots,6N}$ is assigned in the order; for $k = 1,2,\ldots,N$

$$\begin{bmatrix} E_{k,p} \\ V_{k,p} \\ T_{k,p} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_k) \\ -\ln(1-\mathrm{LD}_{N+k}) \\ T \times \mathrm{LD}_{2N+k} \end{bmatrix}, \quad \begin{bmatrix} E_{k,n} \\ V_{k,n} \\ T_{k,n} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_{3N+k}) \\ -\ln(1-\mathrm{LD}_{4N+k}) \\ T \times \mathrm{LD}_{5N+k} \end{bmatrix}.$$

Scheme II (Alternate Assignment) Every six coordinate of the $6N$-dimensional low discrepancy sequence $\{\mathrm{LD}_k\}_{k=1,\ldots,6N}$ is assigned alternately;

$$\begin{bmatrix} E_{k,p} \\ V_{k,p} \\ T_{k,p} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_{6k-5}) \\ -\ln(1-\mathrm{LD}_{6k-3}) \\ T \times \mathrm{LD}_{6k-1} \end{bmatrix}, \quad \begin{bmatrix} E_{k,n} \\ V_{k,n} \\ T_{k,n} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_{6k-4}) \\ -\ln(1-\mathrm{LD}_{6k-2}) \\ T \times \mathrm{LD}_{6k} \end{bmatrix}.$$

Scheme III (Suboptimal GLT) Set the $6N$-dimensional low discrepancy sequence $\{\mathrm{LD}_k\}_{k=1,\ldots,6N}$ as

$$W_k \leftarrow \Phi^{-1}(\mathrm{LD}_k), \quad k = 1,\ldots,2N,$$

$$W_{2N+k} \leftarrow \Phi^{-1}(\mathrm{LD}_{3N+k}), \quad k = 1,\ldots,2N.$$

Let A be an orthogonal matrix in $\mathbb{R}^{4N\times 4N}$, define $W := [W_1,\ldots,W_{4N}]^\top$ and $W' := [W_1',\ldots,W_{4N}']^\top := AW$, and set

$$\mathrm{LD}_k' \leftarrow \Phi\left(W_k'\right), \quad k = 1,\ldots,4N,$$

and for $k = 1,\ldots,N$,

$$\begin{bmatrix} E_{k,p} \\ V_{k,p} \\ T_{k,p} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_k') \\ -\ln(1-\mathrm{LD}_{N+k}') \\ T \times \mathrm{LD}_{2N+k} \end{bmatrix}, \quad \begin{bmatrix} E_{k,n} \\ V_{k,n} \\ T_{k,n} \end{bmatrix} \leftarrow \begin{bmatrix} -\ln(1-\mathrm{LD}_{2N+k}') \\ -\ln(1-\mathrm{LD}_{3N+k}') \\ T \times \mathrm{LD}_{5N+k} \end{bmatrix}.$$

Under this allocation, we implement the optimization problems (P_k) to determine the orthogonal matrix A.

Although we have described the GLT method here based on Scheme I, it is certainly possible to start with Scheme II or any other assignments instead. In fact, the choice of the original assignment has no effect on the result because the assignment can

Table 3 Cumulative explanatory ratio (CER) in percentile up to 30 dimension.

M	1			4			12			50			256		
Scheme	I	II	III	I	II	III	I	II	III	I	II	III	I	II	III
6	10	37	85	9	35	72	9	35	67	8	35	65	8	35	64
12	10	61	89	9	60	75	9	59	70	9	59	67	8	59	67
18	10	76	90	9	74	76	9	74	71	9	73	68	8	73	68
24	10	85	92	9	84	77	9	83	71	9	83	69	9	83	68
30	11	90	95	10	90	79	9	89	73	9	89	71	9	89	70

be rearranged by a permutation matrix, that is orthogonal. In other words, the GLT method can be carried out in a consistent manner, regardless of the original assignment.

In our experiments, we fix parameter values

$$(\mu,\, \sigma,\, \theta,\, T,\, r,\, S_0,\, K) = (-0.1436,\, 0.12136,\, 0.3,\, 1.0,\, 0.1,\, 100,\, 101)\,,$$

which we draw from [23]. We use a scrambled version of Sobol' low discrepancy sequences and employ the Latin supercube sampling method to reduce the dimension of the Sobol' sequence. (See [6] and references therein for details.) We examine the effectiveness of the proposed method through CER. In Table 3, we report CERs of every six dimensions up to 30, since then nominal dimension is $6N$. Recall that we fix $N = 100$. We have confirmed that this truncation level is sufficient in our preliminary numerical experiences. The numbers (6, 12, 18, 24, 30) in the leftmost column indicate the dimension. Note that the nominal dimension here is not affected by the averaging frequency M. In particular, with Euler discretization methods, the nominal dimension increases in proportion to M.

The CERs in Scheme II are much larger than those in Scheme I. This is consistent with the results reported by [6] and ensures the importance of assignment of the low discrepancy sequence. Scheme III proves most efficient in terms of CER. It achieves the highest CER with the first six dimensions, while the effectiveness decreases as M increases. This is so because the optimality gained in Scheme III is derived for the terminal value B_1 (not even its function $\max[B_1 - K, 0]$). Although Scheme II captures less CER than Scheme III with the first six dimension, it succeeds to capture as many as almost 90% with the first thirty dimensions in all the settings. This is also consistent with [6], indicating the advantage of applying the low discrepancy sequences to the series representation. Generally speaking, compared to the other dimension reduction methods in the QMC framework, the GLT method seems to work efficiently in a broader range of simulation problems as it looks closely at the structure of the integrands. The method is applicable, in principle, if probability density functions are available (or computable), while discontinuity of the integrand needs to be avoided. Finally, it is worth mentioning that direct applications of other dimension reduction methods, such as generalized principal component construction (L'Ecuyer et al. [17]) and generalized Brownian bridge construction (Leobacher [18]), seem to fail, as they do not take the structure of the integrands into consideration, unlike in the GLT method.

4 Concluding Remarks

In this article, we have proposed various ways of enhancing efficiency of the series representation of infinitely divisible laws and Lévy processes in Monte Carlo and quasi-Monte Carlo methods. The variance reduction methods, such as stratified sampling, control variates and importance sampling, applied to exponential interarrival times forming the shot noise series, have proved their significance in estimator accuracy. We have also observed that the GLT method in the QMC framework is well applicable in the series representation. We expect that the proposed techniques, together with [6, 7], contribute to wider use and the dissemination of the series representation for numerical purposes in various potential fields of application.

Acknowledgements The authors would like to thank an anonymous referee for various valuable comments and Japan Society for the Promotion of Science for Grant-in-Aid for Scientific Research 21340024 and 21710157.

References

1. Asmussen, S., Glynn, P.W.: *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) 1930–1944 (2006)
3. Bondesson, L.: On simulation from infinitely divisible distributions, *Advances in Applied Probability*, **14**(4) 855–869 (1982)
4. Caflisch, R., Morokoff, W., Owen, A.: Valuation of mortgaged-backed securities using Brownian bridges to reduce effective dimension, *Journal of Computational Finance*, **1**(1) 27–46 (1997)
5. Ferguson, T.S., Klass, M.J.: A representation of independent increment processes with Gaussian components, *Annals of Mathematical Statistics*, **43**(5) 1634–1643 (1972)
6. Imai, J., Kawai, R.: Quasi-Monte Carlo methods for infinitely divisible random vectors via series representations, *SIAM Journal on Scientific Computing*, **32**(4) 1879–1897 (2010)
7. Imai, J., Kawai, R.: Numerical inverse Lévy measure method for infinite shot noise series representation, preprint.
8. Imai, J., Kawai, R.: On finite truncation of infinite shot noise series representation of tempered stable laws, *Physica A*, **390**(23–24) 4411–4425 (2011)
9. Imai, J., Tan, K.S.: A general dimension reduction technique for derivative pricing, *Journal of Computational Finance*, **10** 129–155 (2007)
10. Imai, J., Tan, K.S.: An accelerating quasi-Monte Carlo method for option pricing under the generalized hyperbolic Lévy process, *SIAM Journal on Scientific Computing*, **31**(3) 2282–2302 (2009)
11. Kawai, R.: An importance sampling method based on the density transformation of Lévy processes, *Monte Carlo Methods and Applications,* **12**(2) 171–186 (2006)

12. Kawai, R.: Adaptive Monte Carlo variance reduction with two-time-scale stochastic approximation, *Monte Carlo Methods and Applications*, **13**(3) 197–217 (2007)
13. Kawai, R.: Adaptive Monte Carlo variance reduction for Lévy processes with two-time-scale stochastic approximation, *Methodology and Computing in Applied Probability*, **10**(2) 199–223 (2008)
14. Kawai, R.: Optimal importance sampling parameter search for Lévy processes via stochastic approximation, *SIAM Journal on Numerical Analysis*, **47**(1) 293–307 (2008)
15. Kawai, R.: Asymptotically optimal allocation of stratified sampling with adaptive variance reduction by strata, *ACM Transactions on Modeling and Computer Simulation*, **20**(2) Article 9 (2010)
16. Kawai, R., Takeuchi, A.: Greeks formulas for an asset price model with gamma processes, *Mathematical Finance*, **21**(4) 723–742 (2011)
17. L'Ecuyer, P., J-S. Parent-Chartier, M. Dion: Simulation of a Lévy process by PCA sampling to reduce the effective dimension, In: S.J. Mason, R.R., et al. (Eds.) *Proceedings of the 2008 Winter Simulation Conference*, 436–442 (2008)
18. Leobacher, G.: Stratified sampling and quasi-Monte Carlo simulation of Lévy processes, *Monte Carlo Methods and Applications*, **12**(3–4) 231–238 (2006)
19. LePage, R.: Multidimensional infinitely divisible variables and processes II, In: *Lecture Notes in Mathematics* 860, Springer-Verlag, Berlin, New York, Heidelberg, 279–284 (1980)
20. Madan, D.B., Seneta, E.: The variance gamma (V.G.) model for share market returns, *Journal of Business*, **63**(4) 511–524 (1990)
21. Madan, D.B., Yor, M.: Representing the CGMY and Meixner Lévy processes as time changed Brownian motions, *Journal of Computational Finance*, **12**(1) (2008)
22. Owen, A.B.: Randomly permuted (t,m,s)-nets and (t,s)-sequence, In: *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, H. Niederreiter, J.S. Shiue (Eds.) Springer-Verlag, New York, 299–317 (1995)
23. Ribeiro, C., Webber, N.: Valuing path-dependent options in the variance-gamma model by Monte Carlo with a gamma bridge, *Journal of Computational Finance*, **7**, 81–100 (2003)
24. Rosiński, J.: Series representations of Lévy processes from the perspective of point processes, In: *Lévy Processes – Theory and Applications*, O-E. Barndorff-Nielsen et al. (Eds.) Birkhäuser, Boston, 401–415 (2001)
25. Sato, K.: *Lévy processes and infinitely divisible distributions,* Cambridge University Press, Cambridge (1999)
26. Sobol', I.M.: Distribution of points in a cube and integration nets, *UMN*, **5**(131) 271–272 (1966)
27. Wang, X., Fang, K.T.: The effective dimension and quasi-Monte Carlo integration, *Journal of Complexity*, **19**(2) 101–124 (2003)

Parallel Quasi-Monte Carlo Integration by Partitioning Low Discrepancy Sequences

Alexander Keller and Leonhard Grünschloß

Abstract A general concept for parallelizing quasi-Monte Carlo methods is introduced. By considering the distribution of computing jobs across a multiprocessor as an additional problem dimension, the straightforward application of quasi-Monte Carlo methods implies parallelization. The approach in fact partitions a single low-discrepancy sequence into multiple low-discrepancy sequences. This allows for adaptive parallel processing without synchronization, i.e. communication is required only once for the final reduction of the partial results. Independent of the number of processors, the resulting algorithms are deterministic, and generalize and improve upon previous approaches.

1 Introduction

The performance of many algorithms can be increased by parallelization and in fact parallel processors are ubiquitous. A recent survey [9, Sect. 6.4] identifies three approaches to parallel quasi-Monte Carlo integration [16]: Besides leapfrogging along a low discrepancy sequence or enumerating blocks of a low discrepancy sequence, low discrepancy sequences can be randomized. While randomization is simple, it requires a sacrifice of some convergence speed and enumerating blocks is not necessarily deterministic due to race conditions [18, Sect. 3.1]. The most desirable scheme would be deterministic for exact reproducibility and avoid any compromises on convergence.

A. Keller
NVIDIA ARC GmbH, Berlin, Germany
e-mail: keller.alexander@googlemail.com

L. Grünschloß
Rendering Research, Weta Digital, New Zealand
e-mail: leonhard@gruenschloss.org

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_27, © Springer-Verlag Berlin Heidelberg 2012

Finance and computer graphics are among the domains that would benefit from such an approach. In the latter domain, a method for the parallel generation of photon maps [1] has been introduced (see [8] for a solid introduction to photon mapping and [11] in this volume for a compact summary). Core of the method was a number theoretic argument similar to [12] that allowed for partitioning one Halton sequence into a number of sequences. Since all of these sequences were of low discrepancy, too, each job using such a subsequence consumed about a similar number of samples when independently adaptively terminated without communication for synchronization. The resulting union of samples is a complete initial segment of the Halton sequence followed by a comparatively small segment of samples, where the Halton sequence is used only partially.

As summarized in the survey [9, Sect. 6.4] and reported in [2, 9, 12, 18], transferring the approach to (t, s)-sequences in a straightforward way has defects and rank-1 lattice sequences [7] have not yet been considered.

In the following a strictly deterministic scheme is introduced: Based on a generalized and simplified argument on how to partition quadrature rules for parallel processing, efficient algorithms for generating the stream of samples inside each parallel job are derived.

2 Parallelization as an Additional Problem Dimension

The distribution of jobs over multiple processors working in parallel can be considered as one additional problem dimension.

For the example of the integration problem this means that in addition to the integrand dimensions we also integrate over the maximum number N of possibly parallel jobs. A job $j \in \{0, \dots, N-1\} \subset \mathbb{N}_0$ will be selected by the characteristic function

$$\chi_j(x') := \begin{cases} 1 & j \le x' < j+1 \\ 0 & \text{otherwise,} \end{cases}$$

that simply assigns the interval $[j, j+1)$ to the j-th job. Exploiting the fact that

$$1 = \sum_{j=0}^{N-1} \chi_j(x') \text{ for } 0 \le x' < N, \tag{1}$$

we rewrite the s-dimensional integral of a function f over the unit cube as

$$\begin{aligned}\int_{[0,1)^s} f(\mathbf{x}) d\mathbf{x} &= \frac{1}{N} \int_0^N 1 \cdot \int_{[0,1)^s} f(\mathbf{x}) d\mathbf{x} dx' \\ &= \sum_{j=0}^{N-1} \underbrace{\int_{[0,1)} \int_{[0,1)^s} \chi_j(N \cdot x') \cdot f(\mathbf{x}) d\mathbf{x} dx'}_{=:S_j},\end{aligned}$$

where we first added an integral over the number of jobs N, inserted the partition of one from Eq. 1, and finally transformed everything to the $s+1$-dimensional unit cube.

Selecting one $s+1$-dimensional low-discrepancy sequence [16] of points $\mathbf{x}_i = (x_{i,0}, \ldots, x_{i,s})$ to simultaneously approximate all summands

$$S_j \approx \frac{1}{n} \sum_{i=0}^{n-1} \chi_j(N \cdot x_{i,c}) \cdot f(x_{i,0}, \ldots, x_{i,c-1}, x_{i,c+1}, \ldots, x_{i,s}) \qquad (2)$$

lends itself to a parallelization scheme: Due to the above property from Eq. 1 the characteristic function[1] χ_j partitions the set of samples by job number j. In fact an arbitrarily chosen dimension c is partitioned into N equally sized intervals (see the illustration in Fig. 1) and each job only consumes the points of the sequence which fall into its interval.[2]

3 Algorithms for Partitioning Low Discrepancy Sequences

Given a low discrepancy sequence $\mathbf{x}_i$, the point sets

$$P_j := \left\{\mathbf{x}_i : \chi_j(N \cdot x_{i,c}) = 1, i \in \mathbb{N}_0\right\} = \{\mathbf{x}_i : j \leq N \cdot x_{i,c} < j+1, i \in \mathbb{N}_0\}$$

are low discrepancy sequences, too, because they result from a partitioning by planes perpendicular to the axis c, which does not change the order of discrepancy [16]. Similarly, any subsequence $(x_{i,0}, \ldots, x_{i,c-1}, x_{i,c+1}, \ldots, x_{i,s})$ resulting from the omission of the component c, is of low discrepancy. In fact this can be interpreted as a simple way to partition a low discrepancy sequence into low discrepancy sequences (see the illustration in Fig. 1).

For the common number theoretic constructions of quasi-Monte Carlo point sequences and a suitable choice of N, the integer part of $N \cdot x_{i,c}$ results in successive permutations of $\{0, \ldots, N-1\}$. Based on this observation we derive efficient algorithms to enumerate the set

[1] Actually, any quadrature rule could be chosen.

[2] The partitions can also be scaled to fill the $(s+1)$-dimensional unit cube again. In other words, one could reuse the component chosen for selecting samples for each job, which is more efficient since one component less must be generated. Reformulating Eq. 2 accordingly, requires only the generation of s-dimensional samples:

$$S_j \approx \frac{1}{n} \sum_{i=0}^{n-1} \chi_j(N \cdot x_{i,c}) \cdot f(x_{i,1}, \ldots, x_{i,c-1}, N \cdot x_{i,c} - j, x_{i,c+1}, \ldots, x_{i,s})$$

However, this variant is not recommended, because the resulting ensemble of samples may not be well-stratified in the dimension c.

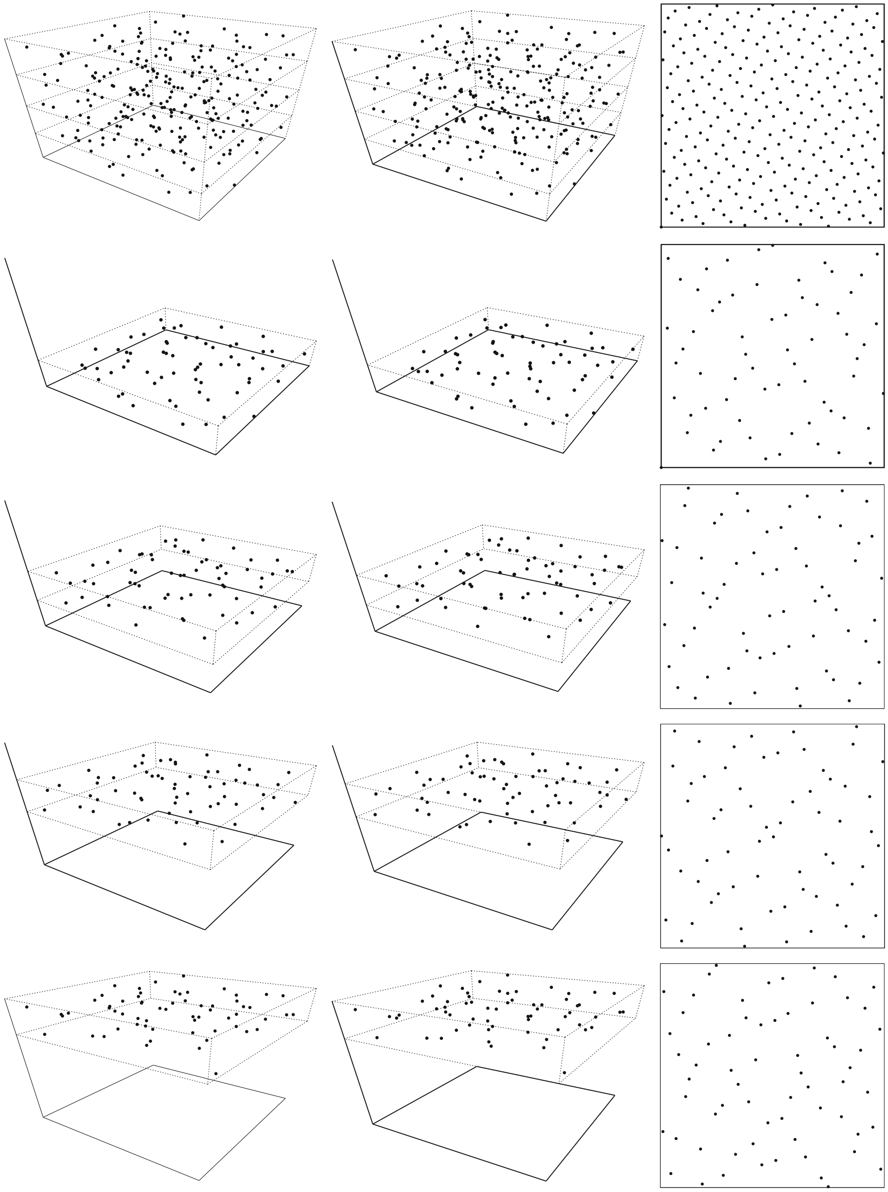


Fig. 1 The points of the three-dimensional Sobol' sequence in the first row are partitioned along the vertical axis. The resulting partitions are shown in the following four rows. Each row shows two three-dimensional plots of the same points from two perspectives that can be viewed cross-eyed, resulting in a stereoscopic impression. The rightmost plot in each row shows the two-dimensional projection of the corresponding points along the vertical axis.

$$P_j = \left\{\mathbf{x}_{i_{j,l}} : l \in \mathbb{N}_0\right\} \tag{3}$$

for the j-th job using an index of the form $i_{j,l} := lN + k_{j,l}$, where $k_{j,l} \in \{0, \ldots, N-1\}$.

3.1 Preliminaries on Digits and Digital Radical Inverses

We introduce some notation and facts that are used throughout the derivations.

For any number $r \in \mathbb{R}_0^+$, we define the k-th digit $a_k(r) \in \{0, \ldots, b-1\}$ in integer base b by

$$r = \sum_{k=-\infty}^{\infty} a_k(r) b^k.$$

Note that this definition includes fractional digits for $k < 0$.

Digital radical inverses

$$\begin{aligned}\phi_{b,C} &: \mathbb{N}_0 \to \mathbb{Q} \cap [0,1)\\ i &\mapsto \left(b^{-1} \cdots b^{-M}\right) \left[C \begin{pmatrix} a_0(i) \\ \vdots \\ a_{M-1}(i) \end{pmatrix} \right]\end{aligned} \tag{4}$$

in base b are computed using a generator matrix C, where the inverse mapping $\phi_{b,C}^{-1}$ exists, if C is regular. While in theory these matrices are infinite-dimensional, in practice they are finite due to the finite precision of computer arithmetic. M is the number of digits, which allows for generating up to $N = b^M$ points. Note that the matrix-vector multiplications are performed in the finite field $\mathbb{F}_b$ (for the theory and mappings from and to $\mathbb{F}_b$ see [16]) and are additive in $\mathbb{F}_b$ in the sense that, for any $0 \le M' \le M$,

$$C \begin{pmatrix} a_0(i) \\ \vdots \\ a_{M-1}(i) \end{pmatrix} = C \begin{pmatrix} a_0(i) \\ \vdots \\ a_{M'-1}(i) \\ 0 \\ \vdots \\ 0 \end{pmatrix} + C \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{M'}(i) \\ \vdots \\ a_{M-1}(i) \end{pmatrix}. \tag{5}$$

3.2 Halton-Type Sequences

The components

$$\begin{aligned} \phi_b : \mathbb{N}_0 &\to \mathbb{Q} \cap [0,1) \\ i = \sum_{k=0}^{\infty} a_k(i) b^k &\mapsto \sum_{k=0}^{\infty} a_k(i) b^{-k-1} \end{aligned} \tag{6}$$

of the Halton sequence [16] are the radical inverses in integer base b, where all bases are relatively prime. In fact, $\phi_b = \phi_{b,C}$ for $C = I$ the identity matrix.

While originally the digit $a_k(i)$ has been the k-th digit of the index i represented in base b, the uniformity of the points has been improved by applying permutations to the digits before computing the radical inverse: Zaremba [23] was successful with the simple permutation $\pi_b(a_k(i)) := a_k(i) + k \bmod b$, while later on Faure [5] developed a more general set of permutations improving upon Zaremba's results.

For $x_{i,c} = \phi_b(i)$ choosing the number of jobs as $N = b^m$, $m \in \mathbb{N}$, yields

$$\lfloor N \cdot x_{i,c} \rfloor = \lfloor b^m \cdot \phi_b(i) \rfloor = \left\lfloor b^m \cdot \sum_{k=0}^{\infty} a_k(i) b^{-k-1} \right\rfloor,$$

whose integer part is a permutation of $\{0, \ldots, N-1\}$, which is repeated every N points. Each job j thus is assigned the set

$$P_j = \left\{ \mathbf{x}_{l \cdot N + \phi_b^{-1}(j/N)} : l \in \mathbb{N}_0 \right\} \Leftrightarrow P_{\phi_b^{-1}(j/N)} = \left\{ \mathbf{x}_{l \cdot N + j} : l \in \mathbb{N}_0 \right\}$$

which is known as leapfrogging and coincides with previous findings in [1,12]. Note that the offset $\phi_b^{-1}(j/N)$ is constant per job, and for the ease of implementation we can simply permute the assignment of low-discrepancy subsequences to the jobs, i.e. assign the j-th job the set

$$P_j = \left\{ \mathbf{x}_{l \cdot N + j} : l \in \mathbb{N}_0 \right\}.$$

Compared to the previous derivations [1, 12], our argument is simpler and more general as it includes scrambled radical inverses, too: The condition from [1] that N must be relatively prime to the bases of the radical inverses used for actual sampling just follows from the definition of the Halton sequence.

3.3 Digital (t,s)-Sequences in Base b

The d-th component $x_{i,d} = \phi_{b,C_d}(i)$ of a digital (t,s)-sequence in base b is computed using a generator matrix C_d, where ϕ_{b,C_d} is defined in Eq. 4. Since we are considering only the algorithmic part, we refer to [16] for a more profound introduction.

The Sobol' sequence [21] is a common (t,s)-sequence in base $b=2$. Its generator matrix for the first dimension is the identity matrix I, and thus the first component coincides with the van der Corput radical inverse in base $b=2$ from Eq. 6. As a consequence all results from the previous section apply (see the illustration in Fig. 1), however, compared to the Halton sequence, the Sobol' sequence in base 2 can be computed much faster: Efficient implementations of the first two components can be found in [13] and in [6, 22] for the remaining components.

While $N = b^m$ remains a natural choice for an arbitrary C_c, determining the set P_j might not result in a simple leapfrogging scheme, because each column of the generator matrix can influence the final result.

However, if C_c is required to generate a $(0,1)$-sequence, it is known that C_c is invertible [16]. A remark in [14] states that multiplying a regular matrix to the right of all generator matrices generates the same (t,s)-sequence, except for the numbering of the points. Therefore we propose to use

$$C_0C_c^{-1}, \ldots, C_{c-1}C_c^{-1}, I, C_{c+1}C_c^{-1}, \ldots, C_sC_c^{-1}$$

as generator matrices for the sequence. The component c then is generated by the identity matrix $I = C_cC_c^{-1}$, which allows one to efficiently determine the set P_j in analogy with the previous section.

If C_c is a regular upper triangular matrix, it is possible to compute the indices $i_{j,l}$ for every job without reordering: Due to the upper triangular structure, the m least significant digits of the index can only influence the first m digits of the radical inverse, however, the remaining index digits can influence all digits of the radical inverse, especially its first m digits. Given l and the job number j, we thus can compute

$$k_{j,l} = i_{j,l} - lb^m = \phi_{b,C_c}^{-1}\left(\sum_{k=0}^{m-1}(a_k(j) - a_{-k-1}(y_l))b^{-m+k}\right), \tag{7}$$

where the subtraction of digits has to be performed in $\mathbb{F}_b$ and $y_l = \phi_{b,C_c}(lb^m)$. This formula is best understood by first looking at the case $l = 0$, i.e. the first $N = b^m$ points, where

$$k_{j,0} = i_{j,0} = \phi_{b,C_c}^{-1}\left(\sum_{k=0}^{m-1} a_k(j)b^{-m+k}\right) = \phi_{b,C_c}^{-1}(j/N) \in \{0, \ldots, b^m - 1\}$$

just is the inverse permutation generated by C_c that maps the job number to an index. For $l > 0$, the contribution of $y_l = \phi_{b,C_c}(lb^m)$ to the m least significant digits of the index has to be compensated. This is accomplished by subtracting the digits $a_{-k-1}(y_l)$ as in Eq. 7 based on the additive property (5).

3.4 Rank-1 Lattice Sequences

For a suitable generator vector $\mathbf{h} \in \mathbb{N}^{s+1}$, the points of a rank-1 lattice sequence [7] are computed by

$$\mathbf{x}_i := \{\phi_{b,C}(i) \cdot \mathbf{h} + \Delta\} \in [0,1)^{s+1},$$

where $\{\cdot\}$ denotes the fractional part. The radical inverse $\phi_{b,C}$ has been defined in Eq. 4 and $\Delta \in [0,1)^{s+1}$ is an arbitrary shift vector.

Restricting $C \in \mathbb{F}_b^{M \times M}$ to upper triangular, invertible matrices, the l-th run of b^m points consists of the first b^m points shifted by

$$\Delta_l := (\Delta_{l,0}, \dots, \Delta_{l,s}) = \phi_{b,C}(lb^m) \cdot \mathbf{h}.$$

We therefore choose the number of jobs to be $N = b^m$ and enumerate the points of P_j using an index $i_{j,l} := lb^m + k_{j,l}$ with $k_{j,l} \in \{0, \dots, b^m - 1\}$ of the form introduced in Eq. 3. Note that under the above restriction $b^m \phi_{b,C}(k_{j,l})$ is integer.

Given j and l, $k_{j,l}$ is found by solving the following congruence resulting from taking the integer part $\lfloor\cdot\rfloor$ of the component $x_{i,c} = \{\phi_{b,C}(lb^m + k_{j,l})h_c + \Delta_c\}$ used for job selection multiplied by the number of jobs:

$$\begin{aligned}
\lfloor N \cdot x_{i,c} \rfloor &\equiv j \mod b^m \\
\lfloor b^m \{\phi_{b,C}(lb^m)h_c + \phi_{b,C}(k_{j,l})h_c + \Delta_c\} \rfloor &\equiv j \mod b^m \\
\Leftrightarrow \lfloor b^m (\phi_{b,C}(lb^m)h_c + \phi_{b,C}(k_{j,l})h_c + \Delta_c) \rfloor &\equiv j \mod b^m \\
\Leftrightarrow \lfloor b^m \phi_{b,C}(lb^m)h_c + (b^m \phi_{b,C}(k_{j,l}))h_c + b^m \Delta_c \rfloor &\equiv j \mod b^m \\
\Leftrightarrow (b^m \phi_{b,C}(k_{j,l}))h_c + \lfloor b^m \phi_{b,C}(lb^m)h_c + b^m \Delta_c \rfloor &\equiv j \mod b^m
\end{aligned}$$

and hence

$$k_{j,l} = \phi_{b,C}^{-1}\left(\underbrace{\left((j - \lfloor b^m \phi_{b,C}(lb^m)h_c + b^m \Delta_c \rfloor)\,(h_c)^{-1} \bmod b^m\right)}_{\in\{0,\dots,b^m-1\}} b^{-m} \right),$$

where $(h_c)^{-1}$ is the *modular multiplicative inverse* of h_c, which can be computed by using the extended form of Euclid's algorithm [3, Sect. 31.2, p. 937]. Note that this inverse exists if and only if b and h_c are relatively prime. For this last equation, $a \bmod b^m$ denotes the common residue, i.e. the nonnegative integer $x < b^m$, such that $a \equiv x \mod b^m$.

If now C is the identity matrix I, the inverse permutation $\phi_{b,I}^{-1} \equiv \phi_b^{-1}$ can be omitted, which is more efficient to compute and only reorders the jobs similar to Sect. 3.2. In addition $b^m \phi_b(lb^m) = \phi_b(l)$. An alternative and simplifying approach to the case, where the order of the points does not matter, is to multiply the generator matrix C by its inverse C^{-1} (see the aforementioned remark in [14]).

4 Parallel Quasi-Monte Carlo Integration

Considering parallelization as an additional problem dimension leads to a partition of low discrepancy sequences, where all subsequences (as described in Sect. 2) are of low discrepancy, too. For radical inversion based sequences, the subsequences P_j can be efficiently enumerated as derived in Sect. 3. In the following, more practical aspects are discussed.

4.1 Previous Issues with Leapfrogging

In literature (see the survey [9, Sect. 6.4]), leapfrogging for parallelizing quasi-Monte Carlo integration has been discussed in a controversial way. The technique has been introduced in [2], where the Sobol' low discrepancy sequence has been partitioned by leaping along the sequence in equidistant steps of size 2^m. The resulting algorithm allows for the very efficient enumeration of the subsequences, however, it "may lead to dramatic defects" [18] in parallel computation.

The underlying issue can be understood based on a simple example: The subsequences

$$\phi_b(l \cdot b^m + j) = \underbrace{b^{-m} \cdot \phi_b(l)}_{\in[0,b^{-m})} + \phi_b(j) \in [\phi_b(j), \phi_b(j) + b^{-m}) \neq [0,1) \text{ for } m > 0 \tag{8}$$

of the radical inverse (6) resulting from leapfrogging using steps of length b^m are not of low discrepancy and not even uniform over $[0, 1)$ as they reside in strata [10, Sect. 3.4] not covering the whole unit interval.

In fact ϕ_2 is the first component of the Sobol' sequence, which is a (t,s)-sequence in base $b = 2$. As the first component of the subsequence identified by $j = 0$ is completely contained in $[0, 2^{-m})$, this subsequence is neither of low discrepancy nor uniform over the unit cube. In fact Lemma 8 and Remarks 9 and 10 in [19] coincide with the above explanation. As a conclusion, leaping along the sequence in equidistant steps does not guarantee that the subsequences are uniform or of low discrepancy as illustrated by the numerical experiments in [4, Fig. 3].

Obviously, these defects will not show up, if the same number of samples from all subsequences is used, but this would not allow for adaptive sampling, as discussed in the next section.

Although not aimed at parallel computation but at improving the uniformity of low discrepancy sequences, leapfrogging applied to the Halton-sequence and (t,s)-sequences has been investigated in [12], too. Similar to [1], subsequences of the Halton sequence were generated by leapfrogging with a constant step size of a prime, which is relatively prime to all bases used in the Halton sequence [12, Sect. 2.4]. While this coincides with our results, the derivation in Sect. 3.2 is more general, as it includes scrambled radical inverses [15] (see Sect. 3.3), which do not result in equidistant step size for leapfrogging.

A short investigation of the effect of leapfrogging on the Sobol' sequence in [12, Sect. 3.2] yields that for leapfrogging with step size of a power of $b = 2$, components need to be either omitted or scaled. The requirement of scaling (see also Footnote 2) can be explained with the stratification properties (8), while the omission is included in the more general results of Sect. 3.3.

In summary, leapfrogging works whenever one component of the sequence is used to determine the leapfrogging pattern, while the same component is excluded from sampling as discussed in Sect. 2.

4.2 Adaptive Sampling

In adaptive algorithms the total number of samples depends on the input and although adaptive sampling can arbitrarily fail [20], it has proved very useful in practice. Parallel adaptive sampling involves the cost of communicating termination among the processing units, load balancing to equalize for differences in performance, and potentially random elements due to race conditions.

As explained in the previous section, the defects observed in [18] will not show up with algorithms developed in Sect. 3, because now all subsequences are of low discrepancy. In addition, communication cost can be restricted to the minimum of the inevitable final parallel reduction operation, which sums up the partial sums of Eq. 2: Running a copy of the adaptive algorithm for each subsequence, in practice each job will use about the same number of samples, because each subsequence is of low discrepancy. Using all partial sequences, the result is computed by a contiguous beginning block of samples of the underlying partitioned sequence followed by a usually quite short segment of partial samples from the underlying sequence.

Straightforward load balancing for a number P of heterogenous processors can be achieved by selecting the number of jobs $N \gg P$. Then a simple job queue can be used to distribute jobs among the processors. If the number of jobs is not excessive, the queue synchronization overhead is negligible.

The scheme is strictly deterministic, because the total number of samples of each job is independent of the processor on which the job is executed and therefore race conditions cannot occur. Computations are even independent of the number P of processors used: Executing all jobs sequentially results in exactly the same result as obtained by the parallel execution.

4.3 Selecting the Number of Jobs

Halton, (t, s)-, and lattice sequences are all based on radical inversion, which results in a number of jobs of the form $N = b^m$.

For the Halton sequence the base b is determined by the choice of the dimension c used for partitioning the sequence. Identical to [1], c and thus b should be chosen as large as possible in order to benefit from the better discrepancy and stratification properties of the first dimensions of the Halton sequence.

For (t,s)- and lattice sequences the base b is identical for all dimensions. Instead of considering which dimension to use for partitioning, it is much more interesting to choose $b = 2$, which allows for very efficient sample enumeration by using bit vector arithmetic [6, 13, 22].

Choosing the number n of samples in Eq. 2 as a multiple of $N = b^m$, allows one to use finite point sets like the Hammersley points, (t,m,s)-nets, and rank-1 lattices. In this case convergence can benefit from the better discrepancy of finite point sets.

The algorithms remain consistent even for a number of jobs $N < b^m$, because each set of the partition is of low discrepancy (see Sect. 3). However, omitting $b^m - N$ sets of the partition is likely to sacrifice some convergence speed.

5 Conclusion

We introduced a method to partition number theoretic point sequences into subsequences that preserve the properties of the original sequence. The resulting algorithms can be classified as generalized leapfrogging [2, 9, 18]. Instead of multiplying the number of problem dimensions with the processor count [17], adding only one dimension is sufficient for our approach, which in addition allows one to benefit from lower discrepancy.

The presented algorithms are deterministic and run without races in any parallel computing environment, i.e. the computation is identical for a fixed number N of jobs no matter how many processors are used.

As a practical consequence, photon maps now can be generated adaptively in parallel similar to [1], however, taking full advantage of the much faster generation of (t,s)-sequences in base 2 [6, 13, 22], which had not been possible before.

Acknowledgements This work has been dedicated to Stefan Heinrich's 60th birthday. The authors thank Matthias Raab for discussion.

References

1. Abramov, G.: US patent #6,911,976: System and method for rendering images using a strictly-deterministic methodology for generating a coarse sequence of sample points (2002)
2. Bromley, B.: Quasirandom number generators for parallel Monte Carlo algorithms. J. Parallel Distrib. Comput. **38**(1), 101–104 (1996)
3. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press (2009)
4. Entacher, K., Schell, T., Schmid, W., Uhl, A.: Defects in parallel Monte Carlo and quasi-Monte Carlo integration using the leap-frog technique. Parallel Algorithms Appl. pp. 13–26 (2003)
5. Faure, H.: Good permutations for extreme discrepancy. J. Number Theory **42**, 47–56 (1992)
6. Grünschloß, L.: Motion Blur. Master's thesis, Universität Ulm (2008)

7. Hickernell, F., Hong, H., L'Ecuyer, P., Lemieux, C.: Extensible lattice sequences for quasi-Monte Carlo quadrature. SIAM J. Sci. Comput. **22**, 1117–1138 (2000)
8. Jensen, H.: Realistic Image Synthesis Using Photon Mapping. AK Peters (2001)
9. Jez, P., Uhl, A., Zinterhof, P.: Applications and parallel implementation of QMC integration. In: R. Trobec, M. Vajteršic, P. Zinterhof (eds.) Parallel Computing, pp. 175–215. Springer (2008)
10. Keller, A.: Myths of computer graphics. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 217–243. Springer (2006)
11. Keller, A., Grünschloß, L., Droske, M.: Quasi-Monte Carlo progressive photon mapping. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 501–511. Springer (2012)
12. Kocis, L., Whiten, W.: Computational investigations of low-discrepancy sequences. ACM Trans. Math. Softw. **23**(2), 266–294 (1997)
13. Kollig, T., Keller, A.: Efficient multidimensional sampling. Computer Graphics Forum (Proc. Eurographics 2002) **21**(3), 557–563 (2002)
14. Larcher, G., Pillichshammer, F.: Walsh series analysis of the L_2-discrepancy of symmetrisized point sets. Monatsh. Math. **132**, 1–18 (2001)
15. Matoušek, J.: On the L_2-discrepancy for anchored boxes. J. Complexity **14**(4), 527–556 (1998)
16. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
17. Ökten, G., Srinivasan, A.: Parallel quasi-Monte Carlo methods on a heterogeneous cluster. In: K.T. Fang, F. Hickernell, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 406–421. Springer (2002)
18. Schmid, W., Uhl, A.: Parallel quasi-Monte Carlo integration using (t, s)-sequences. In: ParNum '99: Proceedings of the 4th International ACPC Conference Including Special Tracks on Parallel Numerics and Parallel Computing in Image Processing, Video Processing, and Multimedia, pp. 96–106. Springer-Verlag, London, UK (1999)
19. Schmid, W., Uhl, A.: Techniques for parallel quasi-Monte Carlo integration with digital sequences and associated problems. Mathematics and Computers in Simulation **55**(1–3), 249–257 (2001)
20. Schwarz, H., Köckler, N.: Numerische Mathematik. 6. überarb. Auflage, Vieweg+Teubner (2008)
21. Sobol', I.: On the Distribution of points in a cube and the approximate evaluation of integrals. Zh. vychisl. Mat. mat. Fiz. **7**(4), 784–802 (1967)
22. Wächter, C.: Quasi-Monte Carlo Light Transport Simulation by Efficient Ray Tracing. Ph.D. thesis, Universität Ulm (2008)
23. Zaremba, S.: La discrépance isotrope et l'intégration numérique. Ann. Mat. Pura Appl. **87**, 125–136 (1970)

Quasi-Monte Carlo Progressive Photon Mapping

Alexander Keller, Leonhard Grünschloß, and Marc Droske

Abstract The simulation of light transport often involves specular and transmissive surfaces, which are modeled by functions that are not square integrable. However, in many practical cases unbiased Monte Carlo methods are not able to handle such functions efficiently and consistent Monte Carlo methods are applied. Based on quasi-Monte Carlo integration, a deterministic alternative to the stochastic approaches is introduced. The new method for deterministic consistent functional approximation uses deterministic consistent density estimation.

1 Introduction

Photorealistic image synthesis aims at simulating the process of taking photographs. In principle, such simulations sum up the contributions of all transport paths which connect light sources with sensors.

An obvious approach to numerical simulation are bidirectional path tracing algorithms, where random walk methods are used to generate paths from the sensors and lights in order to connect them (as illustrated in Fig. 1). However, there are common situations, where establishing such connections by checking visibility using so-called shadow rays can be arbitrarily inefficient.

As an example, one might think of light entering a car through a window, hitting the interior, and being transported back through the window to an outside

A. Keller · M. Droske
NVIDIA ARC GmbH, Berlin, Germany
e-mail: keller.alexander@gmail.com; marc.droske@gmail.com

L. Grünschloß
Rendering Research Weta Digital, New Zealand
e-mail: leonhard@gruenschloss.org

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_28, © Springer-Verlag Berlin Heidelberg 2012

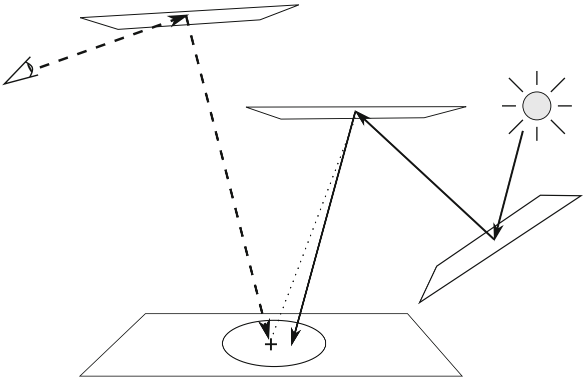


Fig. 1 Bidirectional generation of light transport paths: a path started from the eye (*dashed rays*) and a path started from a light source (*solid rays*) can be connected by a shadow ray (*dotted line*), which checks whether the vertices to connect are mutually visible. Alternatively, the basic idea of photon mapping is to relax the precise visibility check by allowing for a connection of both paths if their end points are sufficiently close as indicated by the circle.

observer. A similarly difficult situation is the observation of a room through a mirror (see Fig. 2), where substantial illumination of the room is due to a small light source through the mirror, too. Such light transport paths cannot be established efficiently, because the direction of the connecting shadow ray has to coincide with the direction of a reflection on the mirror, which in fact happens with probability zero. In the context of bidirectional path tracing this problem has been characterized as the problem of “insufficient techniques” [13, Fig. 2].

Key to efficiency is a shift of paradigm: Instead of considering unbiased Monte Carlo algorithms, allowing for a certain bias that vanishes in the limit opens up a new class of more efficient algorithms. Such algorithms are called consistent.

In computer graphics, photon mapping [7] has been developed in order to deal with the problem of “insufficient techniques”. While in its first formulation, the technique was consistent only within infinite memory, progressive photon mapping [4] was introduced as an algorithm that converges pointwise within finite memory. In a follow-up article [3], a stochastic algorithm was derived that converges globally. Both latter publications provide example implementations. In [12] the stochastic arguments have been simplified, resulting in a simplified algorithm as well.

In contrast to previous work, which we detail in the next section, we introduce a deterministic photon mapping algorithm and prove its convergence. The method is based on sampling path space using quasi-Monte Carlo methods [14], which on the average allow for faster convergence as compared to Monte Carlo methods [20]. As a consequence of the deterministic nature, parallelization is simplified and results can be exactly reproduced even in a heterogeneous computing environment [11].

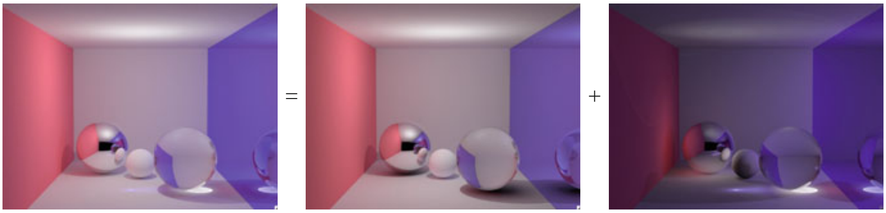


Fig. 2 The complete simulation of light transport (*left*) is the sum of light transported by square integrable surface properties (*middle*) and light transported by surfaces, whose physical properties are described by Dirac distributions (*right*). Unbiased Monte Carlo algorithms fail in robustly simulating the light transport path from the point light source in the center of the ceiling to the mirror onto the floor back to mirror into the camera. Consistent photon mapping efficiently can handle such paths. Caustics, i.e., the focal pattern underneath the glass ball, are another example of such paths.

2 Background on Photon Mapping

The principles of light transport simulation by tracing light transport paths are depicted in Fig. 1. The basic idea of bidirectional path tracing [18] is to start paths from both the eye and the light source in order to connect them. Connecting the end points of both paths requires to check their mutual visibility using a shadow ray. As mentioned in the introduction, this technique can be arbitrarily inefficient if the reflection properties of at least one of the end points are not square integrable.

In this quite common situation, it is helpful to give up on precise visibility. Instead of tracing a shadow ray, end points are connected unless they are too distant. Photon mapping algorithms implement this principle by first tracing the trajectories of photons p that are emitted from the light sources and storing the incident energy, direction, and location, whenever a photon interacts with a surface. In a second step, paths are traced from the eye, whose contribution is determined by estimating the radiance [7, Sect. 7.2, Eq. 7.6]

$$L(x,\omega) \approx \frac{1}{\pi r^2} \sum_{p \in \mathscr{B}_x} f_s(\omega, x, \omega_p) \Delta\Phi_p \tag{1}$$

in the end point x of each eye path. The approximation is computed as an average over the area of a disk of radius r (see the circle in Fig. 1), where the incident flux $\Delta\Phi_p$ attenuated by the bidirectional scattering distribution function (BSDF) f_s of all photons p in the ball $\mathscr{B}_x$ with respect to the query point x is summed up. The direction ω is the direction from which x is observed, while ω_p is the direction of incidence of the photon p.

Originally, the radius r had been selected as the radius of the ball enclosing the k closest photons around the query point x, which formed the set $\mathscr{B}_x$ [7]. Thus the radius was large in sparsely populated regions and small in regions of high photon density. In practice, this choice can result in numerical problems, because in high photon density regions the radius can approach zero arbitrarily close and thus the term $\frac{1}{\pi r^2}$ cannot be bounded. The resulting overmodulation usually is not perceived, as it appears in bright regions anyhow.

A new class of algorithms that progressively refine the solution has been introduced in [4]. While the radius remains related to photon density in the aforementioned sense, each query location has its own radius, which is decreased upon photon interaction. As a consequence the radius decreases faster in brighter regions and may remain unchanged if shadowed, which, however, does not affect convergence. This approach has been reformulated in [12, Sect. 4.3] such that no longer local statistics are required.

Since effects like perfect mirror reflection or refraction are modeled by Dirac-δ distributions, which are not square-integrable, they should not be part of the numerical evaluation of the reflective or refractive surface properties f_s. Instead, whenever such a component is encountered during tracing a path, Russian roulette is used to either terminate or prolong the path by simulating the perfect reflection or refraction, respectively [15]. Thus in practice the unbounded parts of f_s are never evaluated.

3 Pointwise Consistent Density Estimation

Photon mapping is strongly related to density estimation, where the radius is called smoothing parameter or smoothing length [16, Sect. 3.4]. Proofs of the consistency of density estimation [16, Sects. 3.7.1 and 3.7.2] are based on choosing the radius in reciprocal dependence on a polynomial in the number n of emitted particles. Early work on photon mapping did not establish this reciprocal relationship and therefore only allowed for plausibility arguments [7, Sect. 7.2, Eq. 7.7]. Recent work [3,4,12] implicitly includes the reciprocal relationship, which allowed for more profound derivations.

In the following, a simpler and more general argument to prove

$$L(x,\omega) = \lim_{n\to\infty} \frac{1}{\pi \cdot r^2(x,n)} \sum_{p\in\mathscr{B}(x,r(x,n))} f_s(\omega,x,\omega_p)\Delta\Phi_p, \tag{2}$$

where $\mathscr{B}(x,r(x,n))$ is the set of all photons in the ball of radius $r(x,n)$ around the point x, is derived by explicitly choosing a squared radius

$$r^2(x,n) := \frac{r_0^2(x)}{n^\alpha} \qquad \text{for } 0 < \alpha < 1 \tag{3}$$

that includes the reciprocal dependence on a power of the number n of emitted photons. The explicit dependence on the query location x allows for choosing an initial radius $r_0(x) > 0$ in dependence of an initial photon distribution, similar to [12, Sect. 4.3] and early photon mapping work.

The radiance estimator

$$L_n(x,\omega) := \frac{n^\alpha}{n \cdot \pi \cdot r_0^2(x)} \sum_{p \in \mathscr{B}(x,r(x,n))} f_s(\omega, x, \omega_p)\phi_p \tag{4}$$

results from including the number n of emitted photons in the photon flux $\Delta\Phi_p := \frac{\phi_p}{n}$ and inserting it into Eq. 2.

The radiance estimator can be generalized by using any other kernel that in the limit results in a Dirac-δ distribution [12]. Such kernels, other than the characteristic function of the set $\mathscr{B}(x, r(x,n))$, are found in [16] or in the domain of smoothed particles hydrodynamics (SPH). In analogy with the SPH approach, using the derivative of such a kernel allows one to compute irradiance gradients.

3.1 Choice of the Parameter α

For $n > 1$ we have $\frac{n^\alpha}{n} < 1$ due to the postulate $0 < \alpha < 1$. As a consequence L_n will always be bounded, because the evaluation of f_s is bounded as established at the end of Sect. 2.

Since light transport is a linear problem, the number of photons in $\mathscr{B}(x, r(x,n))$ asymptotically must be linear in n: For $\alpha = 1$ doubling the number n of emitted photons results in half the squared radius, meaning half the area, while the number of photons in $\mathscr{B}(x, r(x,n))$ remains the same. For $0 < \alpha < 1$ the squared radius decreases slower than the increase in number of photons. As a consequence, more and more photons are collected with increasing n, which guarantees $L(x,\omega) = \lim_{n\to\infty} L_n(x,\omega)$.

Note that Eq. 2 does neither converge for $\alpha = 0$, because the initial radius will not be decreased, nor for $\alpha = 1$ as the noise level does not decrease. This can be easily verified by running the algorithm with either one of the extremal values. Comparing the graphs of $\frac{n^\alpha}{n}$ for the two extreme cases reveals that $\alpha = \frac{1}{2}$ in fact best balances the two interests of fast convergence and noise reduction. However, this choice is not crucial at all as shown in the next section.

3.2 Choice of the Initial Radius r_0

The limit of the ratio of the $(n+1)$-st and n-th squared radius reveals that the squared radius is vanishing arbitrarily slowly:

$$\lim_{n\to\infty} \frac{r^2(x,n+1)}{r^2(x,n)} = \lim_{n\to\infty} \frac{n^\alpha}{(n+1)^\alpha} = \lim_{n\to\infty} \left(\frac{n}{n+1}\right)^\alpha = 1$$

As a consequence, a larger value of α is only effective for smaller n and therefore the initial radius r_0 becomes most influential. However, competing goals need to be satisfied: In favor of efficiency, a smaller radius requires less photons to be averaged, while on the other hand a larger radius allows for more efficient high frequency noise reduction by averaging more photons.

While a local initial radius allows for adapting to the photon density and thus a better trade-off between noise and smoothing, it requires the retrieval of $r_0(x)$ [12, Sect. 4.3]. For example $r_0(x)$ can be obtained from an initial set of photons in analogy to the original photon mapping algorithm. Alternatively, an individual radius can be stored for each functional, for example for each pixel to be computed. If in addition $r_0(x)$ can be bounded efficiently, for example by determining its maximum, the efficiency of nearest neighbor search can be improved.

Of course the simplest choice is a global initial radius r_0, which we prefer to choose rather smaller than larger, as the human visual system is more comfortable with high frequency noise than blotchy low-frequency averaging artifacts.

4 Consistent Functional Approximation

In fact, Eq. 4 can be considered an integro-approximation problem: Given one set of photons generated by n paths started at the light sources, the radiance L_n is defined for any location x and any direction ω.

This allows one to compute the color

$$\begin{aligned} L_P &:= \lim_{m\to\infty} \frac{|P|}{m} \sum_{q=0}^{m-1} \chi_P(x_q) W(x_q) L(h(x_q), \omega(x_q)) \\ &= \lim_{m\to\infty} \frac{|P|}{m} \sum_{q=0}^{m-1} \chi_P(x_q) W(x_q) \lim_{n\to\infty} L_n(h(x_q), \omega(x_q)) \qquad (5) \\ &\approx \frac{|P|}{mn} \sum_{q=0}^{m-1} \chi_P(x_q) W(x_q) \frac{n^\alpha}{\pi \cdot r_0^2(h(x_q))} \qquad (6) \\ &\qquad \cdot \sum_{p \in \mathscr{B}(h(x_q), r(h(x_q), n))} f_s(\omega(x_q), h(x_q), \omega_p) \phi_p \end{aligned}$$

of a pixel P using an infinite sequence of uniform samples x_q to determine query locations $h(x_q)$: The x_q define eye paths, where $W(x_q)$ is the accumulated weight along the path, which is multiplied by the radiance $L(h(x_q), \omega(x_q))$. The paths associated with the pixel P are selected by the characteristic function χ_P, while $|P|$ is the area of pixel P.

Computing the functional (5) requires the enumeration of all pairs of indices (q, p) of query paths and photons (see Fig. 3). This way each query location $h(x_q)$

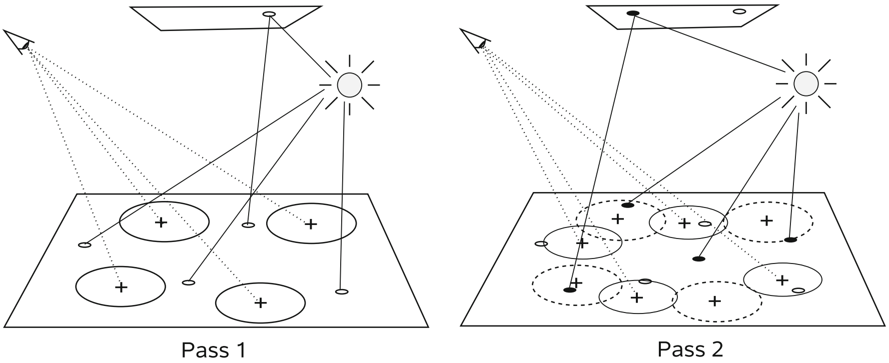


Fig. 3 Just taking two deterministic point sequences to enumerate all eye and light paths in passes can fail: in the illustration, the photons of the second pass would interact with the query locations of the first pass, however, these interactions never become enumerated as pairs. A black image results, although light is transported to the eye. While such illustrative situations can be constructed artificially, they occur in a practical setting as well.

can interact with all photons, which guarantees the pointwise convergence of Eq. 4 and consequently the approximation (6) is consistent.

4.1 Algorithm

As derived in the previous section, each query location must interact with all photons, which requires the enumeration of all pairs of query path and light path indices. Therefore $\mathbb{N}_0 \times \mathbb{N}_0$ is partitioned into contiguous blocks of m_b query location indices times n_b light path indices. The ratio $\frac{m_b}{n_b}$ allows for balancing pixel anti-aliasing and photon density. The blocks are enumerated using the index i along the dimension of query paths and j along the dimension of light paths.

Obviously it is most efficient to keep as many query locations and photons as possible in memory. However, as an unavoidable consequence of finite memory, both query locations and photons need to be recomputed. This excess amount of computation depends on the order of how the blocks are enumerated. While the diagonal order in Fig. 4a requires permanent recomputation, the rectangular order in Fig. 4b allows for frequent reuse of either the set of query locations or the set of photons. Such space filling curves are easily implemented, even with direct block access, which allows for parallelization without communication or synchronization [11].

The rigid partition into blocks of equal size can be avoided by generating query locations and photons until a given block of memory is filled. The resulting starting indices m_i and n_j for the query locations and light paths, respectively, are stored in an array each in order to allow for the direct retrieval of the i-th range $m_i, \dots, m_{i+1} - 1$ of query paths and the j-th range $n_j, \dots, n_{j+1} - 1$ of light paths.

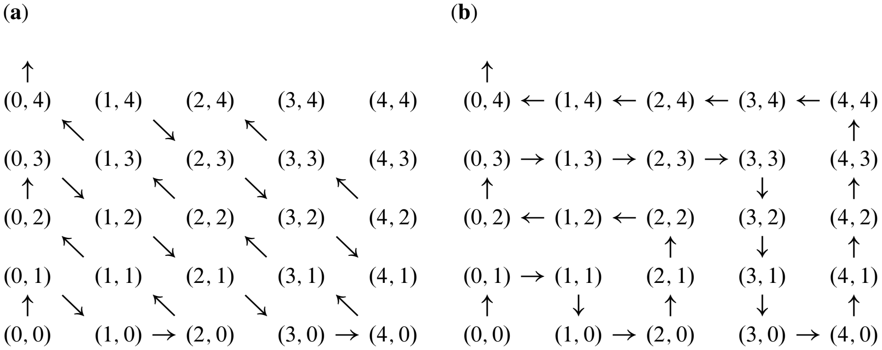


Fig. 4 Enumerating all combinations of integers in the (**a**) classic diagonal order used for enumerating the rational numbers and (**b**) an order that results in much better data coherence and caching.

4.2 Consistency of Block-Wise Computation

If the number n of light paths can be fixed in advance, the radius $r(x, n)$ will be used throughout the computation and the sums will be weighted by $\frac{1}{mn}$ as shown in approximation (6).

If the ultimate value of n is unknown, the computation will have to be conducted in a progressive way. The weight for the intermediate results then amounts to the reciprocal of the number of currently processed blocks multiplied by $m_b n_b$, which is the number processed pairs of query points and light paths.

A block with light path block index j is processed using the radius $r(x, j \cdot n_b)$. Note that the algorithm (5) remains consistent, because the weight of each single summand decreases with increasing number of blocks. As j increases, less photons interact with the query locations, since the query radius decreases (see Fig. 5). This can be interpreted as slight blur that sharpens with the progress of the computation. As the radius decreases arbitrarily slow (see Sect. 3.2), this effect is hardly visible, which again emphasizes that the choice of the initial radius is much more important than the overall progression of the radius.

4.3 Deterministic Sampling using Quasi-Monte Carlo Points

Pseudo-random number generators in fact are deterministic algorithms that try to mimic random numbers. However, the approximate independence of pseudo-random numbers is no longer visible once the samples are averaged. More important, the speed of convergence depends on the uniformity of the samples. In that

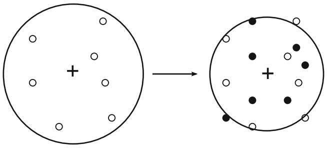


Fig. 5 The radius shrinks with the number of photons blocks enumerated, while at the same time the contribution of the individual photons that were collected with larger radius fades out.

respect deterministic low discrepancy sequences are preferable, as they are more uniform as compared to random samples and therefore improve the speed of convergence [6,8,10,14]. Finally, such deterministic algorithms are simple to parallelize in heterogeneous computing environments and results are exactly reproducible [11].

Interpreting the radiance estimator (4) as an integro-approximation problem allows for applying the results of [10, Theorem 1], which guarantee that generating the photons using deterministic low discrepancy sequences [14] results in a consistent deterministic estimation with all the aforementioned advantages.

With the point-wise convergence of the radiance estimate established, any consistent quadrature rule can be used for sampling the query location paths. Especially, the same low discrepancy sequence as used for photon generation can be applied, which simplifies implementations.

Constructions of low discrepancy sequences are found in [14]. The images in Fig. 2 have been computed using the Halton sequence. We also verified the theoretical results using fast implementations of (t, s)-sequences in base b, especially the Sobol' sequence [17, 19], and rank-1 lattices sequences [5].

Note that the point sequences must be dimension extensible in order to account for potentially infinite length transport paths, which in theory would rule out rank-1 lattices and constructions similar to the Faure sequences [1]. However, due to finite precision, path length cannot be infinite on a computer and it is reasonable and acceptable to limit path length by a sufficiently large bound. While in theory this leads to inconsistent results, in practice the resulting bias is not observable in most cases.

For the sake of completeness, we note that the algorithms derived in this article work with any point sequence that is uniform, i.e., has vanishing discrepancy. This includes random, pseudo-random, or randomized point sequences such as for example randomized low discrepancy sequences.

Samples of uniform sequences can be transformed to path space samples using approaches explained in detail in [18]. We therefore only point out that the paths resulting in the query points are generated by sampling the whole image plane or tiles thereof instead of sampling on a pixel-by-pixel basis. While it is possible to simply map the image plane or tiles thereof to the unit square, it may be preferable to directly map pixels to sample indices [2, 9, 10].

5 Results and Discussion

Figure 2 shows a classic test scene, where the algorithm was used to simulate light transport completely and only in parts, especially caustics. The derived method has been proven to unconditionally converge and can be used as an efficient substitute for other photon mapping implementations.

Other than in Eq. 5, the derivation of stochastic progressive photon mapping [3] does not allow all query locations to interact with all photons. While it is still possible to argue that stochastic progressive photon mapping is converging as long as random sampling is used, the algorithm cannot be derandomized by just using deterministic samples, because then it is possible to construct scenarios that do not converge (see Fig. 3). If for example the camera is used as light source at the same time, query paths and light paths are identical and therefore perfectly correlated. As path space is not uniformly sampled, visible illumination reconstruction artifacts, like for example overmodulation, become visible.

6 Conclusion

We introduced quasi-Monte Carlo progressive photon mapping. Based on the principles of enumerating all pairs of non-negative integers, convergence has been proven for the deterministic case.

The simple derivation and algorithmic principle enable the deterministic and consistent computation of many more linear problems as for example all kinds of (bidirectional) path tracing, in which query and light paths are connected by shadow rays. If path space sampling is extended to consider participating media, the proposed schemes generalize to volume scattering as well [12, Sect. 4.2].

Acknowledgements This work has been dedicated to Jerry Spanier's 80th birthday.

References

1. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. **41**(4), 337–351 (1982)
2. Grünschloß, L., Raab, M., Keller, A.: Enumerating quasi-Monte Carlo point sequences in elementary intervals. In: H. Woźniakowski, L. Plaskota (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 399–408 in this volume. Springer (2012)
3. Hachisuka, T., Jensen, H.: Stochastic progressive photon mapping. In: SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers, pp. 1–8. ACM, New York, NY, USA (2009)
4. Hachisuka, T., Ogaki, S., Jensen, H.: Progressive photon mapping. ACM Transactions on Graphics **27**(5), 130:1–130:8 (2008)
5. 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)

6. Hlawka, E., Mück, R.: Über eine Transformation von gleichverteilten Folgen II. Computing **9**, 127–138 (1972)
7. Jensen, H.: Realistic Image Synthesis Using Photon Mapping. AK Peters (2001)
8. Keller, A.: Quasi-Monte Carlo Methods for Photorealistic Image Synthesis. Ph.D. thesis, University of Kaiserslautern, Germany (1998)
9. Keller, A.: Strictly Deterministic Sampling Methods in Computer Graphics. SIGGRAPH 2003 Course Notes, Course #44: Monte Carlo Ray Tracing (2003)
10. Keller, A.: Myths of computer graphics. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 217–243. Springer (2006)
11. Keller, A., Grünschloß, L.: Parallel quasi-Monte Carlo methods. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 489–500 in this volume. Springer (2012)
12. Knaus, C., Zwicker, M.: Progressive photon mapping: A probabilistic approach. ACM Transactions on Graphics (TOG) **30**(3) (2011)
13. Kollig, T., Keller, A.: Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In: H. Niederreiter, K. Fang, F. Hickernell (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 290–305. Springer (2002)
14. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
15. Shirley, P.: Realistic Ray Tracing. AK Peters, Ltd. (2000)
16. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC (1986)
17. Sobol', I.: Uniformly Distributed Sequences with an additional Uniform Property. Zh. vychisl. Mat. mat. Fiz. **16**(5), 1332–1337 (1976)
18. Veach, E.: Robust Monte Carlo Methods for Light Transport Simulation. Ph.D. thesis, Stanford University (1997)
19. Wächter, C.: Quasi-Monte Carlo Light Transport Simulation by Efficient Ray Tracing. Ph.D. thesis, Universität Ulm (2008)
20. Woźniakowski, H.: Average case complexity of multivariate integration. Bull. Amer. Math. Soc. **24**, 185–194 (1991)

Value Monte Carlo Algorithms for Estimating the Solution to the Coagulation Equation

Mariya Korotchenko

Abstract The pure coagulation Smoluchowski equation with additive coefficients is considered. We construct the weight value algorithms and analyze their efficiency for estimating total monomer concentration as well as total monomer and dimer concentration in ensemble governed by the equation under study. We managed to achieve considerable gain in computational costs via approximate value modeling of the time between collisions in the ensemble combined with the value modeling of the interacting pair number.

1 Introduction

In this paper we develop value modifications of statistical simulation for the approximate solution to the Smoluchowski equation, which describes a wide class of coagulation processes in physical systems. In spatially homogeneous case it has the following form:

$$\frac{\partial n_l(t)}{\partial t} = \frac{1}{2} \sum_{i+j=l} K_{ij} n_i(t) n_j(t) - \sum_{i \geq 1} K_{il} n_i(t) n_l(t), \tag{1}$$

where

- $n_l(t)$ is an average number of l-sized particles at the instant t;
- Particle size l is a positive integer;
- K_{ij} are the coagulation coefficients, which are supposed to be given.

M. Korotchenko
Institute of Computational Mathematics and Mathematical Geophysics (Siberian Branch of the Russian Academy of Sciences), prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia
e-mail: kmaria@osmf.sscc.ru

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_29, © Springer-Verlag Berlin Heidelberg 2012

Adding to (1) the initial data

$$n_l(0) = n_0(l), \;\; l > 0, \tag{2}$$

we obtain a Cauchy problem for the nonlinear Smoluchowski equation. In this article we will estimate linear functionals of the function $n_l(t)$.

Solution to a kinetic equation can be numerically estimated using simulation of a homogeneous Markov chain, which describes the evolution of the N-particle system [2,5]. Note, that transitions in this Markov chain are due to elementary pair interactions (collisions).

Let us introduce the following notations:

- N_0 is the initial number of particles in the system, be given at time $t = 0$;
- l_i is the size of the particle with number i;
- $k(l_i, l_j \to l) = N_0^{-1} K_{l_i,l_j} \delta_{l_i+l_j,l}$;
- $N \leq N_0$ is the current number of particles in the system;
- $\varpi = (i, j)$ is the interacting pair number responsible for a collision in the system;
- $A_S(X) = \sum\limits_{i=1}^{N-1} \sum\limits_{j=i+1}^{N} a(N, l_i, l_j)$, where $a(1, l_i, l_j) \equiv 0$, while for $N > 1$ we have $a(\varpi) \equiv a(N, l_i, l_j) = \sum\limits_{l=1}^{\infty} k(l_i, l_j \to l)$;
- $X = (N, L_N) = (N, l_1, \ldots, l_N)$ describes the phase state of the system;
- $P(X, t)$ is set of probabilities, which determines the state distribution of the system at the instant t.

Note that under the molecular chaos assumption one can obtain in the limit (see [7] for details)

$$n_l^*(t) \equiv \frac{1}{N_0} \sum_{N=1}^{\infty} \sum_{l_2=1}^{\infty} \cdots \sum_{l_N=1}^{\infty} NP(N, l, l_2 \cdots, l_N, t) \to n_l(t), \text{ when } N_0 \to \infty.$$

The interaction density of the system $\phi(X, t) = A_S(X)P(X, t)$ satisfies the integral N-particle Kolmogorov-type equation, as it is shown in [7]. However, it is impossible to construct the standard weight modifications of the statistical simulation using this equation, because its kernel is the sum of mutually singular terms.

The weight modifications developed further are based on the technique suggested in [6]. The authors of that paper suggest to modify the phase space by introducing the pair number ϖ to the set of the phase coordinates. This allowed to derive a special integral equation for the function $F(X, \varpi, t) = a(\varpi)P(X, t)$ in the transformed phase space $\mathbf{Z} \times [0, T]$:

$$F(Z, t) = \int\limits_0^t \int\limits_{\mathbf{Z}} F(Z', t')K(Z', t' \to Z, t)\, \mathrm{d}Z'\, \mathrm{d}t' + F_0(Z)\delta(t).$$

Here the following notations are used: $Z = (X, \varpi)$, $\mathrm{d}Z = \mathrm{d}X \mathrm{d}\mu_0(\varpi)$. Integration with respect to the measure μ_0 implies summation over all various pairs $\varpi = (i, j)$, and integration over $\mathrm{d}X$ means summation over all the values of N and L_N. The latter equation can be used to construct standard weight modifications of statistical simulation for the many-particle system due to multiplicative structure of its kernel $K(Z', t' \to Z, t) = K_1(t' \to t|X')K_2(\varpi|X')K_3(X' \to X|\varpi)$. Here the distribution density of the time between elementary interactions (collisions) is exponential:

$$K_1(t' \to t|X') = A_{\mathrm{S}}(X') \exp\{-A_{\mathrm{S}}(X')(t - t')\}.$$

The probability that a pair of particles $\varpi = (i, j)$ is responsible for a collision in the N-particle system is $K_2(\varpi|X') = a'(\varpi)/A_{\mathrm{S}}(X')$. Finally, the function $K_3(X' \to X|\varpi)$ defines the transformation of the system after a collision of the pair ϖ, which results in replacement of interacting particles i and j by a single particle of the size $l = l_i + l_j$, so $N = N' - 1$.

The following functionals of the particle flux $A_{\mathrm{S}}^{-1}(X)\phi(X, T)$ are usually of interest:

$$J_H(T) = \int H(X)A_{\mathrm{S}}^{-1}(X)\phi(X, t)\, \mathrm{d}X.$$

For $\tilde{H}(X, t) = H(X) \exp\{-A_{\mathrm{S}}(X)t\}$, $H(X) \in L_\infty$, the following equality was derived [6]:

$$J_H(T) = \int_0^T \int_{\mathbf{Z}} \tilde{H}(X, T - t')F(Z, t')\, \mathrm{d}Z\, \mathrm{d}t' = (F, \tilde{H}),$$

which we will make use of later. In this paper we are interested in estimating the monomer concentration, i.e. the functional $J_{H_1}(T)$ with

$$H_1(X) = H(N, l_1, \ldots, l_N) = \sum_{i=1}^{N} \delta(l_i - 1)/N_0,$$

as well as the monomer and dimer concentration, i.e. the functional $J_{H_{12}}(T)$ with

$$H_{12}(X) = \sum_{i=1}^{N} [\delta(l_i - 1) + \delta(l_i - 2)]/N_0.$$

2 Value Simulation for Smoluchowski Equation

In the following section we suggest the value simulation algorithms applied to estimation of the functionals $J_{H_1}(T)$ and $J_{H_{12}}(T)$.

Define another Markov chain $\{Z_n, t_n\}$, $n = 0, 1, \ldots, \nu$; $\nu = \max\limits_n\{n : t_n < T\}$ with a transition density $P^*(Z', t' \to Z, t) = P_1(t' \to t|X')P_2(\varpi|X')P_3(X' \to X|\varpi)$ and a distribution density $P_0(Z)\delta(t)$ of the initial state (Z_0, t_0).

Let us define random weights by the formulas [8]

$$Q_0 = \frac{F_0(Z)}{P_0(Z)}, \quad Q_n = Q_{n-1}Q(Z_{n-1}, t_{n-1}; Z_n, t_n);$$

$$Q(Z', t'; Z, t) = Q_t \cdot Q_\varpi \cdot Q_X = \frac{K_1(t' \to t|X')}{P_1(t' \to t|X')}\frac{K_2(\varpi|X')}{P_2(\varpi|X')}\frac{K_3(X' \to X|\varpi)}{P_3(X' \to X|\varpi)}.$$

In order to estimate the functional $J_H(T)$, the "weight" collision estimator ξ can be used (see [8]):

$$\xi = \sum_{i=0}^{\nu} Q_n \tilde{H}(X_n, T - t_n).$$

Further we will use the following theorem.

Theorem 1 ([6]). *If $P_0(Z) \neq 0$ for $F_0(Z) \neq 0$ and $Q(Z', t'; Z, t) < +\infty$ for $Z', Z \in \mathbf{Z}$ and $t', t < T$, then $\mathbf{E}\xi = J_H(T)$. Moreover, if the weights are uniformly bounded and $H \in L_\infty$, then there exists such T^* that $\mathbf{V}\xi < +\infty$ whenever $T < T^*$.* □

To minimize the variance of the estimator ξ, we suggest to use a "value" simulation, i.e. to choose a better transition density $P^*(Z', t' \to Z, t)$ with the help of the value function. Value function F^* is the solution of the conjugate integral equation $F^* = K^*F^* + H$. Moreover, it is known (see e.g. [8]) that if we simulate the Markov chain according to the probability densities, proportional to the value function, i.e. $P^* \sim K \cdot F^*$ and $P_0 \sim F_0 \cdot F^*$, then $\mathbf{V}\xi = 0$. Since the value function is usually unknown, we should use an approximation of it. To construct this approximation, we can take into account any a priori information about the form of the value function and use it to improve our simulation. For the problem under consideration we can use the following theorem.

Theorem 2 ([3]). *Let N_1' and N_1 be the number of monomers in ensemble before the next collision and after one, respectively. If the mean value of N_1 is proportional to N_1', then the value function is proportional to N_1.*

This theorem is valid for constant coefficients $K_{ij} = 1$, as well as for additive ones $K_{ij} = (i + j)/2$. Note, that for $N_1 + N_2$ (where N_2 stands for the number of dimers in the system) the hypothesis of the theorem is approximately true, i.e. $\mathbf{E}(N_1 + N_2) = A \cdot (N_1' + N_2') + \delta$ with $\delta = \mathscr{O}(N_0^{-1})$. In this case we will also apply the theorem and use an approximation of the value function, which is proportional to $N_1 + N_2$. The formula for $\mathbf{E}(N_1 + N_2)$ for the coagulation coefficients considered in the paper is given in Sect. 2.3.

In the sequel we describe the construction of the value algorithm for the case of additive coefficients $K_{ij} = (i + j)/2$. In this case

$$a(N, l_i, l_j) = \frac{l_i + l_j}{2N_0}, \quad A_{\rm S}(X) = \frac{N-1}{2}.$$

The simulation process of the next collision in the Markov chain includes two successive elementary transitions:

1. The choice of the next instant of collision (the time interval between collisions);
2. The choice of the next interacting pair number in the system.

In the same order we construct the value simulation algorithm. For the first elementary transition we use a value probability density $P_1(t' \to t|X')$ to simulate the time between collisions. Then we calculate value probabilities $P_2(\varpi|X')$ to choose two particles for collision. Let us now describe each of these transitions in detail.

2.1 Modeling of the Time Between Collisions

For the first elementary transition we use the "value" modeling of the time interval between collisions. We suggest to use an exponential approximation to the time value function, which was obtained in [4] for a simpler problem (1) with the constant coagulation coefficients. For estimating the functionals of interest we have

$$P_1(t' \to t|X') = \mathrm{I}_\varepsilon(t) \frac{(A_{\rm S}(X') - A_1)\exp\{-(A_{\rm S}(X') - A_1)(t - t')\}}{1 - \exp\{-(A_{\rm S}(X') - A_1)(T_\varepsilon - t')\}},$$

where

- $T_\varepsilon = T + \varepsilon$, ε is the extension length of the time interval in which our Markov chain is constructed;
- $\mathrm{I}_\varepsilon(t)$ is the indicator of the time interval $[0, T_\varepsilon]$.

This extension of the time interval is necessary in case of the value simulation to terminate the Markov chain. In the value algorithm, theoretically we sample t within the interval (t', T), but it is impossible numerically: we will never stop since the probability to reach T in a finite number of steps is equal to zero. That is why we extend the time interval $[0, T]$ by a small value ε (to show this fact we introduce the indicator function $\mathrm{I}_\varepsilon(t)$), and stop simulation when $t > T$, i.e. we assume that $\tilde{H}(X, T - t) \equiv 0$ for $t > T$. This extension does not make the functional estimator biased, but it insignificantly affects the variance and computation time.

The random weight Q_t has the following form:

$$Q_t = \frac{SA_{\rm S}(X')}{A_{\rm S}(X') - A_1}\exp\{-A_1(t - t')\}, \quad S = 1 - \exp\{-(A_{\rm S}(X') - A_1)(T_\varepsilon - t')\}.$$

For the case of estimating the monomer concentration we have

$$A_1 = \frac{2}{2+T_\varepsilon},$$

for the case of estimating the monomer and dimer concentration we have

$$A_1 = \frac{2T_\varepsilon + 1}{(2+T_\varepsilon)(1+T_\varepsilon)}.$$

Next elementary transition in simulation process is the **Value Modeling of the Interacting Pair Number** (**VMIPN**). This stage depends on the type of the functional and is described below.

2.2 VMIPN to Estimate the Monomer Concentration

Let us denote N' to be the total number of particles, and N_1' to be the number of monomers in the ensemble at the end of the free path of the system (before the choice of ϖ). In this case $H(X) = N_1/N_0$.

The VMIPN algorithm suggested below aims at preservation of the monomers in the simulated ensemble. It results in a better estimation of the average amount of monomers at the time instant T.

Each of all possible interacting pairs falls into one of the non-overlapping subsets. The choice of the subset depends on the change in the number of monomers, which will take place after the collision: $\varpi_1 \cup \varpi_2 \cup \varpi_0$. The number of monomers may decrease by 1 as a result of the 'minus-1-pairs' collision, by 2 – which results from collision of 'minus-2-pairs', or it may not change in case of interaction of 'minus-0-pairs' from the subset ϖ_0:

ϖ_1 contains 'minus-1-pairs' of the form $\{$*monomer, multimer*$\}$; there are $\mathscr{N}_1$ pairs of this type:

$$\mathscr{N}_1 = N_1'(N' - N_1');$$

ϖ_2 contains 'minus-2-pairs' of the form $\{$*monomer, monomer*$\}$; there are $\mathscr{N}_2$ pairs of this type:

$$\mathscr{N}_2 = N_1'(N_1' - 1)/2;$$

ϖ_0 contains 'minus-0-pairs' of the form $\{$*multimer, multimer*$\}$; there are $\mathscr{N}_0$ pairs of this type:

$$\mathscr{N}_0 = (N' - N_1')(N' - N_1' - 1)/2.$$

Note, here by a multimer we understand an l-sized particle, where $l \geq 2$.

Further, we introduce a representation of the "physical" distribution density $\mathscr{P}_0(i,j) = \dfrac{a(l_i, l_j)}{A_S(X')}$ of the interacting pair number in the following randomized form:

$$1 \equiv \sum_{\varpi} \mathscr{P}_0(i,j) = p_1 \sum_{\varpi_1} f_1(i,j) + p_2 \sum_{\varpi_2} f_2(i,j) + p_0 \sum_{\varpi_0} f_0(i,j). \tag{3}$$

Here p_{m} is the probability to choose the subset ϖ_{m}, and $f_{\mathrm{m}}(i,j)$ is the probability to choose the pair (i,j) from the subset ϖ_{m}, m $= 0, 1, 2$:

$$p_2 = \frac{N_1'(N_1'-1)}{N_0(N'-1)}, \quad p_1 = \frac{N_1'(N'-2N_1'+N_0)}{N_0(N'-1)}, \quad p_0 = 1 - p_1 - p_2. \tag{4}$$

The monomers are chosen uniformly within the subsets ϖ_1 and ϖ_2. Multimers are chosen within subsets ϖ_0 and ϖ_1 by their "physical" probabilities $\mathscr{P}_j$, $j = N_1' + 1, \ldots, N'$, having the following form:

- For pairs from ϖ_1: $\mathscr{P}_j = \dfrac{1 + l_j}{N' - 2N_1' + N_0}$;
- For pairs from ϖ_0: $\mathscr{P}_j = \dfrac{(N_0 - N_1') + l_j(N' - N_1' - 2)}{2(N_0 - N_1')(N' - N_1' - 1)}$.

In order to "preserve" the monomers, let us carry out the simulation according to (3) by replacing probabilities p_{m} from (4) by the probabilities q_{m}, proportional to the number of monomers left in the system:

$$q_1 = \frac{p_1\left(N_1'-1\right)}{C_m}, \quad q_2 = \frac{p_2\left(N_1'-2\right)}{C_m}, \quad q_0 = \frac{p_0 N_1'}{C_m},$$
$$C_m = \mathbf{E}(N_1) = N_1' \frac{N_0 - 1}{N_0} \frac{N' - 2}{N' - 1}.$$

Such modification is taken into account, when the weight is calculated:

$$Q_{\varpi} = \frac{p_{\mathrm{m}}}{q_{\mathrm{m}}}.$$

2.3 VMIPN to Estimate the Monomer and Dimer Concentration

Let us denote N_2' to be the number of dimers in the ensemble at the end of the free path of the system. In this case $H(X) = (N_1 + N_2)/N_0$. We introduce the distribution density proportional to the quantity $N_1(X) + N_2(X)$ in order to introduce VMIPN. Further on we will refer to the l-sized particle with $l \geq 3$ as a multimer.

Taking this into account, let us split the set of all possible interacting pairs into six non-overlapping subsets. The choice of the subset is related to the change in the total number of monomers and dimers, which results from the collision (this quantity may decrease by 1, 2, or not change): $\varpi_{11} \cup \varpi_{1k} \cup \varpi_{2k} \cup \varpi_{22} \cup \varpi_{12} \cup \varpi_{kk}$, where

ϖ_{11} contains 'minus-1-pairs' of the form {*monomer, monomer*}; there are $\mathscr{N}_{11}$ pairs of this type:

$$\mathscr{N}_{11} = N_1'(N_1' - 1)/2;$$

ϖ_{1k} contains 'minus-1-pairs' of the form {*monomer, multimer*}; there are $\mathscr{N}_{1k}$ pairs of this type:

$$\mathscr{N}_{1k} = N_1'(N' - (N_1' + N_2'));$$

ϖ_{2k} contains 'minus-1-pairs' of the form {*dimer, multimer*}; there are $\mathscr{N}_{2k}$ pairs of this type:

$$\mathscr{N}_{2k} = N_2'(N' - (N_1' + N_2'));$$

ϖ_{22} contains 'minus-2-pairs' of the form {*dimer, dimer*}; there are $\mathscr{N}_{22}$ pairs of this type:

$$\mathscr{N}_{22} = N_2'(N_2' - 1)/2;$$

ϖ_{12} contains 'minus-2-pairs' of the form {*monomer, dimer*}; there are $\mathscr{N}_{12}$ pairs of this type:

$$\mathscr{N}_{12} = N_1'N_2';$$

ϖ_{kk} contains 'minus-0-pairs' of the form {*multimer, multimer*}; there are $\mathscr{N}_{kk}$ pairs of this type:

$$\mathscr{N}_{kk} = \frac{(N' - (N_1' + N_2'))(N' - (N_1' + N_2') - 1)}{2}.$$

Let us represent the "physical" distribution density $\mathscr{P}_0$ of the interacting pair number in the form, similar to (3)

$$\begin{aligned} 1 \equiv \sum_{\varpi} \mathscr{P}_0(i,j) = p_{11} \sum_{\varpi_{11}} f_{11}(i,j) + p_{1k} \sum_{\varpi_{1k}} f_{1k}(i,j) + p_{2k} \sum_{\varpi_{2k}} f_{2k}(i,j) + \\ + p_{12} \sum_{\varpi_{12}} f_{12}(i,j) + p_{22} \sum_{\varpi_{22}} f_{22}(i,j) + p_{kk} \sum_{\varpi_{kk}} f_{kk}(i,j), \end{aligned} \tag{5}$$

where p_{mn} is the probability to choose the subset ϖ_{mn}, and $f_{\mathrm{mn}}(i,j)$ is the probability to choose the pair (i,j) from the subset ϖ_{mn}, $\mathrm{m}, \mathrm{n} \in \{1, 2, k\}$.

Note that

$$\begin{aligned} p_{12} &= \frac{3N_1'N_2'}{N_0(N'-1)}, \quad p_{1k} = \frac{N_1'(N' - 2N_1' - 3N_2' + N_0)}{N_0(N'-1)}, \\ p_{22} &= \frac{2N_2'(N_2'-1)}{N_0(N'-1)}, \quad p_{2k} = \frac{N_2'(2N' - 3N_1' - 4N_2' + N_0)}{N_0(N'-1)}, \\ p_{11} &= \frac{N_1'(N_1'-1)}{N_0(N'-1)}, \quad p_{kk} = \frac{(N_0 - N_1' - 2N_2')(N' - N_1' - N_2' - 1)}{N_0(N'-1)}. \end{aligned} \tag{6}$$

Monomers and dimers are chosen uniformly within the subsets ϖ_{11}, ϖ_{1k}, ϖ_{2k}, ϖ_{22} and ϖ_{12}. The multimers are chosen within the subsets ϖ_{1k}, ϖ_{2k} and ϖ_{kk} according to "physical" probabilities $\mathscr{P}_j$, $j = N_1' + N_2' + 1, \ldots, N'$, which have the following form:

- For a pair from the subset ϖ_{1k}: $\mathscr{P}_j = \dfrac{1 + l_j}{N' - 2N_1' - 3N_2' + N_0}$;
- For a pair from the subset ϖ_{2k}: $\mathscr{P}_j = \dfrac{2 + l_j}{2N' - 3N_1' - 4N_2' + N_0}$;
- For a pair from the subset ϖ_{kk}: $\mathscr{P}_j = \dfrac{(N_0 - N_1' - 2N_2') + l_j(N' - N_1' - N_2' - 2)}{2(N_0 - N_1' - 2N_2')(N' - N_1' - N_2' - 1)}$.

In order to "preserve" the monomers and dimers, we will choose the interacting pair according to (5) with the probabilities (6) replaced by q_{mn}, proportional to the sum of the monomers and dimers left in the system:

$$q_{11} = (N_1' + N_2' - 1)\frac{p_{11}}{C_{md}};\ \ q_{1k} = (N_1' + N_2' - 1)\frac{p_{1k}}{C_{md}};\ \ q_{2k} = (N_1' + N_2' - 1)\frac{p_{2k}}{C_{md}};$$

$$q_{12} = (N_1' + N_2' - 2)\frac{p_{12}}{C_{md}};\ \ q_{22} = (N_1' + N_2' - 2)\frac{p_{22}}{C_{md}};\ \ q_{kk} = (N_1' + N_2' - 0)\frac{p_{kk}}{C_{md}},$$

where

$$C_{md} = \mathbf{E}(N_1 + N_2) = (N_1' + N_2')\frac{(N_0 - 2)(N' - 2)}{N_0(N' - 1)} + \frac{N_1'(N' + N_1' - 3)}{N_0(N' - 1)}.$$

This modification is taken into consideration, when the weight is calculated:

$$Q_\varpi = \frac{p_{\mathrm{mn}}}{q_{\mathrm{mn}}}.$$

2.4 Direct Method Vs. Value Algorithm

In this section we give a description of direct and value simulation algorithms, which were used to perform numerical experiments.
The direct simulation method

1. For $t_0 = 0$ we sample the initial state Z_0 according to the probability density $F_0(Z)$; $N = N_0$.
2. Then for a given state (Z_{n-1}, t_{n-1}) we choose the next state (Z_n, t_n) in the following manner:

   a. We choose t_n according to the density $K_1(t_{n-1} \to t | X_{n-1})$;
   b. We sample two particles $\varpi = (i, j)$ for collision according to $K_2(\varpi | X_{n-1})$;
   c. We modify the ensemble: $l_j = l_i + l_j$, $l_i = 0$; $N_n = N_{n-1} - 1$.

3. If $t_n < T$ then we repeat step 2, otherwise we calculate $H(X_n)$ and terminate the chain.

The value simulation algorithm

1. For $t_0 = 0$ we sample the initial state Z_0 according to the probability density $P_0(Z)$; $Q_0 = F_0(Z_0)/P_0(Z_0)$; $N = N_0$.
2. Then for a given state (Z_{n-1}, t_{n-1}) we choose the next state (Z_n, t_n) in the following manner:

   a. We choose t_n according to the density $P_1(t_{n-1} \to t|X_{n-1})$;
   b. We sample two particles $\varpi = (i, j)$ for collision according to $P_2(\varpi|X_{n-1})$;
   c. We modify the ensemble: $l_j = l_i + l_j$, $l_i = 0$; $N_n = N_{n-1} - 1$.

3. If $t_n < T$ then the summand $Q_{n-1} \cdot Q_t \cdot Q_\varpi \cdot \tilde{H}(X_n, T - t_n)$ is calculated, otherwise the chain terminates.

It appears that the value algorithm uses information about the trajectory for the whole interval $[0, T]$, while the direct method uses only one value $H(X)$ per trajectory.

3 Results of the Numerical Experiments

In this section the results of simulation according to the suggested algorithms are presented and compared to the analytic solution for the test problem. As a test problem for implementation of the algorithms described above, we take the Cauchy problem (1)–(2) with $K_{ij} = (i + j)/2$, $n_0(l) = \delta_{l,1}$. This problem has an exact solution of the form (see [1])

$$n_l(t) = \mathrm{e}^{-0.5t} B\left(1 - \mathrm{e}^{-0.5t}, l\right), \quad B\,(x, l) = \frac{(lx)^{l-1}\mathrm{e}^{-lx}}{l!}, \quad l \geq 1.$$

In numerical experiments parameter $\varepsilon = 10^{-5}$ is used. It leads to an almost minimal variance (with respect to ε) and does not increase the average number of interactions in the system much, as compared to the direct simulation method. We used the following notations in the tables:

- $\bar{\sigma}$ is the mean square error (square root of the corresponding variance, which is estimated simultaneously with the functional);
- PE $(\%)$ is the percent error;
- $t^{(c)}$ is the computation time;
- M is the number of simulated trajectories;
- $\mathrm{S}_d = \bar{\sigma}_d^2 t_d^{(c)}$ and $\mathrm{S}_v = \bar{\sigma}_v^2 t_v^{(c)}$ are the computational costs for the direct and value simulations respectively.

When analyzing the numerical results we should mention, that the deterministic error of order $\mathscr{O}(N_0^{-1})$ occurs due to the finiteness of the initial number of

Table 1 Estimation of $J_{H_1}(T)$ ($T = 1; 5; 10; 15$; $M = 10^3$; $N_0 = 10^3$).

Simulation	$\tilde{J}_{H_1}(T)$	$\bar{\sigma}$	PE (%)	t_c	$\mathrm{S}_d/\mathrm{S}_v$
$T = 1,\ n_1(1) = 4.09234 \cdot 10^{-1}$					
Direct	$4.09075 \cdot 10^{-1}$	$6.2 \cdot 10^{-4}$	0.04	1.5	–
Value	$4.09627 \cdot 10^{-1}$	$9.1 \cdot 10^{-5}$	0.10	1.7	**41.4**
$T = 5,\ n_1(5) = 3.27807 \cdot 10^{-2}$					
Direct	$3.28220 \cdot 10^{-2}$	$1.8 \cdot 10^{-4}$	0.13	3.3	–
Value	$3.28776 \cdot 10^{-2}$	$5.4 \cdot 10^{-5}$	0.30	3.7	**9.76**
$T = 10,\ n_1(10) = 2.49551 \cdot 10^{-3}$					
Direct	$2.45300 \cdot 10^{-3}$	$4.9 \cdot 10^{-5}$	1.70	3.4	–
Value	$2.51194 \cdot 10^{-3}$	$1.7 \cdot 10^{-5}$	0.65	3.8	**6.96**
$T = 15,\ n_1(15) = 2.03581 \cdot 10^{-4}$					
Direct	$2.26000 \cdot 10^{-4}$	$1.5 \cdot 10^{-5}$	11.0	3.4	–
Value	$2.07266 \cdot 10^{-4}$	$3.9 \cdot 10^{-6}$	1.81	3.9	**12.7**

Table 2 Estimation of $J_{H_{12}}(T)$ ($T = 1; 5; 10; 15$; $M = 10^3$; $N_0 = 10^3$).

Simulation	$\tilde{J}_{H_{12}}(T)$	$\bar{\sigma}$	PE (%)	t_c	$\mathrm{S}_d/\mathrm{S}_v$
$T = 1,\ n_1(1) + n_2(1) = 5.17876 \cdot 10^{-1}$					
Direct	$5.18496 \cdot 10^{-1}$	$6.1 \cdot 10^{-4}$	0.12	1.2	–
Value	$5.18513 \cdot 10^{-1}$	$2.1 \cdot 10^{-4}$	0.12	1.6	**6.81**
$T = 5,\ n_1(5) + n_2(5) = 4.47971 \cdot 10^{-2}$					
Direct	$4.46120 \cdot 10^{-2}$	$2.2 \cdot 10^{-4}$	0.41	2.3	–
Value	$4.48527 \cdot 10^{-2}$	$9.2 \cdot 10^{-5}$	0.12	2.9	**4.27**
$T = 10,\ n_1(10) + n_2(10) = 3.41354 \cdot 10^{-3}$					
Direct	$3.43000 \cdot 10^{-3}$	$5.9 \cdot 10^{-5}$	0.48	2.4	–
Value	$3.42655 \cdot 10^{-3}$	$2.6 \cdot 10^{-5}$	0.38	3.1	**4.10**
$T = 15,\ n_1(15) + n_2(15) = 2.78474 \cdot 10^{-4}$					
Direct	$2.55000 \cdot 10^{-4}$	$1.6 \cdot 10^{-5}$	8.43	2.4	–
Value	$2.84914 \cdot 10^{-4}$	$5.3 \cdot 10^{-6}$	2.31	3.1	**7.01**

particles N_0, e.g. for the monomer concentration we have: $|n_1(T) - J_{H_1}(T)| = \mathcal{O}(N_0^{-1})$ (see [7] for details). The statistical error has the following form $|J_{H_1}(T) - \tilde{J}_{H_1}(T)| = \mathcal{O}(\bar{\sigma})$ and it is of order $\mathcal{O}(M^{-1/2})$. From the tables we can see, that the value modeling decreases the computational cost of simulation (several times, as compared to the direct one), which shows advantages of the value simulation for both elementary transitions simultaneously (Tables 1 and 2).

4 Conclusion

We constructed value algorithms for estimating the total monomer concentration, as well as the total monomer and dimer concentration in ensembles governed by the coagulation Smoluchowski equation. We succeeded to reduce the computational costs with the help of combining the value modeling of the time between collisions and the value modeling of the interacting pair number.

In future we plan to apply introduced methodology to the case of linear coefficients $K_{ij} = a + b(i + j)$. These coefficients are of practical use, e.g. in one of classical polymer models. Moreover, the problem of optimal choice of parameter ε, which depends on T, N_0, H, needs further consideration.

Acknowledgements The author acknowledges the kind hospitality of the Warsaw University and the MCQMC'2010 conference organizers. The author would also like to thank Prof. Gennady Mikhailov and Dr. Aleksandr Burmistrov for valuable discussions.

This work was partly supported by Russian Foundation for Basic Research (grants 09-01-00035, 09-01-00639, 11-01-00252) and SB RAS (Integration Grant No. 22).

References

1. Aldous, D.J.: Deterministic and stochastic models for coalescence (agregation and coagulation): a review of the mean-field theory for probabilists. Bernoulli, **5** (1), 3–48 (1999)
2. Gillespie, D.T.: The stochastic coalescence model for cloud droplet growth. J. Atmos. Sci. **29** (8), 1496–1510 (1972)
3. Korotchenko, M.A., Mikhailov, G.A., Rogasinsky, S.V.: Modifications of weighted Monte Carlo algorithms for nonlinear kinetic equations. Comp. Math. Math. Phys. **47** (12), 2023–2033 (2007)
4. Korotchenko, M.A., Mikhailov, G.A., Rogasinsky, S.V.: Value modifications of weighted statistical modeling for solving nonlinear kinetic equations. Russ. J. Numer. Anal. Math. Modelling, **22** (5), 471–486 (2007)
5. Lushnikov, A.A.: Some new aspects of coagulation theory. Izv. Akad. Nauk SSSR, Ser. Fiz. Atmosfer. i Okeana, **14** (10), 738–743 (1978) [in Russian]
6. Mikhailov, G.A., Rogasinsky, S.V.: Weighted Monte Carlo methods for approximate solution of the nonlinear Boltzmann equation. Sib. Math. J. **43** (3), 496–503 (2002)
7. Mikhailov, G.A., Rogasinsky, S.V., Ureva, N.M.: Weighted Monte Carlo methods for an approximate solution of the nonlinear coagulation equation. Comp. Math. Math. Phys. **46** (4), 680–690 (2006)
8. Mikhailov, G.A., Voitishek, A.V.: Numerical Statistical Modelling (Monte Carlo Method). Akademia, Moscow (2006) [in Russian]

Numerical Simulation of the Drop Size Distribution in a Spray

Christian Lécot, Moussa Tembely, Arthur Soucemarianadin, and Ali Tarhini

Abstract Classical methods of modeling predict a steady-state drop size distribution by using empirical or analytical approaches. In the present analysis, we use the maximum of entropy method as an analytical approach for producing the initial data; then we solve the coagulation equation to approximate the evolution of the drop size distribution. This is done by a quasi-Monte Carlo simulation of the conservation form of the equation. We compare the use of pseudo-random and quasi-random numbers in the simulation. It is shown that the proposed method is able to predict experimental phenomena observed during spray generation.

1 Introduction

The disintegration of bulk fluid into droplets in a surrounding gas (referred to as atomization) is extensively developed and applied to a variety of industrial processes. Jet printing technology has a broad range of utilization in areas such as biotechnology, pharmacology, electronic printing or fuel cell manufacturing. In certain applications, the drop size distribution must have particular form, and constitutes one of the most important spray characteristics.

C. 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

M. Tembely · A. Soucemarianadin
Université de Grenoble, Laboratoire des Écoulements Géophysiques et Industriels, UMR 5519 UJF & CNRS & INPG, 1023 rue de la Piscine, Domaine universitaire, 38400 Saint Martin d'Hères, France
e-mail: Moussa.Tembely@ujf-grenoble.fr; Arthur.Soucemarianadin@ujf-grenoble.fr

A. Tarhini
Département de Mathématiques, Université Libanaise, Faculté des Sciences I, Campus universitaire, Hadath – Beyrouth, Liban

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_30, © Springer-Verlag Berlin Heidelberg 2012

One method for modeling drop size distribution is empirical: given a collection of "standard" distributions, a form is located that fits the data collected for a range of atomizers. A problem with the empirical approach is the difficulty of extrapolating the data to regimes outside the experimental range [1]. As an alternative to this approach, the maximum entropy formalism (MEF) determines the most likely drop size distribution as the one that maximizes an entropy function under a set of physical constraints: we refer to the recent review [5].

Both approaches are essentially time-independent. But the drop size distribution may change at varying distances from the atomizer nozzle. Some measurements of drop diameters show distributions with two peaks, that the previous modeling does not clearly explain [2, 6]. In the present analysis, we model the evolution of droplet size distribution. First MEF is used for approximating the initial distribution. The time-dependent distribution is then computed as the solution of the coagulation equation [3].

We solve the equation with a Monte Carlo (MC) simulation. Particles are sampled from the initial drop size distribution, time is discretized and the sizes are changed according to the coagulation equation. If particles simulate directly drops, they may coalesce, so the total number will decrease. Hence the system has to be enriched to make the results statistically reliable. Here we solve the conservation form of the coagulation equation, so that the number of particles in the simulation remains constant.

A drawback of usual MC methods is their low convergence rate. In certain cases, it is possible to improve it by replacing the pseudo-random numbers used to simulate the i.i.d. random variables by quasi-random numbers. This is the basis of quasi-Monte Carlo (QMC) methods [11]. In the present case, each step of the simulation is formulated as a numerical integration and we find it necessary to sort the particles by increasing size before performing a QMC quadrature [8, 9]: we are then able to prove the convergence of the method.

For the application, we focus on an innovative spray on demand (SOD) technology, where spray is generated only if required (in contrast with a continuous jetting device) [15]. A modeling of the spray is carried out, and the drop size distribution can be computed, which paves the way for optimization of the atomizer.

The outline of the paper is as follows. In Sect. 2, we derive the conservation form of the coagulation equation, we describe the simulation schemes and we analyze the convergence of the QMC algorithm. In Sect. 3, we present the SOD device, we compute the drop size distribution for a given operating condition and we compare MC and QMC approaches. Conclusions are drawn in Sect. 4.

2 Simulation of the Coagulation Equation

The representation of drop size distribution is an important issue in liquid atomization. Several types of functions may be defined. The droplets ensemble can be subdivided into subsets, where each subset consists of drops whose diameter are in a given interval: by counting the number of drops in each subset, one constructs the frequency histogram with respect to diameter. The continuous version of the histogram is the number-based diameter density $f_n(D) \geq 0$. It is convenient to assume that the drop diameters range from zero to infinity, hence we have $\int_0^{+\infty} f_n(D)dD = 1$. It is also possible to construct a number-based volume density $g_n(V)$. Assuming the droplets are spherical, $g_n(V) = 2f_n(D)/(\pi D^2)$. If we consider the increment of volume in each class, we define the volume-based diameter density $f_v(D) := \pi \mathscr{N} D^3 f_n(D)/(6\mathscr{V})$, where $\mathscr{N}$ is the total number and $\mathscr{V}$ is the total volume of droplets. The volume-based volume density is $g_v(V) := \mathscr{N} V g_n(V)/\mathscr{V}$.

In the following, we use time-dependent quantities. The coagulation equation for $\mathscr{N}(t)g_n(V,t)$ is [3]:

$$\frac{\partial}{\partial t}(\mathscr{N}(t)g_n(V,t)) = \frac{1}{2}\int_0^V K_c(V-W,W)\mathscr{N}(t)g_n(V-W,t)\mathscr{N}(t)g_n(W,t)dW$$
$$-\int_0^{+\infty} K_c(V,W)\mathscr{N}(t)g_n(V,t)\mathscr{N}(t)g_n(W,t)dW, \qquad (1)$$

with the initial condition $g_n(V,0) = g_{n,0}(V)$, where $g_{n,0}(V)$ is a given density. Here $K_c(V,W)$ is the coagulation kernel describing the rate of coalescence between two drops of volume V and W to form one drop of volume $V+W$. The kernel is assumed to be nonnegative and symmetric. The total number of droplets $\mathscr{N}(t)$ tends to decrease over time due to coalescence, while the total volume $\mathscr{V}(t)$ remains unchanged.

By multiplying Eq. 1 by $V/\mathscr{V}$, we obtain the following conservation form:

$$\frac{\partial g_v}{\partial t}(V,t) = \int_0^V \widetilde{K}_c(V-W,W)g_v(V-W,t)g_v(W,t)dW$$
$$-\int_0^{+\infty} \widetilde{K}_c(V,W)g_v(V,t)g_v(W,t)dW, \qquad (2)$$

where $\widetilde{K}_c$ is the modified coagulation kernel defined by: $\widetilde{K}_c(V,W) := \mathscr{V} K_c(V,W)/W$. We denote by $g_{v,0}(V)$ the initial volume-based volume density.

We introduce a weak formulation of Eq. 2. We denote by $\mathscr{M}^+$ the set of all measurable functions $\sigma : (0,+\infty) \to [0,+\infty)$. By multiplying Eq. 2 by $\sigma \in \mathscr{M}^+$ and by integrating, we obtain:

$$\frac{d}{dt}\int_0^{+\infty} g_v(V,t)\sigma(V)dV =$$
$$\int_0^{+\infty}\int_0^{+\infty} \widetilde{K}_c(V,W)g_v(V,t)g_v(W,t)(\sigma(V+W)-\sigma(V))dWdV. \qquad (3)$$

2.1 The QMC Algorithm

We propose a QMC scheme for the numerical simulation of Eq. 2. We recall from [11] some basic notations and concepts of QMC methods. Let $s \geq 1$ be a fixed dimension and $I^s := [0,1)^s$ be the s-dimensional unit cube and λ_s be the s-dimensional Lebesgue measure. For a set $U = \{\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}\}$ of N points in I^s and for a measurable set $B \subset I^s$ we define the local discrepancy by

$$D_N(B, U) := \frac{1}{N} \sum_{0 \leq k < N} 1_B(\mathbf{u}_k) - \lambda_s(B),$$

where 1_B is the indicator function of B. The star discrepancy of U is $D_N^\star(U) := \sup_{J^\star} |D_N(J^\star, U)|$, where $J^\star$ runs through all subintervals of I^s with a vertex at the origin. For an integer $b \geq 2$, an elementary interval in base b is a set of the form $\prod_{i=1}^s [a_i b^{-d_i}, (a_i + 1)b^{-d_i})$, with integers $d_i \geq 0$ and $0 \leq a_i < b^{d_i}$ for $1 \leq i \leq s$. If $0 \leq t \leq m$ are integers, a (t, m, s)-net in base b is a point set U of b^m points in I^s such that $D_N(J, U) = 0$ for every elementary interval J in base b with measure b^{t-m}. If $b \geq 2$ and $t \geq 0$ are integers, a sequence $\mathbf{u}_0, \mathbf{u}_1, \ldots$ of points in I^s is a (t, s)-sequence in base b if, for all integers $n \geq 0$ and $m > t$, the point set $U^n := \{\mathbf{u}_p : nb^m \leq p < (n+1)b^m\}$ forms a (t, m, s)-net in base b.

We suppose that $\widetilde{K}_c$ is bounded and we set $\widetilde{K}_c^\infty := \sup_{V,W>0} \widetilde{K}_c(V, W)$. The algorithm uses a constant number of $N := b^m$ particles, where $b \geq 2$ and $m \geq 1$ are integers . Time is discretized, using a time step Δt satisfying $\Delta t \widetilde{K}_c^\infty < 1$. We denote $t_n := n\Delta t$. We need a sequence U of quasi-random numbers for the simulation. We assume that U is a $(t, 3)$-sequence in base b (for some $t \geq 0$). In addition, for $n \in \mathbf{N}$, let: $U^n := \{\mathbf{u}_p : nN \leq p < (n+1)N\}$. We assume: that $\pi_{1,2}(U^n)$ is a $(0, m, 2)$-net in base b, where $\pi_{1,2}$ is the projection defined by $\pi_{1,2}(x_1, x_2, x_3) := (x_1, x_2)$.

We first generate a set $\mathrm{V}^0 := \{V_0^0, \ldots, V_{N-1}^0\}$ of N positive numbers – the *particles* – such that the initial volume-based volume probability $g_{v,0}(V)dV$ is approximated by the probability distribution:

$$g_v^0(V) := \frac{1}{N} \sum_{0 \leq k < N} \delta(V - V_k^0),$$

where $\delta(V - V_k)$ is the Dirac delta measure at V_k. This can be done by choosing:

$$V_k^0 = G_{v,0}^{-1}\left(\frac{2k+1}{2N}\right), \quad 0 \leq k < N, \tag{4}$$

where $G_{v,0}$ is the cumulative distribution function: $G_{v,0}(V) := \int_0^V g_{v,0}(W)dW$.

For $n \geq 0$, let $g_{v,n}(V) := g_v(V, t_n)$. We suppose that a set $\mathrm{V}^n = \{V_0^n, \ldots, V_{N-1}^n\}$ of particles has been computed so that

$$g_v^n(V) := \frac{1}{N} \sum_{0 \leq k < N} \delta(V - V_k^n)$$

approximates, in a certain sense (see below), the probability $g_{v,n}(V)dV$. The approximation of the solution at time t_{n+1} is calculated as follows.

Particles are relabeled at the beginning of the time step by increasing size:

$$V_0^n \le V_1^n \le \cdots \le V_{N-1}^n. \tag{5}$$

This type of sorting was used in [8, 9]. It guarantees theoretical convergence: since the algorithm can be described by a series of numerical integration, the sorting reverts to minimizing the amplitude of the jumps of the function to be integrated.

We define a probability measure $\hat{g}_v^{n+1}$ by Euler discretization of Eq. 3:

$$\frac{1}{\Delta t}\left(\int_0^{+\infty} \hat{g}_v^{n+1}(V)\sigma(V) - \int_0^{+\infty} g_v^n(V)\sigma(V)\right) = \\ \int_0^{+\infty}\int_0^{+\infty} \widetilde{K}_c(V,W) g_v^n(V) g_v^n(W)(\sigma(V+W)-\sigma(V)), \tag{6}$$

that is, replacing $g_v^n(x)$ with its expression,

$$\int_0^{+\infty} \hat{g}_v^{n+1}(V)\sigma(V) = \frac{1}{N}\sum_{0\le k<N}\left(1-\frac{\Delta t}{N}\sum_{0\le \ell<N}\widetilde{K}_c(V_k^n, V_\ell^n)\right)\sigma(V_k^n) \\ + \frac{\Delta t}{N^2}\sum_{0\le k,\ell<N}\widetilde{K}_c(V_k^n, V_\ell^n)\sigma(V_k^n+V_\ell^n). \tag{7}$$

The measure $\hat{g}_v^{n+1}$ approximates $g_{v,n+1}(V)dV$, but it is not a sum of Dirac delta measures. We recover this kind of approximation by using a QMC quadrature rule. Let $1_{R_{k,\ell}}$ be the indicator function of $R_{k,\ell} := [k/N, (k+1)/N) \times [\ell/N, (\ell+1)/N)$ and denote by $1_{I_{k,\ell}^n}$ the indicator function of $I_{k,\ell}^n := [0, \Delta t \widetilde{K}_c(V_k^n, V_\ell^n))$. To $\sigma \in \mathscr{M}^+$ corresponds the indicator function:

$$C_\sigma^{n+1}(\mathbf{u}) := \sum_{0\le k,\ell<N} 1_{R_{k,\ell}}(u_1,u_2)\left(\left(1-1_{I_{k,\ell}^n}(u_3)\right)\sigma(V_k^n) + 1_{I_{k,\ell}^n}(u_3)\sigma(V_k^n+V_\ell^n)\right)$$

(for $\mathbf{u} = (u_1, u_2, u_3) \in I^3$), which is such that

$$\int_0^{+\infty} \hat{g}_v^{n+1}(V)\sigma(V) = \int_{I^3} C_\sigma^{n+1}(\mathbf{u})d\mathbf{u}. \tag{8}$$

We determine g_v^{n+1} by a quadrature in I^3, using the nodes U^n:

$$\int_0^{+\infty} g_v^{n+1}(V)\sigma(V) = \frac{1}{N}\sum_{nN\le p<(n+1)N} C_\sigma^{n+1}(\mathbf{u}_p). \tag{9}$$

It is possible to summarize the calculation on a time step as follows. If $u \in I$, let $k(u) := \lfloor Nu \rfloor$. Then, for $nN \le p < (n+1)N$, we have:

$$V^{n+1}_{k(u_{p,1})} = \begin{cases} V^n_{k(u_{p,1})} + V^n_{k(u_{p,2})} & \text{if } u_{p,3} < \Delta t \widetilde{K}_c(V^n_{k(u_{p,1})}, V^n_{k(u_{p,2})}), \\ V^n_{k(u_{p,1})} & \text{otherwise.} \end{cases} \tag{10}$$

The numbers $u_{p,1}$ and $u_{p,2}$ select particles; particle $k(u_{p,1})$ has for coagulation partner particle $k(u_{p,2})$, and the coagulation probability is $P_c := \Delta t \widetilde{K}_c(V^n_{k(u_{p,1})}, V^n_{k(u_{p,2})})$. Then $u_{p,3}$ is used to select an event:

- If $0 \le u_{p,3} < P_c$, particles $k(u_{p,1})$ and $k(u_{p,2})$ coalesce,
- If $P_c \le u_{p,3} < 1$, no coalescence occurs.

The corresponding MC scheme is as follows: there is no reordering of particles and, for $0 \le k < N$,

$$V^{n+1}_k = \begin{cases} V^n_k + V^n_{L_k} & \text{if } U_k < \Delta t \widetilde{K}_c(V^n_k, V^n_{L_k}), \\ V^n_k & \text{otherwise.} \end{cases} \tag{11}$$

Here $L_0, \dots, L_{N-1}$ are independent random samples drawn from the uniform distribution on $\{0, \dots, N-1\}$, while $U_0, \dots, U_{N-1}$ are independent random samples drawn from the uniform distribution on $[0, 1)$.

2.2 Convergence Analysis

We state a convergence result, which proves that the algorithm converges, as the number N of particles grows to infinity; we then show numerical evidence of the convergence in a simple case, where an analytical solution is available, and we compare MC and QMC strategies.

We first adapt the basic tools of QMC methods to our settings. Let g be a probability density on $(0, +\infty)$. If $X > 0$, let σ_X denote the indicator function of $(0, X)$. The *local discrepancy* of the set $\mathrm{V} = \{V_k : 0 \le k < N\} \subset (0, +\infty)$ relative to g is:

$$D_N(X, \mathrm{V}; g) := \frac{1}{N} \sum_{0 \le k < N} \sigma_X(V_k) - \int_0^{+\infty} \sigma_X(V) g(V) dV.$$

The *star discrepancy* of V relative to g is $D^\star_N(\mathrm{V}; g) := \sup_{X>0} |D_N(X, \mathrm{V}; g)|$. We define the *error* of the QMC scheme at time t_n to be the star discrepancy of V^n relative to $g_{v,n}$. The concept of variation of function in the sense of Hardy and Krause may be extended to a function f defined on $(0, +\infty)^s$ and is denoted by $V_{HK}(f)$. The following proposition is an adaptation of the convergence result of [9] and is similarly established. Details of the proof can be found in [13]. Let $T > 0$.

Proposition 1. *We suppose:*

- *For every $V > 0$, the function $t \to g_v(V,t)$ is twice continuously differentiable over $(0,T)$ and g_v, $\frac{\partial g_v}{\partial t}$, $\frac{\partial^2 g_v}{\partial t^2}$ are integrable over $(0,+\infty) \times (0,T)$,*
- *$\widetilde{K}_c$ is of bounded variation in the sense of Hardy and Krause.*

Then, for $t_n \leq T$,

$$D_N^\star(\mathrm{V}^n; g_{v,n}) \leq \mathrm{e}^{c_2 t_n} D_N^\star(\mathrm{V}^0; g_{v,0}) + \Delta t \int_0^{+\infty} \int_0^{t_n} \mathrm{e}^{c_2(t_n - t)} \left| \frac{\partial^2 g_v}{\partial t^2}(V,t) \right| dt dV$$

$$+ \left(\frac{2}{\Delta t} + c_1 \right) \frac{1}{b^{\lfloor (m-t)/3 \rfloor}} \frac{\mathrm{e}^{c_2 t_n} - 1}{c_2},$$

where

$$c_1 := 4V_{HK}(\widetilde{K}_c) + 3\widetilde{K}_c^\infty \quad \textit{and} \quad c_2 := \sup_{V>0} V_{HK}(\widetilde{K}_c(V,.)) + \sup_{W>0} V_{HK}(\widetilde{K}_c(.,W)) + 3\widetilde{K}_c^\infty.$$

The upper bound is of order $\mathscr{O}(1/N^{1/3})$, which is worse than the length of the confidence interval of MC methods, but numerical experiments show that the QMC method converges faster than the corresponding MC scheme.

Now, we assess the accuracy of the QMC algorithm described above and we compare it to the classical MC scheme: approximate solutions are computed in a case where an analytical solution is available [12]. For QMC, the low-discrepancy sequence used is the $(0,3)$-sequence in base 3 of Faure [11]. Let $K_c(V,W) = 1$; with initial condition e^{-V}, the exact solution of Eq. 1 is:

$$\frac{4}{(2+t)^2} \exp\left(-\frac{2V}{2+t} \right). \tag{12}$$

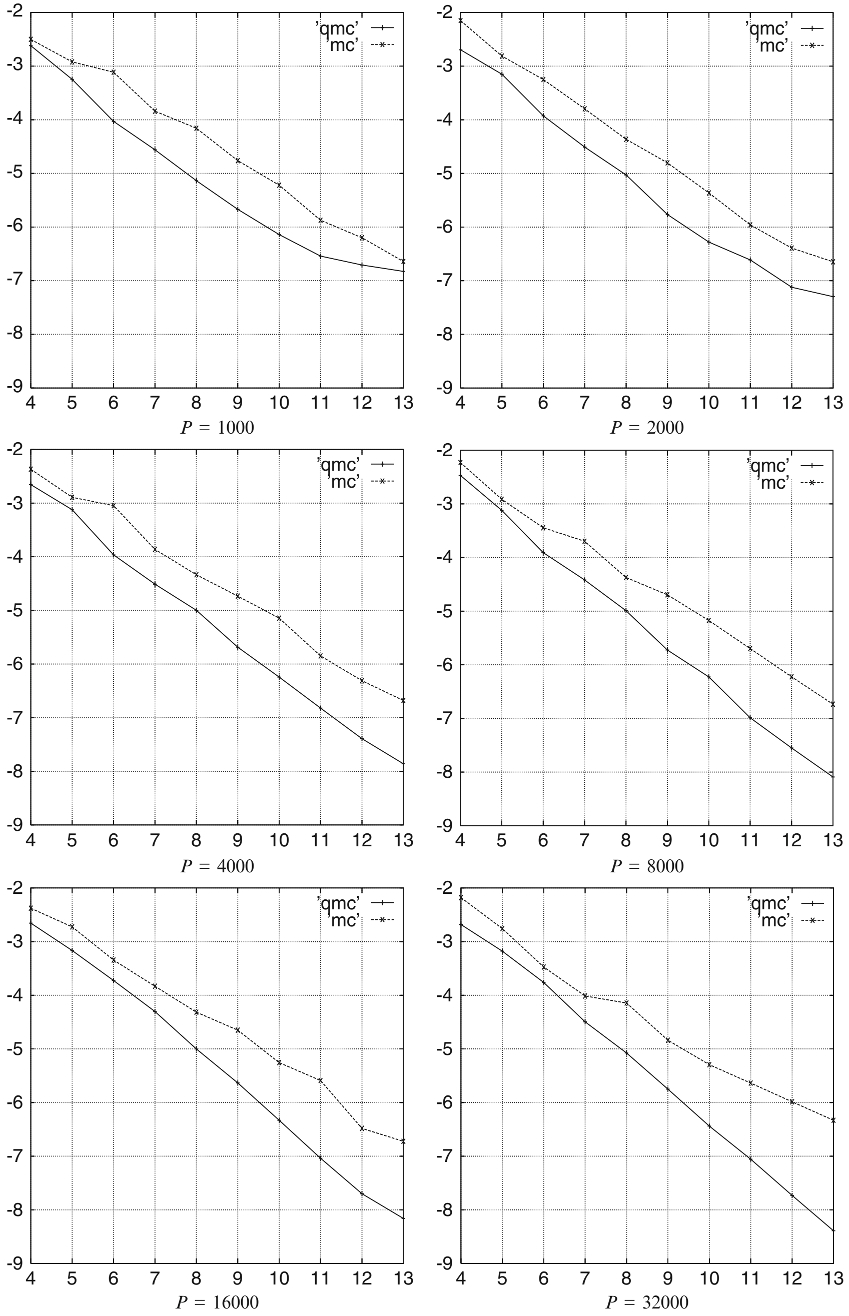
The solution is computed up to time $T = 10.0$ with N particles (N varying between 3^4 and 3^{13}) and P time steps (P varying from $1 \times 1{,}000$ to $2^5 \times 1{,}000$).

In order to reduce scatter, we compute the *averaged discrepancy* defined as:

$$D_{N,P} := \frac{1}{1{,}000} \sum_{h=1}^{1{,}000} D_N^\star(\mathrm{V}^{hp}; g_{v,hp}),$$

where $p := P/1{,}000$. Figure 1 shows log-log plots of $D_{N,P}$ for different values of N and P, for both methods (MC and QMC). For a given number of particles and for a given time step, the error of the QMC scheme is always smaller than the error of the MC scheme; this gain is more effective when both discretization parameters N and P are large enough.

If we assume that the method produces an error of order $\mathscr{O}(N^{-\alpha})$, then the exponent α can be estimated by regression to fit the data, if Δt is sufficiently small, so that the influence of P on the error is negligible versus that of N. If we



P = 1000

P = 2000

P = 4000

P = 8000

P = 16000

P = 32000

Fig. 1 Averaged discrepancy as a function of N (from 3^4 to 3^{13}), for P between 1,000 and 32,000. Log-log plots of QMC (*solid lines*) compared to MC (*dotted lines*) results.

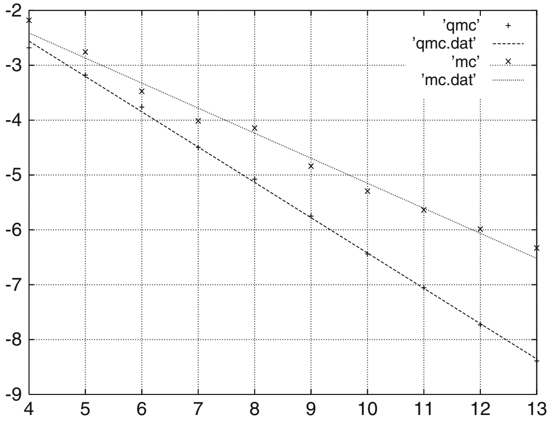


Fig. 2 Linear regression estimates of the averaged discrepancy as a function of N (from 3^4 to 3^{13}) for $P = 32{,}000$. Log-log plots of QMC (*dashed lines*) versus MC (*dotted lines*) outputs.

take $P = 32{,}000$, we find $D_{N,P} = \mathscr{O}(N^{-0.46})$ for MC simulations and $D_{N,P} = \mathscr{O}(N^{-0.64})$ for QMC simulations: see Fig. 2.

Other computations have been done with a linear kernel $K_c(V, W) = V + W$ and the conclusions are the same as in the previous case: we refer to [13] for detailed results.

3 Modeling of SOD Device

In this section we focus on the modeling of the spray generated by a new SOD technology. A microchannel conveying fluid is excited and the drop hanging at the beveled nozzle tip breaks up into droplets: see Fig. 3. A nomenclature for physical quantities is given in Table 1.

Models based on the MEF are used to predict spray diameter density from a small amount of information: the most probable distribution is the one which maximizes the entropy [1, 5]. A new formulation of the MEF is presented in [4]: an additional information is required to limit the production of the small drops. Following Lienhard and Meyer [10], one finally obtains a generalized gamma distribution:

$$f_n(D) = \frac{q}{\Gamma(\frac{\alpha}{q})} \left(\frac{\alpha}{q}\right)^{\alpha/q} \frac{D^{\alpha-1}}{D_{q,0}^{\alpha}} \exp\left(-\frac{\alpha}{q}\left(\frac{D}{D_{q,0}}\right)^{q}\right), \tag{13}$$

where $q > 0$, $\alpha \geq 1$ and $D_{q,0}^{q} = \int_0^{+\infty} D^q f_n(D) dD$. This leads to the following volume-based volume density:

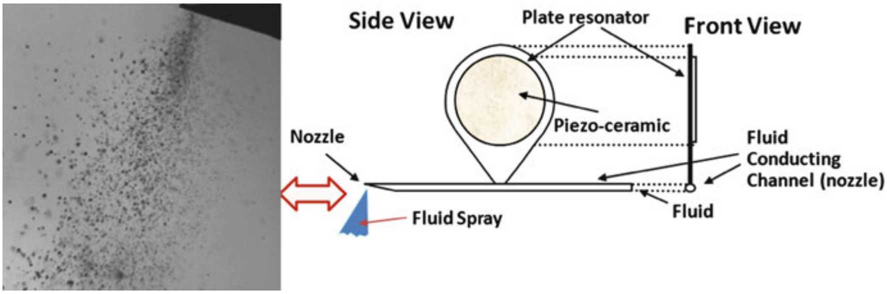


Fig. 3 Spray on demand printhead. The picture on the left-hand side is approximately 2.5 mm long.

Table 1 Nomenclature and values used in the simulation.

Symbol	Name	Value (IS units)
g	Gravity acceleration	9.81
$\alpha'_{\mathcal{V}}$	Spray volume fraction	$5.0\,10^{-4}$
μ_a	Viscosity of air	$18.5\,10^{-6}$
μ_f	Viscosity of fluid	$1.2\,10^{-3}$
ρ_a	Air density	1.0
ρ_f	Fluid density	790.0
σ_f	Fluid surface tension	$22.0\,10^{-3}$

$$g_v(V) = \frac{2}{\pi}\left(\frac{6}{\pi}\right)^{\alpha/3} \frac{q}{\Gamma(\frac{\alpha+3}{q})}\left(\frac{\alpha}{q}\right)^{\frac{\alpha+3}{q}} \frac{V^{\alpha/3}}{D_{q,0}^{\alpha+3}} \exp\left(-\frac{\alpha}{q}\left(\frac{6}{\pi}\right)^{q/3}\frac{V^{q/3}}{D_{q,0}^{q}}\right). \tag{14}$$

This is assumed to describe the spray at the nozzle tip and we take it as an initial condition $g_{v,0}(V)$. The particles are initially sampled by using the inverse transform method: see Eq. 4.

Following [7], we express the coagulation kernel as follows.

$$K_c(V, W) = k_a \lambda_e(V, W) h_f(V, W), \tag{15}$$

where k_a is an adjustable factor, $\lambda_e(V, W)$ is the coalescence efficiency once collision occurs between drops of volume V and W and $h_f(V, W)$ is the collision frequency of drops of volume V and W. The efficiency is defined as the fraction of collisions that result in coalescence, and is given by:

$$\lambda_e(V, W) = \exp\left(-t_{\mathrm{coal}}(V, W)/t_{\mathrm{cont}}(V, W)\right), \tag{16}$$

where $t_{\mathrm{coal}}(V, W)$ is the average coalescence time of drops of volume V and W, while $t_{\mathrm{cont}}(V, W)$ is the contact time for the drops. An estimation of the coalescence time is:

$$t_{\text{coal}}(V,W) = c_c \left(R_{V,W}^3 \rho_f / \sigma_f\right)^{1/2}, \tag{17}$$

where ρ_f is the fluid density, σ_f is the surface tension, c_c is a constant factor; the equivalent radius $R_{V,W}$ is defined by: $1/R_{V,W} := 1/D(V) + 1/D(W)$, with $D(V) := (6V/\pi)^{1/3}$. An expression for the contact time is:

$$t_{\text{cont}}(V,W) = (D(V) + D(W))/(2|\mathbf{u}_r(V,W)|), \tag{18}$$

where $\mathbf{u}_r(V,W)$ is the average relative velocity between drops of volume V and W; the square of velocity may be estimated as follows.

$$|\mathbf{u}_r(V,W)|^2 = u_\ell(V)^2 + u_\ell(W)^2 - 4u_\ell(V)u_\ell(W)/\pi, \tag{19}$$

where $u_\ell(V)$ is the terminal velocity of drops of volume V:

$$u_\ell(V) = \frac{\mu_f + \mu_a}{3\mu_f + 2\mu_a} \frac{D(V)^2 g}{6\mu_a} (\rho_f - \rho_a). \tag{20}$$

Here μ_f is the viscosity of fluid, μ_a is the viscosity of air, ρ_a is the density of air and g is the gravity acceleration. The collision frequency may be expressed in the following form:

$$h_f(V,W) = \pi \left(\frac{D(V)}{2} + \frac{D(W)}{2}\right)^2 |\mathbf{u}_r(V,W)| \frac{\mathscr{N} g_n(V)}{\mathscr{V}_t} \frac{\mathscr{N} g_n(W)}{\mathscr{V}_t} \mathscr{V}_t \tag{21}$$

and we approximate $\mathscr{N} g_n(V)$ by its initial value $\mathscr{N}(0) g_{n,0}(V)$. Here $\mathscr{V}_t$ is the total volume: $\mathscr{V}_t := \mathscr{V}/\alpha'_{\mathscr{V}}$, where $\alpha'_{\mathscr{V}}$ is the spray volume fraction.

We perform the simulation of a SOD device with the following parameters:

$$q = 0.21 \quad \alpha = 25.61 \quad D_{q,0} = 13.39 \quad k_a = 1.0\,10^6 \quad c_c = 1.12$$

The physical data are given in Table 1; the fluid used here is ethanol.

We approximate the number-based diameter density $f_n(D,t)$ up to time $T = 3.0\,10^{-3}$ with $N = 2^{20}$ particles and $P = 600$ time steps. The results are displayed in Figs. 4 and 5. We see that the distribution tends to have two peaks. Similar bimodal distributions were measured [2,6] but no explanation was given for the presence of the small peak. The measurements were done at a certain distance from the atomizer: we think that this peak is due to coalescence of drops, which is simulated here. The results of other experiments and developments are given in [14, 15]. In addition, MC and QMC strategies are compared. For QMC, we use a $(1,3)$-sequence in base 2 of Niederreiter [11]. It is clear that the scattering of the results is reduced when using QMC.

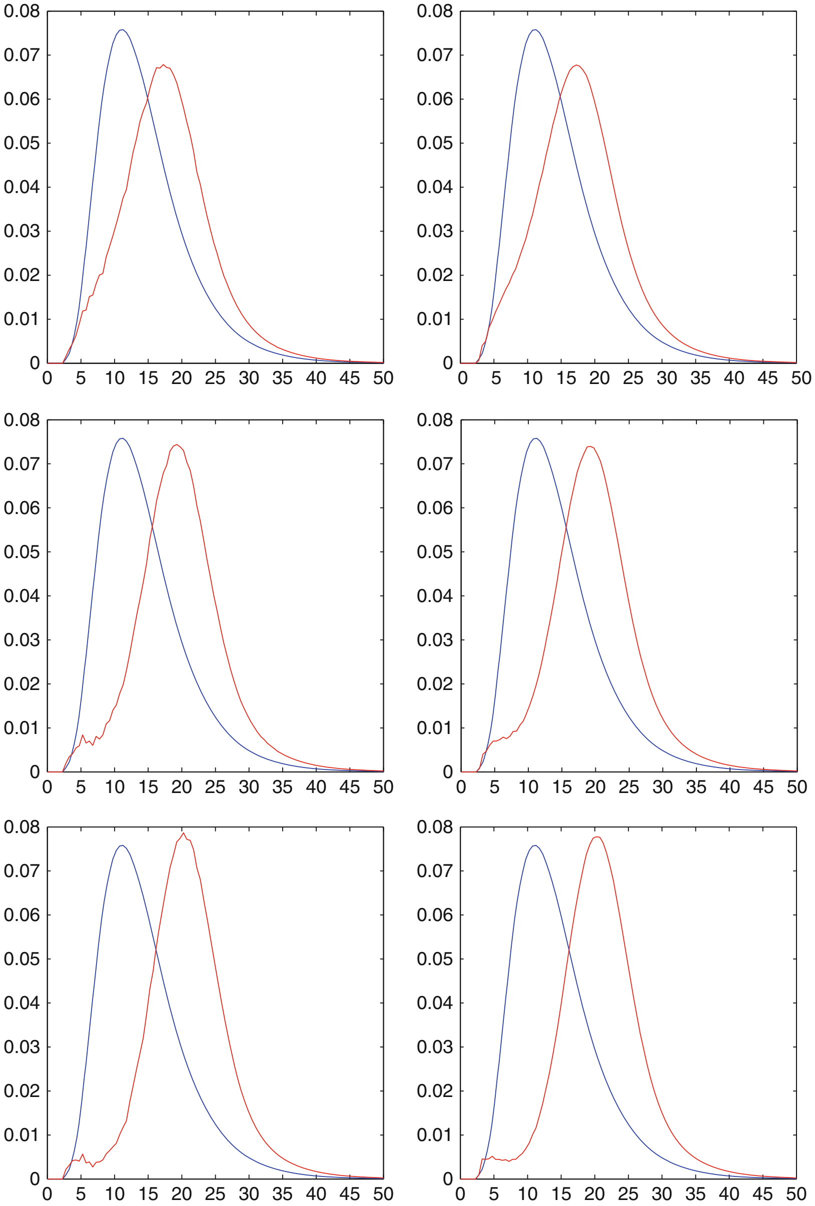


Fig. 4 Simulation of the number-based diameter density: comparison of initial density and density at $t = 0.5$ (*top*), $t = 1.0$ (*middle*), $t = 1.5$ (*bottom*). MC (*left*) versus QMC (*right*) results.

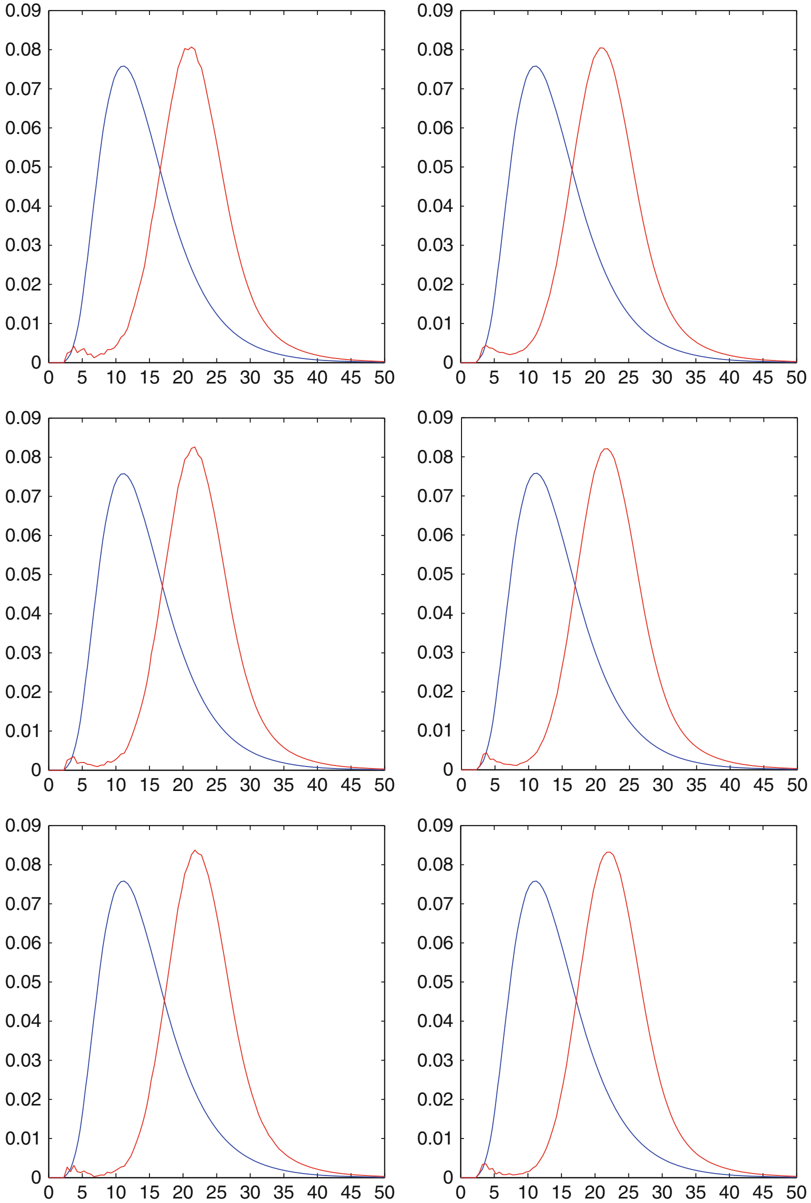


Fig. 5 Simulation of the number-based diameter density: comparison of initial density and density at $t = 2.0$ (*top*), $t = 2.5$ (*middle*), $t = 3.0$ (*bottom*). MC (*left*) versus QMC (*right*) results.

4 Conclusion

In this paper, we have proposed a method for the calculation of drop size distribution in a spray. The method uses a QMC simulation of the coagulation equation.

It starts with a formulation of the MEF including a limitation of small drops: the solution is a generalized gamma distribution, which is used as initial data for the simulation. Time is discretized and the spray is simulated using a constant number of particles. They are sampled from the initial distribution; then they evolve according to the dynamics described in the conservation form of the coagulation equation. A low discrepancy sequence is used for coalescence. In order to make a proper use of the great uniformity of the quasi-random points, the particles are reordered according to their size at every time step. The results of computations show that this algorithm converges faster than its MC counterpart. Finally we apply our scheme to the simulation of the spray generated by a new SOD device. The method is able to produce bimodal distributions which are observed in experiments and which may be due to drop merging.

The QMC method is shown to converge as the number of simulation particles tends to infinity; but there is a gap between the theoretical order of convergence and the order observed in computations: the analysis must be pursued. The present work shows qualitative agreement between computations and experiments; further work is needed to obtain quantitative agreement under various operating conditions.

References

1. Babinsky, E., Sojka, P.E.: Modeling drop size distributions. Progress in Energy and Combustion Science **28**, 303–329 (2002)
2. Dobre, M., Bolle, L.: Practical design of ultrasonic spray devices: experimental testing of several atomizer geometries. Experimental Thermal and Fluid Science **26**, 205–211 (2002)
3. Drake, R.L.: A general mathematical survey of the coagulation equation. In: Hidy, G.M., Brock, J.R. (eds.) Topics in Current Aerosol Research, Part 2, pp. 201–376. Pergamon Press, Oxford (1972)
4. Dumouchel, C.: A new formulation of the maximum entropy formalism to model liquid spray drop-size distribution. Particle and Particle Systems Characterization **23**, 468–479 (2006)
5. Dumouchel, C.: The maximum entropy formalism and the prediction of liquid spray drop-size distribution. Entropy **11**, 713–747 (2009)
6. Dumouchel, C., Sindayihebura, D., Bolle, L.: Application of the maximum entropy formalism on sprays produced by ultrasonic atomizers. Particle and Particle Systems Characterization **20**, 150–161 (2003)
7. Kocamustafaogullari, G., Ishii, M.: Foundation of the interfacial area transport equation and its closure relations. International Journal of Heat and Mass Transfer **38**, 481–493 (1995)
8. Lécot, C., Tarhini, A.: A quasi-stochastic simulation of the general dynamics equation for aerosols. Monte Carlo Methods and Applications **13**, 369–388 (2007)
9. Lécot, C., Wagner, W.: A quasi-Monte Carlo scheme for Smoluchowski's coagulation equation. Mathematics of Computation **73**, 1953–1966 (2004)
10. Lienhard, J.H., Meyer, P.L.: A physical basis for the generalized gamma distribution. Quarterly of Applied Mathematics **25**, 330–334 (1967)

11. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia (1992)
12. Ramabhadran, T.E., Peterson, T.W., Seinfeld, J.H.: Dynamics of aerosol coagulation and condensation. American Institute of Chemical Engineers Journal **22**, 840–851 (1976)
13. Tarhini, A.: Analyse numérique des méthodes quasi-Monte Carlo appliquées aux modèles d'agglomération. PhD thesis, Université de Savoie (2008)
14. Tembely, M.: Étude de l'atomisation induite par interactions fluide-structure. PhD thesis, Université de Grenoble (2010)
15. Tembely, M., Lécot, C., Soucemarianadin, A.: Prediction and evolution of drop-size distribution for a new ultrasonic atomizer. Applied Thermal Engineering **31**, 656–667 (2011)

Nonasymptotic Bounds on the Mean Square Error for MCMC Estimates via Renewal Techniques

Krzysztof Łatuszyński, Błażej Miasojedow, and Wojciech Niemiro

Abstract The Nummellin's split chain construction allows to decompose a Markov chain Monte Carlo (MCMC) trajectory into i.i.d. "excursions". Regenerative MCMC algorithms based on this technique use a random number of samples. They have been proposed as a promising alternative to usual fixed length simulation (Hobert et al., Biometrika 89:731–743, 2002; Mykland et al., J. Am. Statist. Assoc. 90:233–241, 1995; Rosenthal, J. Amer. Statist. Association 90:558–566, 1995). In this note we derive nonasymptotic bounds on the mean square error (MSE) of regenerative MCMC estimates via techniques of renewal theory and sequential statistics. These results are applied to construct confidence intervals. We then focus on two cases of particular interest: chains satisfying the Doeblin condition and a geometric drift condition. Available explicit nonasymptotic results are compared for different schemes of MCMC simulation.

1 Introduction

Consider a typical MCMC setting, where π is a probability distribution on $\mathscr{X}$ and $f : \mathscr{X} \to \mathbb{R}$ a Borel measurable function. The objective is to compute (estimate) the integral

K. Łatuszyński
Department of Statistics, University of Warwick, CV4 7AL, Coventry, UK
e-mail: latuch@gmail.com

B. Miasojedow
Institute of Applied Mathematics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland
e-mail: bmia@mimuw.edu.pl

W. Niemiro
Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Chopina 12/18, 87-100 Toruń, Poland
e-mail: wniemiro@gmail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_31, © Springer-Verlag Berlin Heidelberg 2012

$$\theta := \pi f = \int_{\mathscr{X}} \pi(\mathrm{d}x) f(x). \tag{1}$$

Assume that direct simulation from π is intractable. Therefore one uses an ergodic Markov chain with transition kernel P and stationary distribution π to sample approximately from π. Numerous computational problems from Bayesian inference, statistical physics or combinatorial enumeration fit into this setting. We refer to [9, 29, 31] for theory and applications of MCMC.

Let $(X_n)_{n\geq 0}$ be the Markov chain in question. Typically one discards an initial part of the trajectory (called burn-in, say of length t) to reduce bias, one simulates the chain for n further steps and one approximates θ with an ergodic average:

$$\hat{\theta}^{\mathrm{fix}}_{t,n} = \frac{1}{n} \sum_{i=t}^{t+n-1} f(X_i). \tag{2}$$

The fixed numbers t and n are the parameters of the algorithm. Asymptotic validity of (2) is ensured by a Strong Law of Large Numbers and a Central Limit Theorem (CLT). Under appropriate regularity conditions [4, 31], it holds that

$$\sqrt{n}(\hat{\theta}^{\mathrm{fix}}_{t,n} - \theta) \to \mathscr{N}(0, \sigma^2_{\mathrm{as}}(f)), \qquad (n \to \infty), \tag{3}$$

where $\sigma^2_{\mathrm{as}}(f)$ is called the asymptotic variance. In contrast with the asymptotic theory, explicit *nonasymptotic* error bounds for $\hat{\theta}^{\mathrm{fix}}_{t,n}$ appear to be very difficult to derive in practically meaningful problems.

Regenerative simulation offers a way to get around some of the difficulties. The split chain construction introduced in [2, 27] (to be described in Sect. 2) allows for partitioning the trajectory $(X_n)_{n\geq 0}$ into i.i.d. random tours (excursions) between consecutive regeneration times $T_0, T_1, T_2, \ldots$. Random variables

$$\Xi_k(f) := \sum_{i=T_{k-1}}^{T_k - 1} f(X_i) \tag{4}$$

are i.i.d. for $k = 1, 2, \ldots$ ($\Xi_0(f)$ can have a different distribution). Mykland et al. in [24] suggested a practically relevant recipe for identifying $T_0, T_1, T_2, \ldots$ in simulations (formula (2) in Sect. 2). This resolves the burn-in problem since one can just ignore the part until the first regeneration T_0. One can also stop the simulation at a regeneration time, say T_r, and simulate r full i.i.d. tours, cf. Sect. 4 of [32]. Thus one estimates θ by

$$\hat{\theta}^{\mathrm{reg}}_r := \frac{1}{T_r - T_0} \sum_{i=T_0}^{T_r - 1} f(X_i) = \frac{\sum_{k=1}^{r} \Xi_k(f)}{\sum_{k=1}^{r} \tau_k}, \tag{5}$$

where $\tau_k = T_k - T_{k-1} = \Xi_k(1)$ are the lengths of excursions. The number of tours r is fixed and the total simulation effort T_r is random. Since $\hat{\theta}_r^{\text{reg}}$ involves i.i.d. random variables, classical tools seem to be sufficient to analyse its behaviour. Asymptotically, (5) is equivalent to (2) because

$$\sqrt{rm}(\hat{\theta}_r^{\text{reg}} - \theta) \to \mathcal{N}(0, \sigma^2_{\text{as}}(f)), \qquad (r \to \infty),$$

where $m := \mathbb{E}\,\tau_1$. Now $rm = \mathbb{E}\,(T_r - T_0)$, the expected length of the trajectory, plays the role of n. However, our attempt at nonasymptotic analysis in Sect. 3.1 reveals unexpected difficulties: our bounds involve m in the denominator and in most practically relevant situations m is unknown.

If m is known then instead of (5) one can use an unbiased estimator

$$\tilde{\theta}_r^{\text{unb}} := \frac{1}{rm} \sum_{k=1}^{r} \Xi_k(f), \tag{6}$$

Quite unexpectedly, (6) is *not equivalent* to (5), even in a weak asymptotic sense. The standard CLT for i.i.d. summands yields

$$\sqrt{rm}(\tilde{\theta}_r^{\text{unb}} - \theta) \to \mathcal{N}(0, \sigma^2_{\text{unb}}(f)), \qquad (r \to \infty),$$

where $\sigma^2_{\text{unb}}(f) := \operatorname{Var} \Xi_1(f)/m$ is in general different from $\sigma^2_{\text{as}}(f)$.

We introduce a new regenerative-sequential simulation scheme, for which better nonasymptotic results can be derived. Namely, we fix n and define

$$R(n) := \min\{r : T_r > T_0 + n\}.$$

The estimator is defined as

$$\hat{\theta}_n^{\text{reg-seq}} := \frac{1}{T_{R(n)} - T_0} \sum_{i=T_0}^{T_{R(n)}-1} f(X_i) = \frac{\sum_{k=1}^{R(n)} \Xi_k(f)}{\sum_{k=1}^{R(n)} \tau_k}. \tag{7}$$

We thus generate a random number of tours as well as a random number of samples.

Our approach is based on inequalities for the mean square error,

$$\text{MSE} := \mathbb{E}\,(\hat{\theta} - \theta)^2.$$

Bounds on the MSE can be used to construct fixed precision confidence intervals. The goal is to obtain an estimator $\hat{\theta}$ which satisfies

$$\mathbb{P}(|\hat{\theta} - \theta| \leq \varepsilon) \geq 1 - \alpha, \tag{8}$$

for given ε and α. We combine the MSE bounds with the so called "median trick" [15, 26]. One runs MCMC repeatedly and computes the median of independent estimates to boost the level of confidence. In our paper, the median trick is used in conjunction with regenerative simulation.

The organization of the paper is the following. In Sect. 2 we recall the split chain construction. Nonasymptotic bounds for regenerative estimators defined by (5), (6) and (7) are derived in Sect. 3. Derivation of more explicit bounds which involve only computable quantities is deferred to Sects. 5 and 6, where we consider classes of chains particularly important in the MCMC context. An analogous analysis of the non-regenerative scheme (2) was considered in [20] and (in a different setting and using different methods) in [33].

In Sect. 4 we discuss the median trick. The resulting confidence intervals are compared with *asymptotic* results based on the CLT.

In Sect. 5 we consider Doeblin chains, i.e., uniformly ergodic chains that satisfy a one step minorization condition. We compare regenerative estimators (5), (6) and (7). Moreover, we also consider a perfect sampler available for Doeblin chains, cf. [14,35]. We show that confidence intervals based on the median trick can outperform those obtained via exponential inequalities for a single run simulation.

In Sect. 6 we proceed to analyze geometrically ergodic Markov chains, assuming a drift condition towards a small set. We briefly compare regenerative schemes (5) and (7) in this setting (the unbiased estimator (6) cannot be used, because m is unknown).

2 Regenerative Simulation

We describe the setting more precisely. Let $(X_n)_{n\geq 0}$ be a Markov chain with transition kernel P on a Polish space $\mathcal{X}$ with stationary distribution π, i.e., $\pi P = \pi$. Assume P is π-irreducible. The regeneration/split construction of Nummelin [27] and Athreya and Ney [2] rests on the following assumption.

Assumption 1 (Small Set) *There exist a Borel set $J \subseteq \mathcal{X}$ of positive π measure, a number $\beta > 0$ and a probability measure ν such that*

$$P(x, \cdot) \geq \beta \mathbb{I}(x \in J)\nu(\cdot).$$

Under Assumption 1 we can define a bivariate Markov chain $(X_n, \varGamma_n)$ on the space $\mathcal{X} \times \{0, 1\}$ in the following way. Variable $\varGamma_{n-1}$ depends only on X_{n-1} via $\mathbb{P}(\varGamma_{n-1} = 1 | X_{n-1} = x) = \beta \mathbb{I}(x \in J)$. The rule of transition from $(X_{n-1}, \varGamma_{n-1})$ to X_n is given by

$$\mathbb{P}(X_n \in A | \varGamma_{n-1} = 1, X_{n-1} = x) = \nu(A),$$

$$\mathbb{P}(X_n \in A | \varGamma_{n-1} = 0, X_{n-1} = x) = Q(x, A),$$

where Q is the normalized "residual" kernel given by

$$Q(x,\cdot) := \frac{P(x,\cdot) - \beta\mathbb{I}(x \in J)\nu(\cdot)}{1 - \beta\mathbb{I}(x \in J)}.$$

Whenever $\varGamma_{n-1} = 1$, the chain regenerates at moment n. The regeneration epochs are

$$T_0 := \min\{n : \varGamma_{n-1} = 1\},$$
$$T_k := \min\{n > T_{k-1} : \varGamma_{n-1} = 1\}.$$

The random tours defined by

$$\varXi_k := (X_{T_{k-1}}, \ldots, X_{T_k - 1}, \tau_k), \qquad \text{where} \quad \tau_k = T_k - T_{k-1}, \tag{9}$$

are independent. Without loss of generality, we assume that $X_0 \sim \nu(\cdot)$, unless stated otherwise. Under this assumption, all the tours $\varXi_k$ are i.i.d. for $k > 0$. We therefore put $T_0 := 0$ and simplify notation. In the sequel symbols $\mathbb{P}$ and $\mathbb{E}$ without subscripts refer to the chain started at ν. If the initial distribution ξ is other than ν, it will be explicitly indicated by writing $\mathbb{P}_\xi$ and $\mathbb{E}_\xi$. Notation $m = \mathbb{E}\,\tau_1$ stands throughout the paper.

We assume that we are able to *identify* regeneration times T_k. Mykland et al. pointed out in [24] that actual sampling from Q can be avoided. We can generate the chain using transition probability P and then recover the regeneration indicators via

$$\mathbb{P}(\varGamma_{n-1} = 1 | X_n, X_{n-1}) = \mathbb{I}(X_{n-1} \in J)\frac{\beta\nu(\mathrm{d}X_n)}{P(X_{n-1}, \mathrm{d}X_n)},$$

where $\nu(\mathrm{d}y)/P(x, \mathrm{d}y)$ denotes the Radon-Nikodym derivative (in practice, the ratio of densities). Mykland's trick has been established in a number of practically relevant families (e.g., hierarchical linear models) and specific Markov chains implementations, such as block Gibbs samplers or variable-at-a-time chains, see [17, 25].

3 General Results for Regenerative Estimators

Recall that $f : \mathscr{X} \to \mathbb{R}$ is a measurable function and $\theta = \pi f$. We consider block sums $\varXi_k(f)$ defined by (4). The general Kac theorem states that the mean occupation time during one tour is proportional to the stationary measure (Theorem 10.0.1 in [23] or Eqs. 3.3.4, 3.3.6, 3.4.7, and 3.5.1 in [28]). This yields

$$m = \frac{1}{\beta\pi(J)}, \qquad \mathbb{E}\,\varXi_1(f) = m\pi f = m\theta.$$

From now on we assume that $\mathbb{E}\,\Xi_1(f)^2 < \infty$ and $\mathbb{E}\,\tau_1^2 < \infty$. For a discussion of these assumptions in the MCMC context, see [13]. Let $\bar{f} := f - \pi f$ and define

$$\sigma^2_{\text{as}}(f) := \frac{\mathbb{E}\,\Xi_1(\bar{f})^2}{m}, \tag{10}$$

$$\sigma^2_{\tau} := \frac{\operatorname{Var}\tau_1}{m}. \tag{11}$$

Remark 1. Under Assumption 1, finiteness of $\mathbb{E}\,\Xi_1(\bar{f})^2$ is a sufficient and necessary condition for the CLT to hold for Markov chain $(X_n)_{n\geq 0}$ and function f. This fact is proved in [4] in a more general setting. For our purposes it is important to note that $\sigma^2_{\text{as}}(f)$ in (10) is indeed the *asymptotic variance* which appears in the CLT.

3.1 Results for $\hat{\theta}^{\text{reg}}_r$

We are to bound the estimation error which can be expressed as follows:

$$\hat{\theta}^{\text{reg}}_r - \theta = \frac{\sum_{k=1}^r \big(\Xi_k(f) - \theta\tau_k\big)}{\sum_{k=1}^r \tau_k} = \frac{\sum_{k=1}^r d_k}{T_r}. \tag{12}$$

where $d_k := \Xi_k(f) - \theta\tau_k = \Xi_k(\bar{f})$. Therefore, for any $0 < \delta < 1$,

$$\mathbb{P}(|\hat{\theta}^{\text{reg}}_r - \theta| > \varepsilon) \;\leq\; \mathbb{P}\left(\left|\sum_{k=1}^r d_k\right| > rm\varepsilon(1-\delta)\right) + \mathbb{P}\Big(T_r < rm(1-\delta)\Big).$$

Since d_k are i.i.d. with $\mathbb{E}\,d_1 = 0$ and $\operatorname{Var} d_1 = m\sigma^2_{\text{as}}(f)$, we can use Chebyshev inequality to bound the first term above:

$$\mathbb{P}\left(\left|\sum_{k=1}^r d_k\right| > rm\varepsilon(1-\delta)\right) \;\leq\; \frac{\sigma^2_{\text{as}}(f)}{rm\varepsilon^2(1-\delta)^2}.$$

The second term can be bounded similarly. We use the fact that τ_k are i.i.d. with $\mathbb{E}\,\tau_1 = m$ to write

$$\mathbb{P}\Big(T_r < rm(1-\delta)\Big) \;\leq\; \frac{\sigma^2_{\tau}}{rm^2\delta^2}.$$

We conclude the above calculation with in following Theorem.

Theorem 1 *Under Assumption 1 the following holds for every* $0 < \delta < 1$

$$\mathbb{P}(|\hat{\theta}^{\text{reg}}_r - \theta| > \varepsilon) \;\leq\; \frac{1}{rm}\left[\frac{\sigma^2_{\text{as}}(f)}{\varepsilon^2(1-\delta)^2} + \frac{\sigma^2_{\tau}}{m\delta^2}\right] \tag{13}$$

and is minimized by

$$\delta = \delta^* := \frac{\sigma_\tau^{2/3}}{\sigma_{\rm as}^{2/3}(f)\varepsilon^{-2/3} + \sigma_\tau^{2/3}}.$$

Obviously, $\mathbb{E}\, T_r = rm$ is the expected length of trajectory. The main drawback of Theorem 1 is that the bound on the estimation error depends on m, which is typically unknown. Replacing m by 1 in (13) would be highly inefficient. This fact motivates our study of another estimator, $\hat{\theta}_n^{\text{reg-seq}}$, for which we can obtain much more satisfactory results. We think that the derivation of better nonasymptotic bounds for $\hat{\theta}_r^{\text{reg}}$ (not involving m) is an open problem.

3.2 Results for $\tilde{\theta}_r^{\text{unb}}$

Recall that $\tilde{\theta}_r^{\text{unb}}$ can be used only when m is known and this situation is rather rare in MCMC applications. The analysis of $\tilde{\theta}_r^{\text{unb}}$ is straightforward, because it is simply a sum of i.i.d. random variables. In particular, we obtain the following.

Corollary 1 *Under Assumption 1,*

$$\mathbb{E}\,(\tilde{\theta}_r^{\text{unb}} - \theta)^2 = \frac{\sigma_{\rm unb}^2(f)}{rm}, \qquad \mathbb{P}(|\tilde{\theta}_r^{\text{unb}} - \theta| > \varepsilon) \le \frac{\sigma_{\rm unb}^2(f)}{rm\,\varepsilon^2}.$$

Note that $\sigma_{\rm unb}^2(f) = \operatorname{Var}\Xi_1(f)/m$ can be expressed as

$$\sigma_{\rm unb}^2(f) = \sigma_{\rm as}^2(f) + \theta^2\sigma_\tau^2 + 2\theta\rho(\bar{f}, 1), \tag{14}$$

where $\rho(\bar{f}, 1) := \operatorname{Cov}(\Xi_1(\bar{f}), \Xi_1(1))/m$. This follows from the simple observation that $\operatorname{Var}\Xi_1(f) = \mathbb{E}\,(\Xi_1(\bar{f}) + \theta(\tau_1 - m))^2$.

3.3 Results for $\hat{\theta}_n^{\text{reg-seq}}$

The result below bounds the MSE and the expected number of samples used to compute the estimator.

Theorem 2 *If Assumption 1 holds then*

$$(i) \quad \mathbb{E}\,(\hat{\theta}_n^{\text{reg-seq}} - \theta)^2 \le \frac{\sigma_{\rm as}^2(f)}{n^2}\,\mathbb{E}\,T_{R(n)}$$

and

$$(ii) \quad \mathbb{E}\, T_{R(n)} \le n + C_0,$$

where

$$C_0 := \sigma_\tau^2 + m.$$

Corollary 2 *Under Assumption 1,*

$$\mathbb{E}\, (\hat{\theta}_n^{\text{reg-seq}} - \theta)^2 \le \frac{\sigma_{\text{as}}^2(f)}{n} \left(1 + \frac{C_0}{n}\right), \tag{15}$$

$$\mathbb{P}\, (|\hat{\theta}_n^{\text{reg-seq}} - \theta| > \varepsilon) \le \frac{\sigma_{\text{as}}^2(f)}{n\varepsilon^2} \left(1 + \frac{C_0}{n}\right). \tag{16}$$

Remark 2. Note that the leading term $\sigma_{\text{as}}^2(f)/n$ in (15) is "asymptotically correct" in the sense that the standard fixed length estimator has MSE $\sim \sigma_{\text{as}}^2(f)/n$. The regenerative-sequential scheme is "close to the fixed length simulation", because $\lim_{n\to\infty}\mathbb{E}\, T_{R(n)}/n = 1$.

Proof (of Theorem 2). Just as in (12) we have

$$\hat{\theta}_n^{\text{reg-seq}} - \theta = \frac{\sum_{k=1}^{R(n)} (\varXi_k(f) - \theta\tau_k)}{\sum_{k=1}^{R(n)} \tau_k} = \frac{1}{T_{R(n)}} \sum_{k=1}^{R(n)} d_k,$$

where pairs (d_k, τ_k) are i.i.d. with $\mathbb{E}\, d_1 = 0$ and $\text{Var} d_1 = m\sigma_{\text{as}}^2(f)$. Since $T_{R(n)} > n$, it follows that

$$\mathbb{E}\, (\hat{\theta}_n^{\text{reg-seq}} - \theta)^2 \le \frac{1}{n^2}\mathbb{E} \left(\sum_{k=1}^{R(n)} d_k \right)^2.$$

Since $R(n)$ is a stopping time with respect to $\mathscr{G}_k = \sigma((d_1, \tau_1), \ldots, (d_k, \tau_k))$, we are in a position to apply the two Wald's identities (see Appendix). The second identity yields

$$\mathbb{E} \left(\sum_{k=1}^{R(n)} d_k \right)^2 = \text{Var}\, d_1\, \mathbb{E}\, R(n) = m\sigma_{\text{as}}^2(f)\, \mathbb{E}\, R(n).$$

In this expression we can replace $m\mathbb{E}\, R(n)$ by $\mathbb{E}\, T_{R(n)}$ because of the first Wald's identity:

$$\mathbb{E}\, T_{R(n)} = \mathbb{E} \sum_{k=1}^{R(n)} \tau_k = \mathbb{E}\, \tau_1\, \mathbb{E}\, R(n) = m\mathbb{E}\, R(n)$$

and (i) follows.

We now focus attention on bounding the expectation of the "overshoot" $\varDelta(n) := T_{R(n)} - n$. Since we assume that $X_0 \sim \nu$, the cumulative sums $\tau_1 = T_1 < T_2 < \ldots < T_k < \ldots$ form a (nondelayed) renewal process in discrete time. Let us invoke the following elegant theorem of Lorden [21, Theorem 1]:

$$\mathbb{E}\,\Delta(n) \le \mathbb{E}\,\tau_1^2/m.$$

This inequality combined with (11) yields immediately $\mathbb{E}\,T_{R(n)} = \mathbb{E}\,(n + \Delta(n)) \le n + \sigma_\tau^2 + m$, i.e., (ii).

4 The Median Trick

This ingeneous method of constructing fixed precision MCMC algorithms was introduced in 1986 in [15], later used in many papers concerned with computational complexity and further developed in [26]. We run l independent copies of the Markov chain. Let $\hat{\theta}^{(j)}$ be an estimator computed in jth run. The final estimate is $\hat{\theta} := \mathrm{med}(\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(l)})$. To ensure that $\hat{\theta}$ satisfies (8), we require that $\mathbb{P}(|\hat{\theta}^{(j)} - \theta| > \varepsilon) \le a$ $(j = 1, \dots, l)$ for some modest level of confidence $1 - a < 1 - \alpha$. This is obtained via Chebyshev's inequality, if a bound on MSE is available. The well-known Chernoff's inequality gives for odd l,

$$\mathbb{P}\,(|\hat{\theta} - \theta| \ge \varepsilon) \le \frac{1}{2}\,[4a(1-a)]^{l/2} = \frac{1}{2}\exp\left\{\frac{l}{2}\ln\,[4a(1-a)]\right\}. \tag{17}$$

It is pointed out in [26] that under some assumptions there is a universal choice of a, which nearly minimizes the overall number of samples, $a^* \approx 0.11969$.

Let us now examine how the median trick works in conjunction with regenerative MCMC. We focus on $\hat{\theta}_n^{\text{reg-seq}}$, because Corollary 2 gives the best available bound on MSE. We first choose n such that the right hand side of (16) is less than or equal to a^*. Then choose l big enough to make the right hand side of (17) (with $a = a^*$) less than or equal to α. Compute estimator $\hat{\theta}_n^{\text{reg-seq}}$ repeatedly, using l independent runs of the chain. We can see that (8) holds if

$$n \ge \frac{C_1\sigma_{\text{as}}^2(f)}{\varepsilon^2} + C_0, \tag{18}$$

$$l \ge C_2 \ln(2\alpha)^{-1} \quad \text{and } l \text{ is odd}, \tag{19}$$

where $C_1 := 1/a^* \approx 8.3549$ and $C_2 := 2/\ln\,[4a^*(1-a^*)]^{-1} \approx 2.3147$ are absolute constants. Indeed, (18) entails $C_1\sigma_{\text{as}}^2(f)/(\varepsilon^2 n) \le 1 - C_0/n$, so $C_1\sigma_{\text{as}}^2(f)/(\varepsilon^2 n)(1 + C_0/n) \le 1 - C_0^2/n^2 < 1$. Consequently $\sigma_{\text{as}}^2(f)/(\varepsilon^2 n)(1 + C_0/n) < a^*$ and we are in a position to apply (16).

The overall (expected) number of generated samples is $l\,\mathbb{E}\,T_{R(n)} \sim nl$ as $\varepsilon \to 0$ and $n \to \infty$, by Theorem 2 (ii). Consequently for $\varepsilon \to 0$ the cost of the algorithm is approximately

$$nl \sim C\,\frac{\sigma_{\text{as}}^2(f)}{\varepsilon^2}\log(2\alpha)^{-1}, \tag{20}$$

where $C = C_1 C_2 \approx 19.34$. To see how tight is the obtained lower bound, let us compare (20) with the familiar asymptotic approximation, based on the CLT. Consider an estimator based on one MCMC run of length n, say $\hat{\theta}_n = \hat{\theta}^{\text{fix}}_{t,n}$ with $t = 0$. From (3) we infer that

$$\lim_{\varepsilon \to 0} \mathbb{P}(|\hat{\theta}_n - \theta| > \varepsilon) = \alpha,$$

holds for

$$n \sim \frac{\sigma^2_{\text{as}}(f)}{\varepsilon^2} \left[\Phi^{-1}(1 - \alpha/2)\right]^2, \tag{21}$$

where Φ^{-1} is the quantile function of the standard normal distribution. Taking into account the fact that $[\Phi^{-1}(1 - \alpha/2)]^2 \sim 2\log(2\alpha)^{-1}$ for $\alpha \to 0$ we arrive at the following conclusion. The right hand side of (20) is bigger than (21) roughly by a constant factor of about 10 (for small ε and α). The important difference is that (20) is sufficient for an *exact* confidence interval while (21) only for an *asymptotic* one.

5 Doeblin Chains

Assume that the transition kernel P satisfies the following Doeblin condition: there exist $\beta > 0$ and a probability measure ν such that

$$P(x, \cdot) \geq \beta \nu(\cdot) \quad \text{for every} \quad x \in \mathcal{X}. \tag{22}$$

This amounts to taking $J := \mathcal{X}$ in Assumption 1. Condition (22) implies that the chain is uniformly ergodic. We refer to [31] and [23] for definition of uniform ergodicity and related concepts. As a consequence of the regeneration construction, in our present setting τ_1 is distributed as a geometric random variable with parameter β and therefore

$$m = \mathbb{E}\,\tau_1 = \frac{1}{\beta} \qquad \text{and} \qquad \sigma^2_\tau = \frac{\operatorname{Var}\tau_1}{m} = \frac{1-\beta}{\beta}.$$

Bounds on the asymptotic variance $\sigma^2_{\text{as}}(f)$ under (22) are well known. Let $\sigma^2 = \pi \bar{f}^2$ be the stationary variance. Results in Sect. 5 of [4] imply that

$$\sigma^2_{\text{as}}(f) \leq \sigma^2 \left(1 + \frac{2\sqrt{1-\beta}}{1 - \sqrt{1-\beta}}\right) \leq \frac{4\sigma^2}{\beta}. \tag{23}$$

Since in [4] a more general situation is considered, which complicates the formulas, let us give a simple derivation of (23) under (22). By (10) and the formula (29) given in the Appendix,

$$\sigma^2_{\mathrm{as}}(f) \le \frac{\mathbb{E}\,\Xi_1(|\bar{f}|)^2}{m} = \mathbb{E}_\pi \bar{f}(X_0)^2 + 2\sum_{i=1}^{\infty} \mathbb{E}_\pi |\bar{f}(X_0)\bar{f}(X_i)| \mathbb{I}(\tau_1 > i).$$

The first term above is equal to σ^2. To bound the terms of the series, use Cauchy-Schwarz and the fact that, under (22), random variables X_0 and τ_1 are independent. Therefore $\mathbb{E}_\pi |\bar{f}(X_0)\bar{f}(X_i)|\mathbb{I}(\tau_1 > i) \le \big(\mathbb{E}_\pi \bar{f}(X_i)^2 \mathbb{E}_\pi \bar{f}(X_0)^2 \mathbb{P}_\pi(\tau_1 > i)\big)^{1/2} = \sigma^2(1-\beta)^{i/2}$. Computing the sum of the geometric series yields (23).

If the chain is reversible, there is a better bound than (23). We can use the well-known formula for $\sigma^2_{\mathrm{as}}(f)$ in terms of the spectral decomposition of P (e.g., expression "C" in [11]). Results of [30] show that the spectrum of P is a subset of $[-1+\beta, 1-\beta]$. We conclude that for reversible Doeblin chains,

$$\sigma^2_{\mathrm{as}}(f) \le \frac{2-\beta}{\beta}\sigma^2 \le \frac{2\sigma^2}{\beta}. \tag{24}$$

An important class of reversible chains are Independence Metropolis-Hastings chains (see e.g., [31]) that are known to be uniformly ergodic if and only if the rejection probability $r(x)$ is uniformly bounded from 1 by say $1-\beta$. This is equivalent to the candidate distribution being bounded below by $\beta\pi$ (cf. [1,22]) and translates into (22) with $\nu = \pi$. The formula for $\sigma^2_{\mathrm{as}}(f)$ in (23) and (24) depends on β in an optimal way. Moreover (24) is sharp. To see this consider the following example.

Example 1. Let $\beta \le 1/2$ and define a Markov chain $(X_n)_{n\ge 0}$ on $\mathscr{X} = \{0,1\}$ with stationary distribution $\pi = [1/2, 1/2]$ and transition matrix

$$P = \begin{bmatrix} 1-\beta/2 & \beta/2 \\ \beta/2 & 1-\beta/2 \end{bmatrix}.$$

Hence $P = \beta\pi + (1-\beta)I_2$ and $P(x,\cdot) \ge \beta\pi$. Note that the residual kernel Q is in our example the identity matrix I_2. Thus, before the first regeneration τ_1 the chain does not move. Let $f(x) = x$. Thus $\sigma^2 = 1/4$. To compute $\sigma^2_{\mathrm{as}}(f)$ we use another well-known formula (expression "B" in [11]):

$$\begin{aligned}\sigma^2_{\mathrm{as}}(f) &= \sigma^2 + 2\sum_{i=1}^{\infty} \mathrm{Cov}_\pi\{f(X_0), f(X_i)\} \\ &= \sigma^2 + 2\sigma^2 \sum_{i=1}^{\infty}(1-\beta)^i = \frac{2-\beta}{\beta}\sigma^2.\end{aligned}$$

To compute $\sigma^2_{\mathrm{unb}}(f)$, note that $\Xi_1(f) = \mathbb{I}(X_0 = 1)\tau_1$. Since τ_1 is independent of X_0 and $X_0 \sim \nu = \pi$ we obtain

$$\sigma^2_{\mathrm{unb}}(f) = \beta \mathrm{Var} \Xi_1(f) = \beta \big[\mathbb{E}\,\mathrm{Var}(\Xi_1(f)|X_0) + \mathrm{Var}\mathbb{E}\,(\Xi_1(f)|X_0) \big] \\ = \frac{1-\beta}{2\beta} + \frac{1}{4\beta} = \frac{3-2\beta}{\beta}\sigma^2.$$

Interestingly, in this example $\sigma^2_{\mathrm{unb}}(f) > \sigma^2_{\mathrm{as}}(f)$.

In the setting of this section, we will now compare upper bounds on the total simulation effort needed for different MCMC schemes to get $\mathbb{P}(|\hat{\theta} - \theta| > \varepsilon) \leq \alpha$.

5.1 Regenerative-Sequential Estimator and the Median Trick

Recall that this simulation scheme consists of l MCMC runs, each of approximate length n. Substituting either (23) or (24) in (20) we obtain that the expected number of samples is

$$nl \sim 19.34 \frac{4\sigma^2}{\beta\varepsilon^2} \log(2\alpha)^{-1} \quad \text{and} \quad nl \sim 19.34 \frac{(2-\beta)\sigma^2}{\beta\varepsilon^2} \log(2\alpha)^{-1} \tag{25}$$

(respectively in the general case and for reversible chains). Note also that in the setting of this Section we have an exact expression for the constant C_0 in Theorem 2. Indeed, $C_0 = 2/\beta - 1$.

5.2 Standard One-Run Average and Exponential Inequality

For uniformly ergodic chains a direct comparison of our approach to exponential inequalities [10, 18] is possible. We focus on the result proved in [18] for chains on a countable state space. This inequality is tight in the sense that it reduces to the Hoeffding bound when specialised to the i.i.d. case. For f bounded let $\|f\|_\infty := \sup_{x \in \mathscr{X}} |f(x)|$. Consider the simple average over n Markov chain samples, say $\hat{\theta}_n = \hat{\theta}^{\mathrm{fix}}_{t,n}$ with $t = 0$. For an arbitrary initial distribution ξ we have

$$\mathbb{P}_\xi(|\hat{\theta}_n - \theta| > \varepsilon) \leq 2 \exp\left\{ -\frac{n-1}{2} \left(\frac{2\beta}{\|f\|_\infty}\varepsilon - \frac{3}{n-1} \right)^2 \right\}.$$

After identifying the leading terms we can see that to make the right hand side less than α we need

$$n \sim \frac{\|f\|^2_\infty}{2\beta^2\varepsilon^2} \log(\alpha/2)^{-1} \geq \frac{2\sigma^2}{\beta^2\varepsilon^2} \log(\alpha/2)^{-1}. \tag{26}$$

Comparing (25) with (26) yields a ratio of roughly 40β or 20β respectively. This in particular indicates that the dependence on β in [10, 18] probably can be improved. We note that in examples of practical interest β usually decays exponentially with the dimension of $\mathscr{X}$ and using the regenerative-sequential-median scheme will often result in a lower total simulation cost. Moreover, this approach is valid for an unbounded target function f, in contrast with classical exponential inequalities.

5.3 Perfect Sampler and the Median Trick

For Doeblin chains, if regeneration times can be identified, perfect sampling can be performed easily as a version of read-once algorithm [35]. This is due to the following observation. If condition (22) holds and $X_0 \sim \nu$ then

$$X_{T_k-1}, \quad k = 1, 2, \ldots$$

are i.i.d. random variables from π (see [4, 5, 14, 28] for versions of this result). Therefore from each random tour between regeneration times one can obtain a single perfect sample (by taking the state of the chain prior to regeneration) and use it for i.i.d. estimation. We define

$$\hat{\theta}_r^{\mathrm{perf}} := \frac{1}{r} \sum_{k=1}^{r} f(X_{T_k-1}).$$

Clearly

$$\mathbb{E}\,(\hat{\theta}_r^{\mathrm{perf}} - \theta)^2 = \frac{\sigma^2}{r} \qquad \text{and} \qquad \mathbb{P}(|\hat{\theta}_r^{\mathrm{perf}} - \theta| > \varepsilon) \le \frac{\sigma^2}{r\varepsilon^2}.$$

Note that to compute $\hat{\theta}_r^{\mathrm{perf}}$ we need to simulate $n \sim r/\beta$ steps of the Markov chain. If we combine the perfect sampler with the median trick we obtain an algorithm with the expected number of samples

$$nl \sim 19.34 \frac{\sigma^2}{\beta\varepsilon^2} \log(2\alpha)^{-1}. \tag{27}$$

Comparing (25) with (26) and (27) leads to the conclusion that if one targets rigorous nonasymptotic results in the Doeblin chain setting, the approach described here outperforms other methods.

5.4 Remarks on Other Schemes

The bound for $\hat{\theta}_r^{\mathrm{reg}}$ in Theorem 1 is clearly inferior to that for $\hat{\theta}_n^{\mathrm{reg\text{-}seq}}$ in Corollary 2. Therefore we excluded the scheme based on $\hat{\theta}_r^{\mathrm{reg}}$ from our comparisons.

As for $\tilde{\theta}_r^{\mathrm{unb}}$, this estimator can be used in the Doeblin chains setting, because $m = 1/\beta$ is known. The bounds for $\tilde{\theta}_r^{\mathrm{unb}}$ in Sect. 3.2 involve $\sigma_{\mathrm{unb}}^2(f)$. Although we cannot provide a rigorous proof, we conjecture that in most practical situations we have $\sigma_{\mathrm{unb}}^2(f) > \sigma_{\mathrm{as}}^2(f)$, because $\rho(\bar{f}, 1)$ in (14) is often close to zero. If this is the case, then the bound for $\tilde{\theta}_r^{\mathrm{unb}}$ is inferior to that for $\hat{\theta}_n^{\text{reg-seq}}$.

6 A Geometric Drift Condition

Using drift conditions is a standard approach for establishing geometric ergodicity. We refer to [31] or [23] for the definition and further details. The assumption below is the same as in [3]. Specifically, let J be the small set which appears in Assumption 1.

Assumption 2 (Drift) *There exist a function $V : \mathscr{X} \to [1, \infty[$, constants $\lambda < 1$ and $K < \infty$ such that*

$$PV(x) := \int_{\mathscr{X}} P(x, \mathrm{d}y)V(y) \le \begin{cases} \lambda V(x) & \text{for } x \notin J, \\ K & \text{for } x \in J, \end{cases}$$

In many papers conditions similar to Assumption 2 have been established for realistic MCMC algorithms in statistical models of practical relevance [7,8,12,16,17,34]. This opens the possibility of computing our bounds in these models.

Under Assumption 2, it is possible to bound $\sigma_{\mathrm{as}}^2(f)$, σ_τ^2 and C_0 which appear in Theorems 1 and 2, by expressions involving only λ, β and K. The following result is a minor variation of Theorem 6.5 in [19].

Theorem 3 *If Assumptions 1 and 2 hold and f is such that $\|\bar{f}\|_{V^{1/2}} := \sup_x |\bar{f}(x)|/V^{1/2}(x) < \infty$, then*

$$\sigma_{\mathrm{as}}^2(f) \le \|\bar{f}\|_{V^{1/2}}^2 \left[\frac{1 + \lambda^{1/2}}{1 - \lambda^{1/2}} \pi(V) + \frac{2(K^{1/2} - \lambda^{1/2} - \beta)}{\beta(1 - \lambda^{1/2})} \pi(V^{1/2}) \right]$$

$$C_0 \le 2 \left[\frac{\lambda^{1/2}}{1 - \lambda^{1/2}} \pi(V^{1/2}) + \frac{K^{1/2} - \lambda^{1/2} - \beta}{\beta(1 - \lambda^{1/2})} \right] + 1.$$

To bound σ_τ^2 we can use the obvious inequality $\sigma_\tau^2 = C_0 - m \le C_0 - 1$. Moreover, one can easily control πV and $\pi V^{1/2}$ and further replace $\|\bar{f}\|_{V^{1/2}}$ e.g., by $\|f\|_{V^{1/2}} + (K^{1/2} - \lambda^{1/2})/(1 - \lambda^{1/2})$, we refer to [19] for details.

Let us now discuss possible approaches to confidence estimation in the setting of this section. Perfect sampling is in general unavailable. For unbounded f we cannot apply exponential inequalities for the standard one-run estimate. Since m is unknown we cannot use $\tilde{\theta}_r^{\mathrm{unb}}$. This leaves $\hat{\theta}_r^{\mathrm{reg}}$ and $\hat{\theta}_n^{\text{reg-seq}}$ combined with the median trick. To analyse $\hat{\theta}_r^{\mathrm{reg}}$ we can apply Theorem 1. Upper bounds for $\sigma_{\mathrm{as}}^2(f)$ and σ_τ^2 are available. However, in Theorem 1 we will also need a *lower bound* on m. Without further assumptions we can only write

$$m = \frac{1}{\pi(J)\beta} \geq \frac{1}{\beta}. \tag{28}$$

In the above analysis (28) is particularly disappointing. It multiplies the bound by an unexpected and substantial factor, as $\pi(J)$ is typically small in applications. For $\hat{\theta}_n^{\text{reg-seq}}$ we have much more satisfactory results. Theorems 2 and 3 can be used to obtain bounds which do not involve m. In many realistic examples, the parameters β, λ and K which appear in Assumptions 1 (Small Set) and 2 (Drift) can be explicitly computed, see e.g., [16, 17, 34].

We note that nonasymptotic confidence intervals for MCMC estimators under drift condition have also been obtained in [20], where identification of regeneration times has not been assumed. In absence of regeneration times a different approach has been used and the bounds are typically weaker. For example one can compare [20, Corollary 3.2] (for estimator $\hat{\theta}_{t,n}^{\text{fix}}$) combined with the bounds in [3] with our Theorems 2 and 3 (for estimator $\hat{\theta}_n^{\text{reg-seq}}$).

Appendix

For convenience, we recall the two identities of Abraham Wald, which we need in the proof of Theorem 2. Proofs can be found e.g., in [6, Theorems 1 and 3 in Sect. 5.3].

Assume that $\eta_1, \dots, \eta_k, \dots$, are i.i.d. random variables and R is a stopping time such that $\mathbb{E}\, R < \infty$.

I Wald identity: If $\mathbb{E}\, |\eta_1| < \infty$ then

$$\mathbb{E} \sum_{k=1}^{R} \eta_k = \mathbb{E}\, R\, \mathbb{E}\, \eta_1.$$

II Wald identity: If $\mathbb{E}\, \eta_1 = 0$ and $\mathbb{E}\, \eta_1^2 < \infty$ then

$$\mathbb{E} \left(\sum_{k=1}^{R} \eta_k \right)^2 = \mathbb{E}\, R\, \mathbb{E}\, \eta_1^2.$$

In Sect. 5 we used the following formula taken from [28, Eq. 4.1.4]. In the notation of our Sects. 2 and 3, for every $g \geq 0$ we have

$$\frac{\mathbb{E}_\nu \Xi_1(g)^2}{m} = \mathbb{E}_\pi g(X_0)^2 + 2 \sum_{i=1}^{\infty} \mathbb{E}_\pi g(X_0) g(X_i) \mathbb{I}(T > i). \tag{29}$$

In [28] this formula, with $g = \bar{f}$, is used to derive an expression for the asymptotic variance $\sigma^2_{\text{as}}(f) = \mathbb{E}_{\nu} \varXi_1(\bar{f})/m$ under the assumption that f is bounded. For $g \geq 0$, the proof is the same.

References

1. Y.F. Atchade, F. Perron (2007): On the geometric ergodicity of Metropolis-Hastings algorithms. *Statistics* 41, 77–84.
2. K.B. Athreya and P. Ney (1978): A new approach to the limit theory of recurrent Markov chains, *Trans. Amer. Math. Soc.* 245, 493–501.
3. P.H. Baxendale (2005): Renewal Theory and Computable Convergence Rates for Geometrically Ergodic Markov Chains. *Ann. Appl. Prob.* 15, 700–738.
4. W. Bednorz, R. Latała and K. Łatuszyński (2008): A Regeneration Proof of the Central Limit Theorem for Uniformly Ergodic Markov Chains. *Elect. Comm. in Probab.* 13, 85–98.
5. L.A. Breyer and G.O. Roberts (2001): Catalytic perfect simulation. *Methodol. Comput. Appl. Probab.* **3** 161–177.
6. Y.S. Chow and H. Teicher (1988): *Probability Theory, Independence, Interchangeability, Martingales*. Second Edition, Springer Verlag.
7. G. Fort and E. Moulines (2000): V-subgeometric ergodicity for a Hastings–Metropolis algorithm. *Statist. Probab. Lett.* 49, 401–410.
8. G. Fort, E. Moulines, G.O. Roberts, and J.S. Rosenthal (2003): On the geometric ergodicity of hybrid samplers. *J. Appl. Probab.* 40 (1), 123–146.
9. W.R. Gilks, S. Richardson, D.J. Spiegelhalter: *Markov chain Monte Carlo in practice.* Chapman & Hall, 1998.
10. P.W. Glynn and D. Ormoneit (2002): Hoeffding's inequality for uniformly ergodic Markov chains, *Statist. Probab. Lett.* 56, 143–146.
11. O. Häggström J.S. Rosenthal (2007): On variance conditions for Markov chain CLTs. *Elect. Comm. in Probab.* 12 , 454–464.
12. J.P. Hobert and C.J. Geyer (1998): Geometric ergodicity of Gibbs and block Gibbs samplers for Hierarchical Random Effects Model. *J. Multivariate Anal.* 67, 414–439.
13. J.P. Hobert, G.L. Jones, B. Presnell, and J.S. Rosenthal (2002): On the Applicability of Regenerative Simulation in Markov Chain Monte Carlo. *Biometrika* 89, 731–743.
14. J.P. Hobert and C.P. Robert (2004): A mixture representation of π with applications in Markov chain Monte Carlo and perfect sampling. *Ann. Appl. Probab.* **14** 1295–1305.
15. M.R. Jerrum, L.G. Valiant, V.V. Vazirani (1986): Random generation of combinatorial structures fro, a uniform distribution. *Theoretical Computer Science* 43, 169–188.
16. G.L. Jones, J.P. Hobert (2004): Sufficient burn-in for Gibbs samplers for a hierarchical random effects model. *Ann. Statist.* 32, pp. 784–817.
17. A.A. Johnson and G.L. Jones (2010): Gibbs sampling for a Bayesian hierarchical general linear model. *Electronic J. Statist.* 4, 313–333.
18. I. Kontoyiannis, L. Lastras-Montano, S.P. Meyn (2005): Relative Entropy and Exponential Deviation Bounds for General Markov Chains. *2005 IEEE International Symposium on Information Theory.*
19. K. Łatuszyński, B. Miasojedow and W. Niemiro (2009): Nonasymptotic bounds on the estimation error for regenerative MCMC algorithms. arXiv:0907.4915v1
20. K. Łatuszyński, W. Niemiro (2011): Rigorous confidence bounds for MCMC under a geometric drift condition. *J. of Complexity* 27, 23–38.
21. G. Lorden: On excess over the boundary. *Ann. Math. Statist.* 41, 520–527, 1970.
22. K.L. Mengersen, L.R. Tweedie (1996): Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.* 24, 1, 101–121.

23. S.P. Meyn and R.L. Tweedie: *Markov Chains and Stochastic Stability.* Springer-Verlag, 1993.
24. P. Mykland, L. Tierney and B. Yu (1995): Regeneration in Markov chain samplers. *J. Am. Statist. Assoc..*, 90, 233–241.
25. R. Neath, G.L. Jones (2009): Variable-at-a-time implementation of Markov chain Monte Carlo. Preprint. arXiv:0903.0664v1
26. W. Niemiro, P. Pokarowski (2009): Fixed precision MCMC Estimation by Median of Products of Averages. *J. Appl. Probab.* 46 (2), 309–329.
27. E. Nummelin (1978): A splitting technique for Harris recurrent Markov chains, *Z. Wahr. Verw. Geb.* 43, 309–318.
28. E. Nummelin (2002): MC's for MCMC'ists, *International Statistical Review*, 70, 215–240.
29. C.P. Robert and G. Casella: *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 2004.
30. G.O. Roberts and J.S. Rosenthal (1997): Geometric ergodicity and hybrid Markov chains. *Elec. Comm. Prob.* 2 (2).
31. G.O. Roberts and J.S. Rosenthal (2004): General state space Markov chains and MCMC algorithms. *Probability Surveys* 1, 20–71.
32. J.S. Rosenthal (1995): Minorization conditions and convergence rates for Markov chains. *J. Amer. Statist. Association* 90, 558–566.
33. D. Rudolf (2008): Explicit error bounds for lazy reversible Markov chain Monte Carlo. *J. of Complexity.* 25, 11–24.
34. V. Roy, J.P. Hobert (2010): On Monte Carlo methods for Bayesian multivariate regression models with heavy-tailed errors. *J. Multivariate Anal.* 101, 1190–1202
35. D.B. Wilson (2000): How to couple from the past using a read-once source of randomness. *Random Structures Algorithms* 16 (1), 85–113.

Accelerating the Convergence of Lattice Methods by Importance Sampling-Based Transformations

Earl Maize, John Sepikas, and Jerome Spanier

Abstract Importance sampling is a powerful technique for improving the stochastic solution of quadrature problems as well as problems associated with the solution of integral equations, and a generalization of importance sampling, called weighted importance sampling, provides even more potential for error reduction. Additionally, lattice methods are particularly effective for integrating sufficiently smooth periodic functions. We will discuss the advantage of combining these ideas to transform non-periodic to periodic integrands over the unit hypercube to improve the convergence rates of lattice-based quadrature formulas. We provide a pair of examples that show that with the proper choice of importance transformation, the order in the rate of convergence of a quadrature formula can be increased significantly.

This technique becomes even more effective when implemented using a family of multidimensional dyadic sequences generally called extensible lattices. Based on an extension of an idea of Soboĺ [17] extensible lattices are both infinite and at the same time return to lattice-based methods with the appropriate choice of sample size. The effectiveness of these sequences, both theoretically and with numerical results, is discussed. Also, there is an interesting parallel with low discrepancy sequences generated by the fractional parts of integer multiples of irrationals which may point the way to a useful construction method for extensible lattices.

E. Maize
Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
e-mail: earl.h.maize@jpl.nasa.gov

J. Sepikas
Pasadena City College, Pasadena, CA 91105, USA

J. Spanier
Beckman Laser Institute and Medical Clinic, University of California, 1002 Health Sciences Road East, Irvine, CA 92612, USA

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_32, © Springer-Verlag Berlin Heidelberg 2012

1 Introduction

The need for estimates of multidimensional integrals is widespread. It is well known nowadays that quasi-Monte Carlo (qMC) methods can (sometimes surprisingly) provide better estimates for these purposes than classical deterministic quadrature formulas or pseudorandom Monte Carlo (MC) methods. All such qMC methods rely on the uniformity of the points selected in the integration domain. A highly desirable feature of any technique for forming such estimates is the possibility of adding sample points without recomputing the previously sampled points. For independent samples (the MC case) this is no problem but for correlated samples (the qMC case) the uniformity, as measured by the discrepancy, tends to be lowered in blocks whose size depends on the algorithm that generates the qMC sequence. Moreover, when the qMC sequence is generated by a conventional lattice rule, extending the sequence requires recomputing all of the sequence elements anew, as we will explain below. In other words, conventional lattice rule methods require deciding *in advance* how many qMC points are needed – an uncomfortable constraint when more points are needed. Overcoming this limitation leads to the notion of *extensible lattice rules*.

Let $\mathbf{g}$ be an s-dimensional vector of integers and form the sequence

$$\mathbf{x}_n = \left\{ \frac{n}{N} \mathbf{g} \right\} \quad n = 0, 1, \ldots, N-1 \tag{1}$$

where the braces indicate the fractional part of each vector component. We are interested in using the $\mathbf{x}_n$ as arguments for the approximation of an integral over the s-dimensional hypercube I^s by a sum:

$$\theta = \int_{I^s} f(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n). \tag{2}$$

The formulas (1) and (2) define a rank-1 lattice rule, an idea that originated with Korobov [8], and has given rise to a great deal of interest in the intervening years, especially in the last 20 years.

It is well known that lattice methods are especially attractive when used to estimate integrals of smooth periodic functions. Consider the class $E_s^\lambda(K)$ of all periodic functions f on I^s whose coefficients in the absolutely convergent Fourier series expansion

$$f(\mathbf{x}) = \sum_{\mathbf{h} \in \mathbf{Z}^s} c_{\mathbf{h}} \exp(2\pi i \, \mathbf{h} \cdot \mathbf{x}) \tag{3}$$

satisfy the decay condition

$$|c_{\mathbf{h}}| \leq K \frac{1}{r(\mathbf{h})^\lambda}. \tag{4}$$

with $\lambda > 1$ and where

$$r(\mathbf{h}) = \max(1, |h_1|)\max(1, |h_2|)\cdots\max(1, |h_s|) \tag{5}$$

and $\mathbf{h} = (h_1, \ldots, h_s) \in \mathbf{Z}^s$.

For such functions, a quick calculation shows that the error in a lattice method may be expressed as

$$\left| \int_{I^s} f(\mathbf{x})d\mathbf{t} - \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n) \right| = \sum_{\mathbf{h}\cdot\mathbf{g}\equiv 0(\mathrm{modN})}{}' c_{\mathbf{h}} \tag{6}$$

$$\leq K \sum_{\mathbf{h}\cdot\mathbf{g}\equiv 0(\mathrm{modN})}{}' r(\mathbf{h})^{-\lambda}. \tag{7}$$

where the prime on the summation indicates that the sum is to taken over all $\mathbf{h} \in \mathbf{Z}^s$ except for $\mathbf{h} = (0, \ldots, 0)$. The infinite sum appearing in (7) is a recurring figure of merit for lattice methods which we denote by

$$P_\lambda(\mathbf{g}, N) \equiv \sum_{\mathbf{h}\cdot\mathbf{g}\equiv 0(\mathrm{modN})}{}' r(\mathbf{h})^{-\lambda}. \tag{8}$$

An excellent survey of lattice methods is in [13]. One can find there that there exist lattice methods whose errors satisfy

$$\left| \int_{I^s} f(\mathbf{x})d\mathbf{t} - \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n) \right| = O(N^{-\lambda}(logN)^{\lambda s}). \tag{9}$$

Note that the error expression in (9) now takes advantage of the additional smoothness of the integrand as represented by the size of λ.

It is clear from (1) that the value of $\mathbf{x}_n$ depends on the choice of the integer N. In his 1981 Ph.D. dissertation Maize [11] observed that certain infinite dyadic sequences (and their generalizations to other prime bases) can be used to define arguments $\mathbf{x}_n$ that do not depend on an a priori choice of N, as they do in the formula (1). This gives rise to the possibility of extending N – point lattice rule quadrature formulas to infinite sequences that revert to lattice rules for specific choices of N. The simplest instance of these in the one dimensional case gives rise to the van der Corput sequence $x_n = \phi_2(n)$ and produces an infinite sequence with the property that when N is any power of two, the points $x_1, \ldots, x_N$ define a lattice. Maize pointed out that such sequences can be easily (and exactly) implemented in a computer and he showed how such sequences might be used to improve upon $(\log N)^s/N$ convergence rates for periodic and sufficiently regular integrands.

Many of the ideas of Maize's dissertation were published in the paper [21]. The idea of extensibility for lattice rules reappeared in a paper by Hickernell and Hong [4] and has subsequently been pursued further in [5, 6] and in other papers. Such infinite sequences, defined with respect to a number base b with the property that for every integer b the first b^m points form a lattice, are now called *extensible lattice rules*. Such sequences, therefore, behave in much the same way as the initial segments of (t,s) sequences do to form (t,m,s) nets [12, 13]. The challenge is to establish the *existence* of extensible lattices with favorable uniformity properties and *exhibit constructive algorithms* for their *effective* computation.

To capitalize on this possibility, we explore the potential advantage in converting nonperiodic to periodic integrands by applying transformations of the sort that are commonly used for other purposes in the context of pseudorandom Monte Carlo. Specifically, we will see that importance sampling transformations are useful candidates for such consideration.

These ideas will be illustrated by applying them to the evaluation of a simple three dimensional integral and a more challenging four dimensional example.

2 Extensible Lattices

The generation of extensible lattices in [11] was inspired by the notion of good direction numbers found in Soboĺ [17], which is itself a generalization of Korobov's development of the theory of good lattice points [8]. The essential motivation is to find a way to preserve the desirable convergence properties of good lattice methods while at the same time maintaining an unlimited supply of sampling points.

2.1 Generation of Extensible Lattices

The general method for defining an extensible lattice is to select an increasing sequence of positive integers $N_1 < N_2 < \ldots$ and generating vectors of integers $\mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \ldots$ such that each finite lattice sequence is nested within the next. That is,

$$\left\{ \frac{n}{N_k} \mathbf{g}^{(k)} \right\}_{n=0}^{N_k - 1} \subset \left\{ \frac{n}{N_{k+1}} \mathbf{g}^{(k+1)} \right\}_{n=0}^{N_{k+1} - 1}. \tag{10}$$

Figure 1 depicts such a nested lattice sequence.

One particular method for accomplishing this is to choose a prime p and let $N_j = p^j$. If we then insist that the generating vectors satisfy $\mathbf{g}^{(l+1)} \equiv \mathbf{g}^{(l)}$ mod (p^l) where the congruence is taken component-wise, it is easily seen that the inclusions (10) are satisfied.

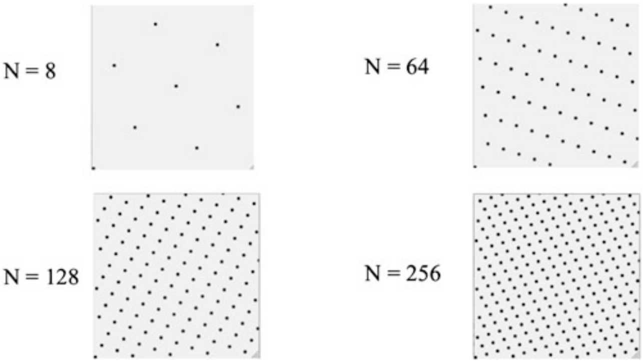


Fig. 1 Nested lattices.

If our only aim were to sample integrands with the fixed sample sizes N_k, $k = 1, 2, \ldots$, the definitions above would be sufficient. However, the practitioner may wish to choose an intermediate sample size, that is a sample size N where $N_k < N < N_{k+1}$. For that we require a way to generate intermediate points in an extended lattice in a manner that distributes them uniformly over I^s. As it turns out, the van der Corput sequence provides an ideal mechanism for accomplishing this.

Since our immediate goal is to explore the practical application of this theory, we will from here on restrict ourselves to dyadic sequences; that is, extensible lattice sequences with $p = 2$. The reader is referred to [5] and [6] for the more general case.

We begin with the $s = 1$ case. Let $\nu^{(k)}, k = 1, 2, \ldots$ be a sequence of integers with $\nu^{(k+1)} \equiv \nu^{(k)} \pmod{2^k}$. For any integer n, represent n in base 2 via $n = \epsilon_1 + \epsilon_2 2 + \cdots + \epsilon_l 2^{l-1}$ and define the nth component of the sequence as

$$x^{(n)} = \left\{\frac{\epsilon_1}{2}\nu^{(1)} + \frac{\epsilon_2}{4}\nu^{(2)} + \cdots + \frac{\epsilon_l}{2^l}\nu^{(l)}\right\}. \tag{11}$$

Since $\nu^{(k+1)} \equiv \nu^{(k)} \pmod{2^k}$ it follows that (11) is equivalent to

$$x^{(n)} = \left\{\phi_2(n)\nu^{(l)}\right\} \text{ where } l = \lfloor \log_2 n \rfloor + 1 \tag{12}$$

and $\phi_2(n)$ is van der Corput's radical inverse function with base 2. Note that the choice $\nu^{(k)} = 1$ for all k produces the classic van der Corput sequence.

Finally, given a sequence of s-dimensional vectors $\boldsymbol{\nu}^{(l)}$ whose components satisfy $\nu_j^{(l+1)} \equiv \nu_j^{(l)} \pmod{2^l}$ we may define the infinite s-dimensional sequence $\mathbf{x}^n$ component-wise via

$$x_j^{(n)} = \frac{\epsilon}{2}\nu_j^{(1)} + \frac{\epsilon_2}{4}\nu_j^{(2)} + \cdots + \frac{\epsilon_l}{2^l}\nu_j^{(l)} \tag{13}$$

or as noted above,

$$x_j^{(n)} = \left\{\phi_2(n)\nu_l^{(n)}\right\} \text{ where } l = \lfloor \log_2 n \rfloor + 1. \tag{14}$$

and then form the s-dimensional sequence via

$$\mathbf{x}^{(n)} = \left\{\phi_2(n)\boldsymbol{\nu}^{(l)}\right\} \tag{15}$$

where the fractional parts are taken component-wise.

Recall that for a prime p, the definition of a p-adic integer is an infinite sequence $\{a^{(1)}, a^{(2)}, \ldots\}$ of integers where $a^{(k+1)} \equiv a^{(k)} \pmod{p^k}$. Denote by O_p the set of all such sequences with the canonical representation $a^{(k+1)} = a^{(k)} + bp^k$ where $b \in \{0, 1, 2, \ldots, p-1\}$. With addition and multiplication defined componentwise (and reduction to canonical form as required), the set O_p, called the *p-adic integers*, is an integral domain and can be imbedded in a field called the *p-adic numbers.* An ordinary integer n is represented in O_p via the sequence $\{n, n, \ldots, n, \ldots\}$. In the context of p-adic integers, ordinary integers are called *rational integers.*

Returning to the definition of the generating vector, we observe that a sequence of integers $\nu^{(k)},\ k = 1, 2, \ldots$ satisfying $\nu^{(k+1)} \equiv \nu^{(k)} \pmod{2^k}$ is simply a 2-adic integer. It is quite straightforward to see [5, 11] that by viewing the binary fractions as naturally imbedded within the 2-adic numbers and applying 2-adic arithmetic, (15) is represented by

$$\mathbf{x}^{(n)} = \{\phi_2(n)\boldsymbol{\nu}\} \tag{16}$$

where $\boldsymbol{\nu}$ is a vector of 2-adic integers.

2.2 Distribution Properties of Extensible Lattices

In the previous section, we described a method for generating extensible lattices which can be compactly expressed via (16). The existence of uniformly distributed extensible lattice sequences is confirmed via the following theorem [11].

Theorem 1. *Let $\nu_1, \ldots \nu_s$ be 2-adic integers and let $\boldsymbol{\nu} = (\nu_1, \ldots \nu_s)$. The s-dimensional infinite sequence $\mathbf{x}^{(n)} = \{\phi_2(n)\boldsymbol{\nu}\}$ is uniformly distributed (see [10]) if and only if $\nu_1, \ldots \nu_s$ are linearly independent over the rational integers.*

The proof is accomplished via the use of Weyl's criterion. Note the very intriguing analog between sequences generated by (16) and sequences of the form $\mathbf{x}^{(n)} = \{n\boldsymbol{\alpha}\}$ where $\boldsymbol{\alpha}$ is an s-dimensional vector of irrational numbers. As it turns out in both cases, the equidistribution of the sequence hinges on the independence of the components of the generating vector over the rational numbers. We will expand upon this observation later.

Clearly there is an ample supply of generating vectors since the cardinality of the 2-adic integers is that of the continuum and any independent set will, in the limit, correctly integrate any function in $E_s^{\lambda}(K)$. The question then turns to the quantitative performance of these sequences. Most recently, Hickernell and

Niederreiter [6] have examined the distribution properties of extensible lattices with respect to several figures of merit. Their result germane to our discussion here is summarized in the following theorem.

Theorem 2. *For a given dimension s and any $\epsilon > 0$, there exist 2-adic generating vectors $\boldsymbol{\nu}$ and a constant $C(\lambda, \epsilon, s)$ such that*

$$P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l) \leq C(\lambda, \epsilon, s) 2^{-\lambda l} (\log 2^l)^{\lambda(s+1)} [\log\log(2^l + 1)]^{\lambda(1+\epsilon)} \tag{17}$$

for $l = 1, 2, \ldots$.

where the figure of merit $P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l)$ is defined in (8). Considering the very slowly increasing log(log) term, we see that, comparing (17) with (7), the potential penalty for requiring that the lattices be nested is only slightly more than an additional factor of $\log(2^l)^\lambda$.

2.3 Construction of Generating Vectors

From the previous sections, we know that uniformly distributed extensible lattices not only exist, but at least in theory, have distribution properties that are worse by only a factor slightly larger than $\log(N)^\lambda$ when compared with the best known results for general lattice methods. Unfortunately, these results are based on averaging techniques and none provides an explicit representation for good generating vectors. We are left in the position of having a good theoretical method but with no practical path to implemention.

One idea for the construction of generating vectors is a "bootstrap" method whereby one picks an appropriate figure of merit and an initial guess for a generating vector. One then examines the figure of merit for all possible candidate generating vectors of the form $\boldsymbol{\nu}^{(l+1)} \equiv \boldsymbol{\nu}^{(l)} \pmod{p^l}$. A potential pitfall of this method of course, is that, while at each step it does guarantee that the next component of the generating vector will be optimal with respect to the previous choices, it does not guarantee *global* optimality. Maize [11] numerically explored this technique for $\lambda = 2$ and $p = 2$. It was further studied by Hickernell et al. [5]. Niederreiter and Pillichshammer [14] have examined this method in the more general context of weighted Korobov spaces using several different figures of merit and have provided some very positive results regarding this process. For the figure of merit $P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l)$ and remaining in base 2, the algorithm in [14] may be described as follows:

Step 1: Set $\boldsymbol{\nu}^{(1)} = (1, 1, \ldots, 1)$
Step 2: For $k = 2, 3, \ldots$, choose $\boldsymbol{\nu}^{(k)} = \boldsymbol{\nu}^{(k-1)} + 2^{k-1}\boldsymbol{\epsilon}$, where $\epsilon_j = 0, 1$, so that $P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l)$ is minimized. The following theorem appears in [14].

Theorem 3. *With the algorithm above,*

$$P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l) = \sum_{\mathbf{h}\cdot\boldsymbol{\nu}^{(\mathbf{l})}=0(\mathrm{mod} 2^{\mathrm{l}})}{}' \quad r(\mathbf{h})^{-2} \le \Big(\prod_{j=1}^{s}(1+2\zeta(\lambda)) - 1\Big)\min\Big(l, \frac{2^{\lambda-1}}{2^{\lambda-1}-1}\Big)\frac{1}{2^l} \tag{18}$$

where ζ is the Riemann zeta function.

We can see that there is a gap between this algorithm's performance and the best results of Sect. 2.2. In particular, we have lost, except in the constant multiplier, any dependence on the smoothness of the integrand, λ. As noted in Maize [11] and Niederreiter and Pillichshammer [14], numerical evidence suggests that there is substantial room for improvement. We present some of the evidence in the next section.

2.4 Numerical Investigations

For the figure of merit $P_\lambda(\boldsymbol{\nu}^{(l)}, 2^l)$ we have implemented the algorithm in the previous section for the choice $\lambda = 2$ and for dimensions $s = 1, 2, \dots, 10$ and for sample sizes up to 2^{28}. The normalized results multiplied by $N = 2^l$ are plotted in Fig. 2. Examining the curves in the figure, it is clear that the error term is approaching zero more rapidly than 2^{-l} suggested by the best known theoretical results for the algorithm. The numerical results appear much more promising. In fact, referring to Fig. 3 where the same results are normalized by the convergence

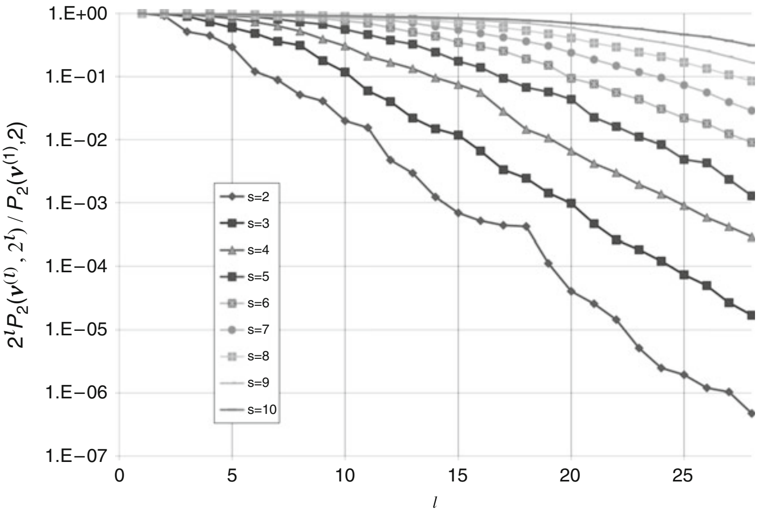


Fig. 2 Normalized figure of merit $2^l * P_2(\boldsymbol{\nu}^{(l)}, 2^l)$.

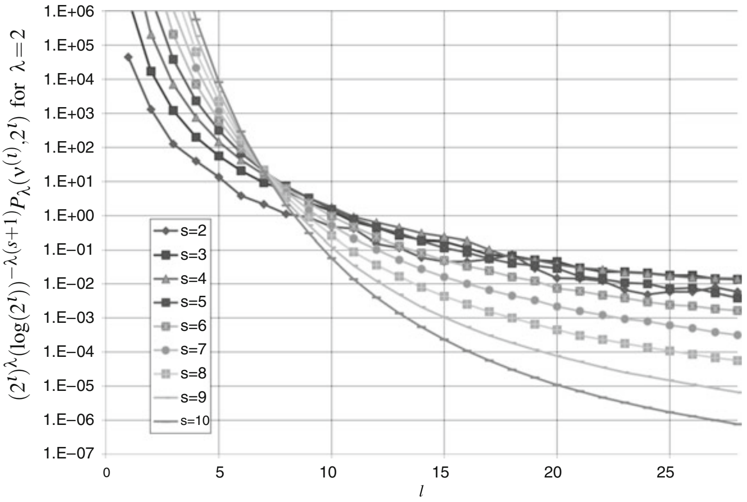


Fig. 3 Normalized figure of merit $(2^l)^{\lambda}(\log(2^l))^{-\lambda(s+1)}P_{\lambda}(\nu^{(l)}, 2^l)$ for $\lambda = 2$.

rates inferred from Theorem 2, it may not be too reckless to conjecture that there are generating vectors that will produce a convergence rate of

$$\left|\int_{I^s} f(\mathbf{x})d\mathbf{t} - \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n)\right| = O((logN)^{\lambda s}/N^{\lambda}) \text{ for } f \in E_s^{\lambda}(K) \qquad (19)$$

and $N = 2^l$.

3 Integration of Periodic and Non-periodic Functions

Traditionally, importance sampling [3, 19] provides a standard Monte Carlo technique for reducing the variance in pseudorandom estimates of an integral and in solving integral equations. It achieves this by evaluating the integrand at points that are nonuniformly distributed and reweighting the integrand values to eliminate the bias engendered by the nonuniform sampling. The method attempts to generate sample points preferentially in subdomains more "important" in the sense of their relative contribution to estimates of the integral under consideration. Here we want to use the methodology of importance sampling, and its generalization to weighted uniform sampling [16,18], not to redistribute the sample points, but rather to convert a nonperiodic integrand to one that is periodic and smooth. Our interest in doing so is to gain access to the higher rates of convergence promised by lattice methods when applied to smooth periodic integrands.

3.1 Theoretical Convergence Rates

Standard MC methods converge, of course, at the rate forecast by the central limit theorem. Thus, as the number N of samples increases the integration error (as measured by the standard deviation, or the relative standard deviation, of the sample mean) reduces asymptotically as $O(N^{-1/2})$ for any $\mathscr{L}_2$ integrand. For qMC methods there are various measures of the error, frequently referred to as *figures of merit,* such as the quantity P_λ that we have consistently used in this paper. Other figures of merit are introduced elsewhere in the literature, and a more extensive discussion can be found, e.g., in [14].

3.2 Use of Importance Sampling to Periodicize

Given the improved convergence rates for lattice methods when applied to smooth periodic functions, it seems reasonable to investigate whether quadrature problems, especially those in high dimensions, can benefit from conversion of nonperiodic to periodic integrands. This is not a new idea; it was discussed already in [7, 9, 22]. More recently, the book [13] provides a number of other references related to this topic.

In Paul Chelson's 1976 dissertation [2], see also [20, 21], a primary focus was to make rigorous the possibility of applying quasirandom methods to the estimation of finite dimensional integrals and solutions of matrix and integral equations. The latter problems are infinite dimensional in the sense that the underlying sample space is infinite dimensional. Further, if that could be shown, Chelson wanted to know whether variance reduction techniques, such as importance sampling, could be useful in the qMC context. Chelson found that this is, indeed, the case and he established a generalized Koksma-Hlawka inequality for both the finite and infinite dimensional instances in which the term $V(f)$ involving the variation of the integrand is replaced by $V(f/g)$, where g plays the role of an importance function. This gives rise to the possibility that the function g can be chosen in such a way that $V(f/g) \ll V(f)$. Such an idea could then improve the estimate of the integral, but it does not increase the *rate of convergence* of the sum to the integral. In [11] Maize generalized Chelson's results to weighted importance sampling [16, 18] and establised a Koksma-Hlawka inequality in which the variation of the integrand $V(f)$ is replaced by $V((f - \theta h)/g)$ where h is a positive weighting function.

These results established that these methods, so useful for MC applications, might offer similar advantages in the qMC context. Whereas importance sampling requires the selection of a suitable importance function, weighted uniform sampling offers much greater flexibility. The reader is referred to [21] for a discussion of these ideas.

For the estimation of s-dimensional integrals, the importance sampling formulation chooses a (usually) nonuniform distribution function $G(\mathbf{x})$ on I^s and uses the estimate

$$\theta = \int_{I^s} f(\mathbf{t})d\mathbf{t} \approx \frac{1}{N}\sum_{n=1}^{N} \frac{f(G^{-1}(\mathbf{x}_n))}{g(G^{-1}(\mathbf{x}_n))} \tag{20}$$

in place of

$$\int_{I^s} f(\mathbf{t})d\mathbf{t} \approx \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n), \tag{21}$$

where $g(\mathbf{x})$ is the probability density function corresponding to G. Instead of choosing G to minimize the variance (or variation in the qMC instance) of the estimator $f(\mathbf{x})/g(\mathbf{x})$, we can choose G to convert f to a smooth periodic function and hopefully take advantage of higher order convergence rates. More generally according to Maize [11] (see also [21]), we can select a nonnegative weighting function $h(\mathbf{t})$ whose integral over I^s is 1 and use

$$\int_{I^s} f(\mathbf{t})d\mathbf{t} \approx \sum_{n=1}^{N} \frac{f(G^{-1}(\mathbf{x}_n))}{g(G^{-1}(\mathbf{x}_n))} \Big/ \sum_{n=1}^{N} \frac{h(G^{-1}(\mathbf{x}_n))}{g(G^{-1}(\mathbf{x}_n))} \tag{22}$$

in place of (21).

For example, a simple way to "periodicize" an integrand is via the Bernstein polynomials

$$\frac{1}{g(x)} = B_\alpha(x) = Kx^\alpha(1-x)^\alpha \tag{23}$$

with a normalizing constant K, which is clearly a periodic function on I^s. With such a definition, it is a simple matter to calculate $G^{-1}(x)$.

A potential danger of this method is that the Jacobian of the resulting transformation can produce much larger derivatives than those of f, adversely affecting the error bounds. While we might obtain a better convergence rate, the implied constants multiplying the error term may have grown to the point where we have lost the advantage. An additional complication is that computing G^{-1} can be quite cumbersome for practical problems. This is where the choice of the weighting function h can be used. As stated above, [11] (see also [21]) provides a bound for the error in weighted uniform sampling that is proportional to $V((f-\theta h)/g)$. From this we can see that if h is chosen to mimic the behavior of the integrand, for example by choosing h to be a low order approximation of f that is readily integrated, it can both relieve the requirement that g closely mimic the integrand and at the same time, reduce the constant multipliers in the error term.

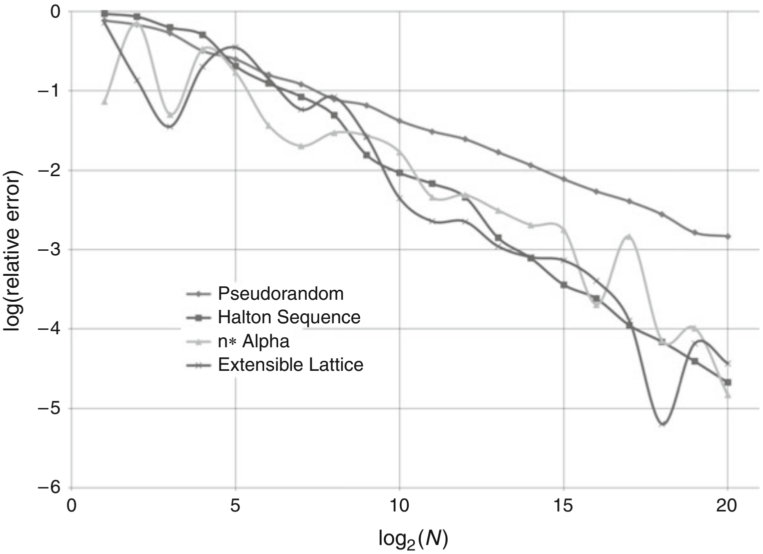


Fig. 4 Pseudorandom and quasi-random estimators.

4 Examples

Let us suppose that we wish to evaluate the integral of the function $f(x, y, z) = 4x^2yze^{xy}$ over the three-dimensional unit cube. This integral is easily evaluated to

$$\theta = \int\limits_{I^3} 4x^2yze^{xy}\,dxdydz = 2(3-e). \tag{24}$$

Figure 4 constrasts the performance of MC and a few qMC estimators in estimating this integral. The quadrature errors incurred from the use of a pseudorandom sequence, the Halton sequence, an $\{n\boldsymbol{\alpha}\}$ sequence, and the 3-dimensional extensible lattice sequence based on the numerical alogrithms from Sect. 2.4 are plotted versus sample size. One can easily see the rather slow $\sqrt{N}$ performance of the pseudorandom technique and the much better performance of the quasi-random sequences.

We now focus on using an extensible lattice and the techniques from Sect. 3.2 to improve the performance of the estimates. We choose a very simple one-dimensional importance function for each variable based on the first order Bernstein polynomial

$$G^{-1}(t) = 3t^2 - 2t^3 \tag{25}$$

where $t = x, y$, or z. A simple calculation shows that

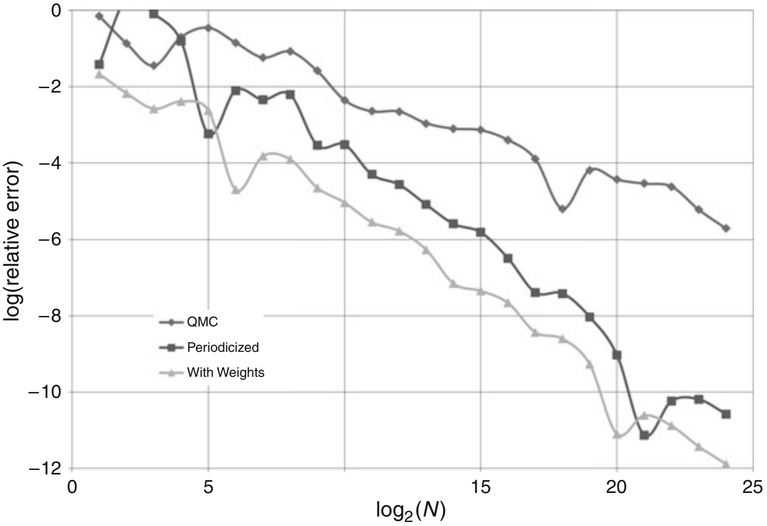


Fig. 5 qMC, importance sampling, and weighted importance sampling with an extensible lattice.

$$\frac{f(G^{-1}(\mathbf{x}))}{g(G^{-1}(\mathbf{x}))} = 6^3 xyz(1-x)(1-y)(1-z)f(G^{-1}(\mathbf{x})). \tag{26}$$

For a weighting function, we will mimic the behavior of f by approximating the exponential term with the first three terms of the appropriate Taylor series. After performing the appropriate normalizations, we obtain

$$h(x,y,z) = \frac{80}{11}x^2yz(1+xy+\frac{(xy)^2}{2}). \tag{27}$$

Figure 5 illustrates the results of these numerical studies. We can see that using importance sampling to periodicize the integrand does, indeed, result in an improved convergence rate. In addition, the use of the weighting function with importance sampling maintains the improved convergence rate while improving the constant multiplier. This fairly modest example shows that when the integrand is sufficiently regular, the technique of weighted importance sampling, implemented with extensible lattices, can be an effective technique for reducing error in qMC computations.

As a second example, let us consider the 4-dimensional integral

$$\theta = \int_{R^4} \left(1 + \|\mathbf{x}\|^2\right)^{1/2} e^{-\|\mathbf{x}\|^2} d\mathbf{x}. \tag{28}$$

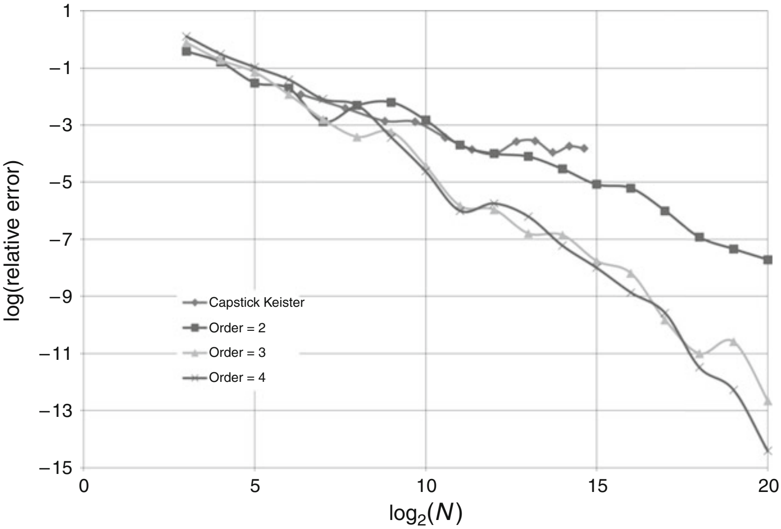


Fig. 6 Periodization with variable order Bernstein polynomials.

which has been studied in [1] and [15]. A change of variables yields the equivalent integral

$$\theta = \int_{I^4} \left(1 + \sum_{j=1}^{4} (\varphi^{-1})^2(t_j)/2\right)^{1/2} d\mathbf{t} \tag{29}$$

where φ is the cumulative normal distribution function. For this example, we will consider varying the order of the Bernstein polynomials used to "periodicize" the integrand. Figure 6 compares the relative errors derived from our method ($\alpha = 2, 3, 4$) with those based on the Genz-Patterson method used [1] for reference. Again in this more challenging example we see the benefits of our approach. Our second order method performs as well and the higher order methods outperform the Genz-Patterson quadrature method.

5 Summary and Future Work

We have described early efforts to develop infinite sequences of points in I^s whose initial segments of length 2^m form a series of ever larger lattices; sequences now called extensible lattice sequences. Aside from their attractiveness for estimating finite dimensional integrals, in Sect. 3.2 we mentioned that extensible lattice sequences can also be used to solve infinite dimensional problems such as those characterized by matrix and integral equations. The sequences introduced

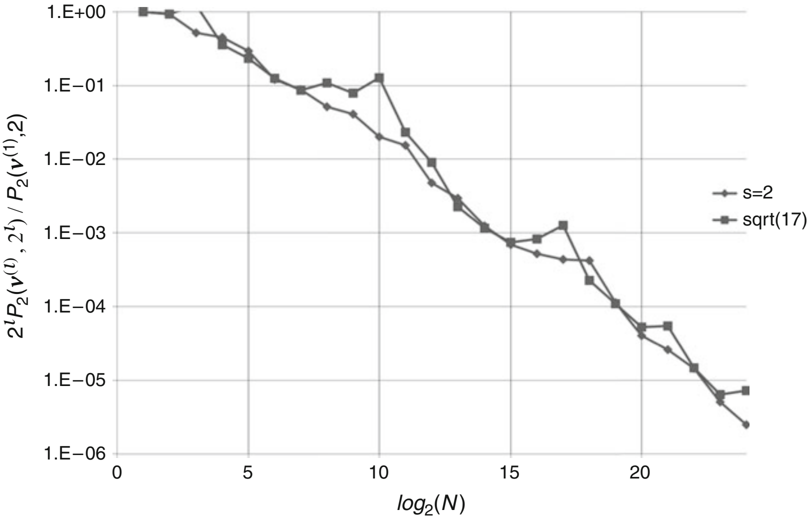


Fig. 7 Normalized figure of merit for the generator $\boldsymbol{\nu} = (1, \sqrt{17})$.

in Sect. 2.1 seem well suited to this task and we hope to report in a subsequent publication on our efforts to explore this idea.

A second area of future work is to fill any part of the gap between the best currently known theoretical convergence rates and the often more optimistic evidence provided in various numerical experiments, including our own.

Finally, we plan to investigate the possible value of constructive methods for extensible lattices that make use of p-adic irrationals, as suggested by Theorem 1. We close with one final piece of evidence that this last idea may be a fruitful approach.

Recall from Theorem 1 that a sufficient condition for uniform distribution of a sequence of the form (16) is that the components of the generating vector $\boldsymbol{\nu}$ viewed as 2-adic integers must be linearly independent over the rationals. The first quadratic irrational in the 2-adic numbers is $\sqrt{17}$ whose first few terms as a 2-adic integer are given by $\sqrt{17} = \{1, 3, 7, 7, 23, 23, 23, 23, 279, \ldots\}$. We can select the generating vector $\boldsymbol{\nu} = (1, \sqrt{17})$ in two dimensions and generate the sequence as defined in (16). Figure 7 plots the normalized figure of merit $P_2(\boldsymbol{\nu}, 2^l)$ alongside the result from the optimum generating vector for $s = 2$. As Fig. 7 clearly shows, while initially not as good as the vector found by the exhaustive search, $\boldsymbol{\nu} = (1, \sqrt{17})$ is indeed an effective generator of an extensible lattice for $s = 2$.

Acknowledgements The first two authors wish to dedicate this paper to their friend and mentor, Dr. Jerome Spanier on the occasion of his 80th birthday. The last author gratefully acknowledges partial support from the Laser Microbeam and Medical Program (LAMMP), an NIH Biomedical Technology Resource Center (P41-RR01192). The authors would also like to thank the referees for helpful remarks and suggestions that improved the manuscript.

References

1. Capstick, S. and Keister. B.D.: Multidimensional Quadrature Algorithms at Higher De- gree and/or Dimension, Journal of Computational Physics, 123, 267–273, 1996.
2. Chelson, P.: Quasi–random techniques for Monte Carlo methods, PhD Dissertation, The Claremont Graduate School, 1976.
3. Hammersley, J. M. and Handscomb, D. C.: Monte Carlo Methods, Methuen, 1964.
4. Hickernell, F. J., Hong, H. S.: Computing multivariate normal probabilities using rank-1 lattice sequences, in Proceedings of the Workshop on Scientific Computing, Hong Kong, 1997, G. H. Golub, S. H. Lui, F. T. Luk, and R. J. Plemmons, eds., Springer-Verlag, Singapore, 1997, pp. 209–215
5. Hickernell, F.J., Hong, H.S., L'Ecuyer, P., Lemieux, C.: Extensible Lattice Sequences for Quasi-Monte Carlo Quadrature, SIAM J. Sci. Comp., 22, (2000), pp. 1117–1138.
6. Hickernell, F.J. and Niederreiter, H.: The existence of good extensible rank-1 lattices, J. Complex., 19, (2003), pp. 286–300.
7. Hua, L. K., Wang, Y.: Applications of Number Theory to Numerical Analysis, Springer Verlag, Berlin, 1981.
8. Korobov, N. M.: Computation of multiple integrals by the method of optimal coefficients. Vestnik Muskov. Univ. Ser. Mat. Meh. Astr. Fiz. Him., no. 4 (1959), pp. 19–25 (Russian).
9. Korobov, N. M.: Number–Theoretic Methods in Approximate Analysis, Fizmatig, Moscow, 1963, (Russian).
10. Kuipers, L. and Niederreiter, H.: Uniform Distribution of Sequences, Wiley, New York, 1974.
11. Maize, E. Contributions to the theory of error reduction in quasi–Monte Carlo methods, PhD Dissertation, The Claremont Graduate School, 1981.
12. Niederreiter, H.: Point sets and sequences with small discrepancy, Monatsch. Math., 104 (1987), pp. 273–337.
13. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. CBMS-NSF Series in Applied Mathematics, vol. 63. SIAM, Philadelphia, 1992.
14. Niederreiter, H. and Pillichshammer, F.: Construction Algorithms for Good Extensible Lattice Rules, Constr. Approx., 30, (2009), pp. 361–393.
15. Papageorgiou, A.: Fast Convergence of Quasi-Monte Carlo for a Class of Isotropic Integrals, Mathematics of Computation, 70, Number 233, pp. 297–306, 2000.
16. Powell, MJD, Swann, J.: Weighted uniform sampling – a Monte Carlo technique for reducing variance, J. Inst. Maths. Applica., 2 (1966), pp. 228–236.
17. Soboĺ, I.M.: The distribution of points in a cube and the approximate evaluation of integrals, Z. Vycisl Mat.i. Mat Fiz., 7, 784–802 = USSR Computational Math and Math Phys., 7 (1967), pp. 86–112.
18. Spanier, J., A new family of estimators for random walk problems, J. Inst. Maths. Applica., 23 (1979), pp. 1–31.
19. Spanier, J. and Gelbard, E.M., Monte Carlo Principles and Neutron Transport Problems, Addison–Wesley Pub. Co., Inc., Reading, Mass., 1969
20. Spanier, J., Li, L.: Quasi-Monte Carlo Methods for Integral Equations, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Proc. Conf. at University of Salzburg, Austria, July 9–12, 1996, H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof, eds., Springer Lecture Notes on Statistics #127, 1998
21. Spanier, J., Maize, E. H.: Quasi-random methods for estimating integrals using relatively small samples, SIAM Rev., 36 (1994), pp. 18–44.
22. Zaremba, S.K., ed.: Applications of Number Theory to Numerical Analysis, Academic Press, New York, 1972.

Exact Simulation of Occupation Times

Roman N. Makarov and Karl Wouterloot

Abstract A novel algorithm for the exact simulation of occupation times for Brownian processes and jump-diffusion processes with finite jump intensity is constructed. Our approach is based on sampling from the distribution function of occupation times of a Brownian bridge. For more general diffusions we propose an approximation procedure based on the Brownian bridge interpolation of sample paths. The simulation methods are applied to pricing occupation time derivatives and quantile options under the double-exponential jump-diffusion process and the constant elasticity of variance (CEV) diffusion model.

1 Introduction

Consider a filtered probability space $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, \mathbb{P})$. For a stochastic process $\mathbf{S} = (S_t)_{t\geq 0}$, adapted to the filtration $(\mathscr{F}_t)_{t\geq 0}$, the occupation times below and above level $L \in \mathbb{R}$ from time 0 to $T > 0$ are respectively defined as follows:

$$A_T^{L,-}(\mathbf{S}) \equiv \int_0^T \mathbf{1}_{S_t \leq L}\, \mathrm{d}t \;\; (\text{below } L) \;\; \text{and} \;\; A_T^{L,+}(\mathbf{S}) \equiv \int_0^T \mathbf{1}_{S_t > L}\, \mathrm{d}t \;\; (\text{above } L). \tag{1}$$

The occupations times $A_T^{L,\pm}$ are nonnegative quantities and satisfy $A_T^{L,+} + A_T^{L,-} = T$. We will also use the notation $A_{[u,v]}^{L,+} \equiv \int_u^v \mathbf{1}_{S_t > L}\, \mathrm{d}t$ and $A_{[u,v]}^{L,-} \equiv \int_u^v \mathbf{1}_{S_t \leq L}\, \mathrm{d}t$ to denote the occupation times on an arbitrary time interval, $[u, v]$, $0 \leq u < v$.

Note that a strictly monotone transformation of a process does not change the distribution of occupation times. Suppose the process $\mathbf{X} = (X_t)_{t\geq 0}$ is obtained by applying a strictly monotone mapping X to the process $\mathbf{S}$, i.e. $X_t = \mathsf{X}(S_t)$ for $t \geq 0$.

R.N. Makarov · K. Wouterloot
Department of Mathematics, Wilfrid Laurier University, Waterloo, Ontario, Canada
e-mail: rmakarov@wlu.ca; kwouterloot@gmail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_33, © Springer-Verlag Berlin Heidelberg 2012

Then, $A_t^{L,\pm}(\mathbf{S}) \overset{d}{=} A_t^{\ell,\pm}(\mathbf{X})$, for every $t > 0$, where $\ell = \mathsf{X}(L)$. In the paper, we consider two asset pricing models that can be mapped to other simpler organized processes. In particular, Kou's model (Sect. 3) is an exponential Lévy process; the CEV diffusion model (Sect. 4) is a power-type transformation of the CIR model.

There have been numerous papers published on the distribution of occupation times for Brownian motion with and without drift. By using the Feynman-Kac formula, the joint density function of the occupation time and terminal asset value was obtained in [14] and [19] (see also [5]). A similar approach was used in [13] to derive the distribution function of the occupation time for a standard Brownian bridge from 0 to 0. Analytical pricing formulae for occupation time derivatives under the constant elasticity of variance (CEV) diffusion models are obtained in [18]. However, a numerical implementation of those results is difficult.

In this paper, we generalize the result of [13]. For one important case, we are able to express the cumulative distribution functions (c.d.f.'s) of occupation times in terms of the error function and elementary functions. This result allows us to apply the inverse c.d.f. method for the efficient Monte Carlo simulation of occupation times for various (jump-)diffusion processes.

Consider a market consisting of three securities: a risk-free bond with the price process $(B_t = B_0 \mathrm{e}^{rt})_{t\geq 0}$, a risky asset with the price process $(S_t)_{t\geq 0} \in \mathbb{R}_+ \equiv [0,\infty)$, and an occupation-time-related option contingent upon the asset. There are a large number of different derivatives whose payoff functions depend on occupation times of an asset price process. In this paper, we are interested in claims $f^{\pm}$ whose payoff is of the form $f^{\pm} = f(S_T, A_T^{L,\pm})$, for some function $f : \mathbb{R}_+ \times [0,T] \to \mathbb{R}_{\pm}$.

Assume there exists an equivalent probability measure (e.m.m. for short) $\widetilde{\mathbb{P}}$ such that the discounted asset price process $(\mathrm{e}^{-rt} S_t)_{t\geq 0}$ is a $\widetilde{\mathbb{P}}$-martingale. The arbitrage free price processes $(V_t^{f,\pm})_{0\leq t\leq T}$ of the claims $f^{\pm}$ are thus defined by

$$V_t^{f,\pm} = \mathrm{e}^{-r(T-t)} \widetilde{\mathbb{E}}\left[f(S_T, A_T^{L,\pm}) \mid \mathscr{F}_t \right]. \tag{2}$$

Step options were first proposed as an alternative to barrier options in [19]. The payoff functions of the proportional step call and step put options are respectively given by $f_{\text{step}}^{\text{call}}(S_T, A_T) = (S_T - K)^+ \mathrm{e}^{-\rho A_T}$ and $f_{\text{step}}^{\text{put}}(S_T, A_T) = (K - S_T)^+ \mathrm{e}^{-\rho A_T}$, where $\rho \geq 0$, and the occupation time A_T in these formulae is given by (1).

As one can see, the payoff function of a step option works under the same principles as knock-and-out barrier options, but with less risk. If a step down option is purchased, the holder's payout will be discounted by the occupation time below L, provided that the process $\mathbf{S}$ does hit L before time T. Letting $\rho \to 0+$, a step option becomes a vanilla European option. Letting $\rho \to \infty$, the payoff of a step option becomes that of a barrier option, since $\lim_{\rho\to\infty} \exp\left(-\rho A_T^{L,-}\right) = \mathbf{1}_{\inf_{0\leq t\leq T} S_t > L}$ a.s..

The payoff functions of the fixed-strike call and floating strike put α-quantile options are respectively defined by $(M_T^{\alpha}(\mathbf{S}) - K)^+$ and $(M_T^{\alpha}(\mathbf{S}) - S_T)^+$, where $M_T^{\alpha}(\mathbf{S}) \equiv \inf\{L : A_T^{L,-} \geq \alpha T\}$ is known as the α-quantile $(0 < \alpha < 1)$. The α-quantile options may be viewed as a generalization of lookback options.

There is a remarkable relationship between the α-quantile of a Lévy process and the distribution of the maximum and minimum values of the process obtained in [11]. Let $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$ be independent copies of a process $\mathbf{X}$ with stationary and independent increments and with $X_0 = Y_0 = 0$. Then, there is the following equivalence in distribution:

$$\begin{pmatrix} X_t \\ M_t^{\alpha}(\mathbf{X}) \end{pmatrix} \overset{d}{=} \begin{pmatrix} X_{\alpha t} + Y_{(1-\alpha)t} \\ \sup_{0\leq s\leq \alpha t} X_s + \inf_{0\leq s\leq (1-\alpha)t} Y_s \end{pmatrix}. \tag{3}$$

In this paper, we consider the simulation of occupation times and quantiles for jump-diffusion models and nonlinear solvable diffusions. The main application is the pricing of occupation-time and α-quantile options. As a result, we obtain efficient Monte Carlo algorithms for two asset pricing models, namely, the Kou jump-diffusion model [16] and the CEV diffusion model [10]. Our approach can be easily extended to other Lévy models with finite jump intensity as well as to other solvable state-dependent volatility diffusion models [7].

2 Occupation Times of a Brownian Bridge

Let $(W_t^x)_{t\geq 0}$ denote the Brownian motion starting at $x \in \mathbb{R}$. The Brownian bridge $W_{[0,T]}^{x,y}$ from x to y over $[0,T]$ is defined by $W_{[0,T]}^{x,y}(t) \overset{d}{=} \left\{W_t^x \mid W_T^x = y\right\}$, $0 \leq t \leq T$.

Theorem 1. *The c.d.f.* $F_\ell^+(\tau; y) \equiv \mathbb{P}\left\{A_1^{\ell,+}(W_{[0,1]}^{0,y}) \leq \tau\right\}$, $0 < \tau < 1$, *of the occupation time above level* ℓ *for a Brownian bridge from* 0 *to* y *over* $[0,1]$ *is given by the following cases.*

Case (I) *For* $y \leq \ell$ *and* $\ell \geq 0$,

$$F_\ell^+(\tau; y) = 1 - \frac{2\sqrt{\tau}}{\pi}\mathrm{e}^{\frac{y^2}{2}} \int_\tau^1 \mathrm{e}^{-\frac{(2\ell-y)^2}{2(1-u)}} \frac{\sqrt{u-\tau}}{u^2\sqrt{1-u}}\,\mathrm{d}u \tag{4}$$

$$= 1 - (1-\tau)\mathrm{e}^{-\frac{b}{\tau}+\frac{y^2}{2}} \left(\mathrm{e}^b(2b+1)\,\mathrm{erfc}\,(\sqrt{b}) - 2\sqrt{\frac{b}{\pi}}\right), \tag{5}$$

where $b = \frac{2(\ell-y/2)^2\tau}{1-\tau}$.

Case (II) *For* $0 \leq \ell < y$,

$$F_\ell^+(\tau; y) = \int_0^\tau \frac{(\tau-u)\mathrm{e}^{\frac{y^2}{2}-\frac{\ell^2}{2(1-u)}-\frac{(y-\ell)^2}{2u}}}{\sqrt{2\pi}(u(1-u))^{\frac{3}{2}}} \times \left(\frac{\ell(y-\ell)^2}{u} - \frac{(y-\ell)^2\ell^2}{1-u} + y - 2\ell\right)\mathrm{d}u. \tag{6}$$

Case (III) *For* $\ell < 0$, $F_\ell^+(\tau; y) = 1 - F_{-\ell}^+(1-\tau; -y)$.

Proof. The case with $x = y = 0$ was done in [13]. For the general case, the argument is almost exactly the same. First, we consider Case (I). Let $f^{t,x}(\tau|y)$ denote the p.d.f. of $A_t^{\ell,+}(W^x)$ conditional on $W_t^x = y$, $0 \le \tau \le t$, $x, y \in \mathbb{R}$. Note that $f^{1,0}(\tau|y) = \frac{\partial}{\partial\tau} F_\ell^+(\tau; y)$. The Fourier transform and double Laplace transform of the joint p.d.f. for $A_t^{\ell,+}$ and W_t^x is $\frac{u(x;p,\lambda,\beta)}{\sqrt{2\pi}} \equiv \mathscr{F}_y\left[\mathscr{L}_t\left[\mathscr{L}_\tau[f^{t,x}(\tau|y)\frac{1}{\sqrt{2\pi t}}\mathrm{e}^{-(y-x)^2/2t}; \beta]; \lambda\right]; p\right]$. By the Feynman-Kac formula, u is a unique solution to $\frac{1}{2}u''(x) - (\lambda + \beta\mathbf{1}_{x>\ell})u(x) = -\mathrm{e}^{ipx}$, subject to conditions $u(\ell-) = u(\ell+)$ and $u'(\ell-) = u'(\ell+)$. From [13], when $x = 0$ we have that $u(0) = \frac{-4\beta\left(\sqrt{2(\lambda+\beta)}+ip\right)}{(2\lambda+p^2)(2(\lambda+\beta)+p^2)} \frac{\exp\left(-\ell\sqrt{2\lambda}+ip\ell\right)}{\sqrt{2(\lambda+\beta)}+\sqrt{2\lambda}} + \frac{2}{2\lambda+p^2}$. Applying the inverse Fourier transform, we obtain that

$$\mathscr{L}_t\left[\mathscr{L}_\tau\left[f^{t,0}(\tau|y)\frac{\mathrm{e}^{-\frac{y^2}{2t}}}{\sqrt{2\pi t}}; \beta\right]; \lambda\right] = \mathscr{F}_p^{-1}\left[\frac{u(0; p, \lambda, \beta)}{\sqrt{2\pi}}; y\right]$$
$$= \frac{\mathrm{e}^{-y\sqrt{2\lambda}}}{\sqrt{2\lambda}} - \frac{\sqrt{\lambda+\beta} - \sqrt{\lambda}}{\sqrt{\lambda+\beta}+\sqrt{\lambda}}\mathrm{e}^{(y-2\ell)\sqrt{2\lambda}}. \tag{7}$$

Taking the inverse Laplace transform of both sides of (7), we obtain

$$1 - \mathscr{L}_\tau\left[f^{1,0}(\tau|y); \beta\right] = \mathrm{e}^{\frac{y^2}{2}} \int_0^1 \frac{\mathrm{e}^{-\frac{\beta u}{2}} I_1\left(\frac{\beta u}{2}\right) \mathrm{e}^{-\frac{(2\ell-y)^2}{2(1-u)}}}{u\sqrt{1-u}} \mathrm{d}u. \tag{8}$$

Integration by parts gives $1 - \mathscr{L}_\tau\left[f^{1,0}(\tau|y); \beta\right] = \beta\mathscr{L}_\tau\left[1 - F_\ell^+(\tau; y); \beta\right]$. Applying the identity $I_1(z/2) = \frac{2z\mathrm{e}^{z/2}}{\pi}\int_0^1 \sqrt{v(1-v)}\mathrm{e}^{-zv}\mathrm{d}v$ in (8), changing the order of integration, and changing variables $uv = \tau$, we obtain (4) by uniqueness of the Laplace transform. Changing variables $u = \frac{\tau+\tau x^2}{1+\tau x^2}$ and simplifying the integral obtained, we arrive at (5).

The proof of Case (II) follows a similar argument. From [13], we obtain the formula for $u(0)$. Taking the inverse Fourier transform and then the double inverse Laplace transform, we obtain (6). The derivation can be done by using tables of Fourier transform and Laplace transform pairs and the shift theorem. Case (III) follows by symmetry of the Brownian motion. □

Here, the complementary error function, denoted *erfc*, is defined as

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \mathrm{e}^{-u^2}\,\mathrm{d}u.$$

Since $A_t^{\ell,-} + A_t^{\ell,+} = t$ holds for every $t \geq 0$, the c.d.f. F_ℓ^- of the occupation time $A_1^{\ell,-}(W_{[0,1]}^{0,y})$ is given by $F_\ell^-(\tau) = 1 - F_\ell^+(1-\tau)$, $0 \leq \tau \leq 1$. Note that if $x = y = \ell = 0$, then $A_1^{0,\pm}(W_{[0,1]}^{0,0}) \sim \text{Uniform}(0,1)$ (see [5]).

The c.d.f.'s $F_\ell^\pm$ for an arbitrary time interval of length T can be obtained from the respective c.d.f.'s for the time interval of length 1 thanks to the property

$$\mathbb{P}\left\{A_T^{\ell,\pm} \leq t \mid W_0 = x, W_T = y\right\} = \mathbb{P}\left\{A_1^{\frac{\ell}{\sqrt{T}},\pm} \leq \frac{t}{T} \mid W_0 = \frac{x}{\sqrt{T}}, W_1 = \frac{y}{\sqrt{T}}\right\}.$$

By using the symmetry properties of the Brownian motion, we can evaluate the c.d.f. of $A_T^{\ell,\pm}(W_{[0,T]}^{x,y})$ for the general case with arbitrary x, y, and ℓ. The following equalities in distribution are valid:

$$A_T^{\ell,\pm}\left(W_{[0,T]}^{x,y}\right) \stackrel{d}{=} A_T^{\ell,\mp}\left(W_{[0,T]}^{2\ell-x,2\ell-y}\right) \stackrel{d}{=} A_T^{\ell-x,\pm}\left(W_{[0,T]}^{0,y-x}\right) \stackrel{d}{=} A_T^{-\ell,\mp}\left(W_{[0,T]}^{-x,-y}\right).$$

In Theorem 1, we obtain the c.d.f. of the occupation time above level ℓ for a Brownian motion pinned at points x and y at times 0 and 1, respectively. In practice, the c.d.f. for the case where both x and y lie on one side with respect to the level ℓ can be computed more easily than for the other case. For example, if $x = 0$, $y \leq \ell$, and $\ell \geq 0$, then the c.d.f. of $A_1^{\ell,+} = A_1^{\ell,+}(W_{[0,1]}^{0,y})$ given in (5) is expressed in terms of the complimentary error function, which is fast and easy to compute. Therefore, one can use the inverse c.d.f. method to draw the occupation time $A_1^{\ell,+}$. Note that there is a non-zero probability that the Brownian bridge $W_{[0,1]}^{0,y}$ with $y < \ell$ does not cross level $\ell > 0$. Thus, the probability $\mathbb{P}\{A_1^{\ell,+} = 0\}$ is not zero in this case. From (5), we obtain $\mathbb{P}\{A_1^{\ell,+} = 0\} = F_\ell^+(0; y) = 1 - \mathrm{e}^{-2\ell(\ell-y)}$, which is also the probability that the Brownian bridge $W_{[0,1]}^{0,y}$ does not hit the level ℓ.

We also need to consider the other case where the Brownian motion is pinned at points x and y that lie on the opposite sides of the barrier ℓ. For example, if $x = 0$ and $0 \leq \ell < y$, then c.d.f. of $A_1^{\ell,+}$ is given by the integral in (6), which is computationally expensive to evaluate during the simulation process when parameters randomly change. To overcome this difficulty, we propose a two-step procedure. First, we sample the first hitting time $\tau_\ell \in (0,T)$ at the barrier ℓ of the Brownian bridge $W_{[0,T]}^{x,y}$, where $x < \ell < y$ or $y < \ell < x$. Then, we sample the occupation time of the Brownian bridge from ℓ to y over $[\tau_\ell, T]$. Since the new bridge starts at the level ℓ, the c.d.f. of the occupation time can be reduced to the integral in (5). Recall that the first hitting time (f.h.t. for short) τ_ℓ of a diffusion process $(X_t)_{t\geq 0}$ with almost surely continuous paths is defined by $\tau_\ell(x) = \inf\{t > 0 : X_t = \ell \mid X_0 = x\}$. The c.d.f. F_ℓ^τ of the f.h.t. τ_ℓ, $\ell > 0$, of the Brownian bridge $W_{[0,T]}^{0,y}$, $\ell < y$, is given entirely in terms of error functions, which are quick to compute. It has the following form for $0 < t < T$ (see [4]):

$$F_\ell^\tau(t;y) = \mathbb{P}\{\tau_\ell \le t \mid W_0 = 0, W_T = y\} = \mathbb{P}\{\max_{0\le s\le t} W_s \ge \ell \mid W_0 = 0, W_T = y\}$$
$$= \frac{1}{2}\mathrm{e}^{-\frac{2\ell(\ell-y)}{T}}\operatorname{erfc}\left(\frac{\ell T-(2\ell-y)t}{\sqrt{2}}\right) + \frac{1}{2}\operatorname{erfc}\left(\frac{\ell T - yt}{\sqrt{2tT(T-t)}}\right). \tag{9}$$

We obtain Algorithm 5 for sampling $A_T^{\ell,\pm}$, where we assume $\ell \ge x$. In the case of $\ell < x$, one can use the equality in distribution: $A_T^{\ell,\pm}(W_{[0,T]}^{x,y}) \overset{d}{=} T - A_T^{-\ell,\pm}(W_{[0,T]}^{-x,-y})$.

Algorithm 5 Sampling occupation times $A_T^{\ell,\pm}$ for a Brownian Bridge $W_{[0,T]}^{x,y}$

```
input $x, y, T > 0, \ell \ge x$
set $y_1 \leftarrow \frac{y-x}{\sqrt{T}},\ \ell_1 \leftarrow \frac{\ell-x}{\sqrt{T}}$
if $y_1 \le \ell_1$ then
    sample $U \sim \mathrm{Uniform}(0,1)$
    set $A \leftarrow \sup\{t \in [0,1] : F_{\ell_1}^+(t;y_1) < U\}$
    set $A_T^{\ell,+} = A \cdot T,\ A_T^{\ell,-} = T - A_T^{\ell,+}$
else
    sample i.i.d. $U, V \sim \mathrm{Uniform}(0,1)$
    set $\tau \leftarrow \sup\{t \in [0,1] : F_{\ell_1}^\tau(t;y_1) < V\}$
    set $y_2 \leftarrow \frac{y_1-\ell_1}{\sqrt{1-\tau}}$
    set $A \leftarrow 1 - \sup\{t \in [0,1] : F_0^+(t;-y_2) < U\}$
    set $A_T^{\ell,+} \leftarrow A \cdot (1-\tau) \cdot T,\ A_T^{\ell,-} \leftarrow T - A_T^{\ell,+}$
end if
return $A_T^{\ell,\pm}$
```

Note that the bridge distribution of a Brownian motion with drift, $\{W_t + \nu t,\ t \ge 0\}$, is the same as that of a standard Brownian motion. Thus, the distributions of occupation times will not change with introducing a non-zero drift (see [5]).

3 Pricing Occupation Time Options Under a Jump Diffusion

In this section, we propose an algorithm for the exact simulation of occupation times of a Lévy process that has a Gaussian component and a jump component of compound Poisson type. Suppose the stock price is governed by the following dynamics:

$$\frac{\mathrm{d}S_t}{S_{t-}} = \nu\,\mathrm{d}t + \sigma\,\mathrm{d}W_t + \mathrm{d}\left(\sum_{i=1}^{N_t}(V_i - 1)\right), \quad S_{t=0} = S_0 > 0, \tag{10}$$

where ν and σ are constants, $(W_t)_{t\ge 0}$ is a standard Brownian motion, $(N_t)_{t\ge 0}$ is a Poisson process with arrival rate λ, and $\{V_i\}_{i=1,2,\ldots}$ is a sequence of independent identically distributed (i.i.d.) random variables. We assume that (W_t), (N_t), and $\{V_i\}$ are jointly independent.

As an example, we consider Kou's double exponential jump diffusion model [16], where the random variables $Y_i = \ln(V_i)$ follows a double exponential distribution with the p.d.f. $f_Y(y) = p\eta_+ \mathrm{e}^{-\eta_+ y}\mathbf{1}_{y\geq 0} + (1-p)\eta_- \mathrm{e}^{-\eta_-|y|}\mathbf{1}_{y<0}$, where $\eta_+ > 1$, $\eta_- > 0$, $p \in [0,1]$. There are two types of jumps in the process: upward jumps (with occurrence probability p and average jump size $\frac{1}{\eta_+}$) and downward jumps (with occurrence probability $1-p$ and average jump size $\frac{1}{\eta_-}$). Both types of jumps are exponentially distributed.

Algorithm 6 Simulation of a sample path, occupation times, and extremes for a jump-diffusion model (S_t)

```
input: moments of jumps $\mathscr{T}_1 < \ldots < \mathscr{T}_N$ on $[0,T]$ and values $\{X_{k-}, X_k\}_{k=1,\ldots,N}$,
       where $X_k = X(\mathscr{T}_k)$, $X_{k-} = X(\mathscr{T}_k -)$, and $X(t) \equiv \frac{1}{\sigma}\ln S_t$
set $m_0^X \leftarrow 0$, $M_0^X \leftarrow 0$,
for $n$ from 1 to $N$ do
    sample $A_n = A^{\ell,+}_{[\mathscr{T}_{n-1},\mathscr{T}_n]}\left(W^{X_{n-1},X_{n-}}_{[\mathscr{T}_{n-1},\mathscr{T}_n]}\right)$
    sample $U_n, V_n \sim \text{Uniform}(0,1)$
    set $m_n^X \leftarrow \min\{m_{n-1}^X, X_{n-1} + \frac{1}{2}\left(B_n - \sqrt{B_n^2 - 2\Delta\mathscr{T}_n \ln U_n}\right)\}$
    set $M_n^X \leftarrow \max\{M_{n-1}^X, X_{n-1} + \frac{1}{2}\left(B_n + \sqrt{B_n^2 - 2\Delta\mathscr{T}_n \ln V_n}\right)\}$
end for
set $S_T \leftarrow S_0\mathrm{e}^{\sigma X_N}$, $m_T \leftarrow S_0\mathrm{e}^{\sigma m_N^X}$, $M_T \leftarrow S_0\mathrm{e}^{\sigma M_N^X}$
set $A_T^{L,+} \leftarrow \sum_{n=1}^N A_n$, $A_T^{L,-} \leftarrow T - A_T^{L,+}$
return $S_T$ and only one of $A_T^{L,\pm}$, $m_T$, $M_T$
```

The stochastic differential equation (s.d.e. for short) in (10) can be solved analytically. Under an e.m.m. $\widetilde{\mathbb{P}}$, we have that $\nu = r - \lambda\zeta$, where $\zeta = \widetilde{\mathbb{E}}[\mathrm{e}^Y - 1]$ is given by $\zeta = \frac{p\eta_+}{\eta_+ - 1} + \frac{(1-p)\eta_-}{\eta_- + 1} - 1$, and $S_t = S_0 \exp\left((r - \frac{\sigma^2}{2} - \lambda\zeta)t + \sigma W_t + \sum_{i=1}^{N_t} Y_i\right)$ (see [17]). Note that $\widetilde{\mathbb{P}}$ can be obtained by using the Esscher transform.

A Lévy process with finite jump intensity behaves like a Brownian motion between successive jumps. The simulation scheme is well known (e.g., see [9]). First, we sample the time and size of each jump occurred on $[0,T]$. Second, we sample the Brownian increment for each time-interval between successive jumps. The only addition to this scheme is the sampling of occupation times. As a result, we obtain Algorithm 6. We can also sample the minimum value m_T and the maximum value M_T of a Lévy sample path. These values are used for pricing α-quantile options thanks to the property in (3). Notice that Algorithm 6 is implemented in a way so that it allows the user to sample the extreme values and the occupation times from their correct marginal distributions, but with an improper joint distribution. Therefore, only one quantity from the list $\{m_T, M_T, A_T^{L,\pm}\}$ can be used after each execution of Algorithm 6. This is sufficient for our applications. To sample an α-quantile option payoff, the user needs to run the algorithm twice to obtain independent sample values of the maximum and minimum. It is possible to sample m_T and M_T from their joint distribution, but the joint distribution of occupation times and extremes for a Brownian bridge is not available to the best of our knowledge.

4 Pricing Occupation Time Options Under the CEV Model

Simulation of path-dependent variables such as the running minimum/maximum and occupation times is a challenging computational problem for general stochastic processes. In the case of Brownian motion (and its derivatives) with or without a compound Poisson component, exact simulation algorithms can be constructed by using the Brownian bridge interpolation. This procedure suggests an approximation for more general diffusions.

Consider a discrete-time skeleton of a sample path. Its continuous-time approximation can be obtained by interpolating over each subinterval using independent Brownian bridges. Such an approach can be used to approximately simulate the minimum and maximum and barrier crossing probabilities (see [1, 3, 15]), however resulting estimates of path-dependent quantities are biased. We apply this idea to approximately simulate occupation times of the constant elasticity of variance (CEV) diffusion for which an exact path sampling algorithm is available in [21].

4.1 Exact Simulation of the CEV Process

The CEV diffusion $\mathbf{S} = (S_t)_{t \ge 0} \in \mathbb{R}_+$ follows $\mathrm{d}S_t = \nu S_t \,\mathrm{d}t + \delta S_t^{\beta+1} \,\mathrm{d}W_t$, $S_{t=0} = S_0 > 0$, where $\delta > 0$ and $\nu \in \mathbb{R}$. Under the e.m.m. $\widetilde{\mathbb{P}}$, we have that $\nu = r$. Here we assume that $\beta < 0$, hence the boundary $s = 0$ of the state space $[0, \infty)$ is regular. Here we consider the case where the endpoint $s = 0$ is a killing boundary. Let τ_0 denote the first hitting time at zero. We assume that $S_t = 0$ for all $t \ge \tau_0$.

The CEV process is a transformation of the Cox-Ross-Ingersoll (CIR) diffusion model $\mathbf{X} = (X_t)_{t \ge 0}$ that follows $\mathrm{d}X_t = (\lambda_0 - \lambda_1 X_t)\,\mathrm{d}t + 2\sqrt{X_t}\,\mathrm{d}W_t$ (see [5]). Indeed, by using Itô's formula, it is easy to show that the mapping $\mathsf{X}(s) \equiv (\delta|\beta|)^{-2} s^{-2\beta}$ (which is strictly increasing since $\beta < 0$) transforms a CEV process into a CIR process with $\lambda_0 = 2 + \frac{1}{\beta}$ and $\lambda_1 = 2\nu\beta$, i.e. $X_t = \mathsf{X}(S_t)$. Moreover, the CIR process can be obtained by a scale and time transformation of the square Bessel (SQB) process. Also note that the radial Ornstein-Uhlenbeck (ROU) process $\mathbf{Z} = (Z_t)_{t \ge 0}$, obeying the s.d.e. $\mathrm{d}Z_t = \left(\frac{\lambda_0 - 1}{2Z_t} - \frac{\lambda_1 Z_t}{2}\right)\mathrm{d}t + \mathrm{d}W_t$, can be obtained by taking the square root of the CIR process, i.e. $Z_t = \sqrt{X_t}$.

The literature on simulating the CIR and other related processes is rather extensive (e.g., see [15] and references therein). However, most of existing algorithms either are approximation schemes or deal with the case without absorption at zero. In [21], a general exact sampling method for Bessel diffusions is presented.

The sampling method allows one to exactly sample a variety of diffusions that are related to the SQB process through scale and time transformations, change of variables, and by change of measure. These include the CIR, CEV, and hypergeometric diffusions described in [7]. The paths of the CEV and CIR processes can be sampled simultaneously at time moments $\{t_i\}_{i=0}^{N}$, $0 = t_0 < t_1 < \ldots < t_N$ conditional on $S_{t=0} = S_0$ as outlined below.

1. Apply Algorithm 7 to sample a path of the SQB process $\mathbf{Y}$ with index $\mu = \frac{1}{2\beta}$, at time points $\{u_i = u(t_i; \lambda_1 = 2\nu\beta)\}_{i=0}^{N}$ conditional on $Y_0 = \mathsf{X}(S_0)$. Here we define $u(t;\lambda_1) = \frac{e^{\lambda_1 t} - 1}{\lambda_1}$ if $\lambda_1 \neq 0$, and $u(t;\lambda_1) = t$ if $\lambda_1 = 0$.
2. Use the scale and time transformation to obtain sample paths of the CIR model $\mathbf{X}$ as follows: $X_{t_i} \equiv e^{\lambda_1 t_i} Y_{u_i}$ for each $i = 0, 1, \ldots, N$.
3. Transform by using the mapping $S_{t_i} = \mathsf{X}^{-1}(X_{t_i})$, $i = 1, \ldots, N$, to obtain a discrete-time sample path of the CEV process $\mathbf{S}$.

Algorithm 7 Simulation of an SQB sample path

The sequential sampling method conditional on the first hitting time at zero, τ_0, for modelling an SQB process with absorption at the origin (see [21]).

```
input $Y_0 > 0$, $0 = u_0 < u_1 < \ldots < u_N$, $\mu < 0$
sample $G \sim \text{Gamma}(|\mu|, 1)$; set $\tau_0 \leftarrow \dfrac{Y_0}{2G}$
for $n$ from $1$ to $N$ do
  if $u_n < \tau_0$ then
    sample $P_n \sim \text{Poisson}\left(\dfrac{Y_{u_{n-1}}(\tau_0 - u_n)}{2(\tau_0 - u_{n-1})(u_n - u_{n-1})}\right)$
    sample $Y_{u_n} \sim \text{Gamma}\left(P_n + |\mu| + 1, \dfrac{\tau_0 - u_{n-1}}{(\tau_0 - u_n)(u_n - u_{n-1})}\right)$
  else
    set $Y_{u_n} \leftarrow 0$
  end if
end for
return $(Y_0, Y_{u_1}, \ldots, Y_{u_N})$
```

4.2 Simulation of Occupation Times for the CEV Processes

The CEV process $\mathbf{S}$ can be obtained by applying a monotone transformation to the ROU process $\mathbf{Z}$ and vice versa. Indeed, $Z_t = \sqrt{\mathsf{X}(S_t)}$, $t \geq 0$. The diffusion coefficient of the s.d.e. describing the ROU process equals one. Therefore, (Z_t) can be well approximated by a drifted Brownian motion on short time intervals. If (Z_t) is pinned at times t_{i-1} and t_i that are close enough together, the process will behave like a Brownian motion pinned at the same times. Therefore, on short time intervals $[t_1, t_2]$, the occupation times of the CEV process conditional on $S_{t_i} = s_i > 0$, $i = 1, 2$, can be approximated by occupation times of a Brownian bridge, i.e.

$$\left(A^{L,\pm}_{[t_1,t_2]}(\mathbf{S}) \mid S_{t_1} = s_1, S_{t_2} = s_2\right) \overset{d}{=} \left(A^{\ell,\pm}_{[t_1,t_2]}(\mathbf{Z}) \mid Z_{t_1} = z_1, Z_{t_2} = z_2\right)$$
$$\overset{d}{\approx} \left(A^{\ell,\pm}_{[t_1,t_2]}(\mathbf{W}) \mid W_{t_1} = z_1, W_{t_2} = z_2\right),$$

where $\ell = \sqrt{\mathsf{X}(L)}$, $L > 0$, and $z_i = \sqrt{\mathsf{X}(s_i)}$, $i = 1, 2$. Note that numerical tests demonstrate that the Brownian bridge interpolation procedure produces more accurate estimates of occupation times if it is applied to the ROU process rather than the CEV diffusion. Alternatively, one can use a piecewise-linear approximation of continuous-time sample paths of the ROU process to approximate occupation times. The latter approach can also be used to compute α-quantiles of a sample path. A more rigorous stochastic analysis of such approximation approaches is the matter of our future research.

Since the origin is an absorbing boundary, occupation times only need to be simulated until the maturity T or τ_0, whichever comes first. For arbitrary $T > 0$ and $L > 0$, we have $A^{L,+}_T(\mathbf{S}) = A^{L,+}_{T\wedge\tau_0}(\mathbf{S})$ and $A^{L,-}_T(\mathbf{S}) = T - A^{L,+}_T(\mathbf{S})$. Our strategy for the approximate sampling of occupation times $A^{L,\pm}_{T\wedge\tau_0}(\mathbf{S})$ for the CEV process works as follows.

1. For a given time partition $\{0 = t_0 < t_1 < \ldots < t_N = T \wedge \tau_0\}$, draw a sample CEV path, $S_{t_1}, \ldots, S_{t_N}$, conditional on S_0 and $\tau_0 = \tau_0(S_0)$.
2. Obtain the respective sample path of the ROU process by using the transformation $Z_{t_i} = \sqrt{\mathsf{X}(S_{t_i})}$ for each $i = 0, 1, \ldots, N$.
3. Sample the occupation times of $A^{\ell,\pm}_{[t_{i-1},t_i]}$ for the Brownian bridge from $Z_{t_{i-1}}$ to Z_{t_i} over $[t_{i-1}, t_i]$ for each $i = 1, \ldots, N$. Here, $\ell = \sqrt{\mathsf{X}(L)}$.
4. Obtain the approximation: $A^{L,\pm}_{t_N}(\mathbf{S}) \approx \sum_{i=1}^N A^{\ell,\pm}_{[t_{i-1},t_i]}$.

4.3 The First Hitting Time Approach

There is another approach that can speed up the pricing of occupation time options. Suppose $S_0 > L$ and consider an option whose payoff depends on $A^{L,-}_T$. By using the fact that the events $\{A^{L,-}_T = 0\}$ and $\{\tau_L > T\}$, where $\tau_L = \tau_L(S_0)$ is the first hitting time down at L, are equivalent, we can rewrite the no-arbitrage price of the option as follows:

$$\begin{aligned} \mathrm{e}^{-rT}\widetilde{\mathbb{E}}\left[f(S_T, A^{L,-}_T)\right] &= \mathrm{e}^{-rT}\widetilde{\mathbb{E}}\left[f(S_T, 0)\mathbf{1}_{A^{L,-}_T=0}\right] + \mathrm{e}^{-rT}\widetilde{\mathbb{E}}\left[f(S_T, A^{L,-}_T)\mathbf{1}_{A^{L,-}_T>0}\right] \\ &= \mathrm{e}^{-rT}\widetilde{\mathbb{E}}\left[f(S_T, 0)\mathbf{1}_{\tau_L>T}\right] + \mathrm{e}^{-rT} p_T \widetilde{\mathbb{E}}\left[f(S_T, A^{L,-}_T) \mid \tau_L \le T\right]. \end{aligned} \tag{11}$$

where the probability $p_T = \mathbb{P}\{\tau_L < T\} = \mathbb{P}\{A^{L,-}_T > 0\}$ can be computed by using results of [20]. Notice that the first term in (11) is the no-arbitrage price for a down-and-out barrier option. The analytical price of the down-and-out barrier option under the CEV model is well known (see [12]). Thus, the first term in (11) can be computed analytically, while the second term can be estimated by the Monte Carlo method.

First, we sample the first hitting time down τ_L with the condition $\tau_L \leq T$. The c.d.f. of the first hitting time down is given by the spectral expansion (see [20]). It is computationally expensive to evaluate such an expansion, thus the c.d.f. of τ_L should be computed once on a fine partition of $[0, T]$ and stored in memory. After that, the inverse c.d.f. method is applied to sample τ_L conditional on $\{\tau_L \leq T\}$. Second, we sample $A_T^{L,-}$. Since the process $\mathbf{S}$ first hits the level L at τ_L, the only time that $\mathbf{S}$ can spend below L occurs after τ_L. Therefore, the process need not be sampled on the interval $[0, \tau_L]$, since we only need the occupation time below L and the terminal asset price to compute the payoff of an option. Alternatively, one can use the f.h.t. approach to speed up the sampling of the occupation times thanks to the following property: $A_{[0,T]}^{L,-}(S_{t=0} = S_0) \overset{d}{=} \mathbf{1}_{\tau_L \leq T} \times A_{[\tau_L,T]}^{L,-}(S_{t=\tau_L} = L)$.

5 Numerical Results

As a test case for Algorithm 6, prices of some proportional step down options with payoffs depending on $A_T^{L,-}$ and α-quantile options were computed. First, we consider pricing under Kou's model. The parameters used in simulations were $S_0 = 100$, $T = 1$ (years), $r = 0.05$, $\sigma = 0.3$, $\lambda = 3$, $p = 0.5$, $\eta_+ = 30$, $\eta_- = 20$, $\rho = 1$, and $L = 102$. Monte Carlo *unbiased* estimates of proportional step option prices were computed for a range of strikes with $N = 10^6$ trials; the results are given in Table 1. In all tables below, s_N denotes the sample standard deviation of the Monte Carlo estimate. Also, all of the simulations in this section were implemented in MATLAB® 7.10.0, and they were run on a Intel Pentium® 4 1.60 GHz processor with 3 GB of RAM.

Simulations of α-quantiles under Kou's model were performed using the exact sampling algorithm. Monte Carlo *unbiased* estimates of fixed strike α-quantile option prices were obtained for various values of K and σ from $N = 10^6$ trials. The other model parameters used in these simulations are $S_0 = 100$, $T = 1$, $r = 0.05$, $\lambda = 3$, $p = 0.6$, $\eta_+ = 34$, $\eta_- = 34$, and $\alpha = 0.2$. The results of these simulations are given in Table 2. Tables 1 and 2 contain the exact prices of the occupation time

Table 1 The Monte Carlo unbiased estimates of proportional step down call option prices under Kou's model are tabulated for various K and S_0. The parameters are $T = 1$, $r = 0.05$, $\sigma = 0.2$, $\lambda = 3$, $p = 0.5$, $\eta_+ = 30$, $\eta_- = 20$, $\rho = 1$, $L = 102$, and $N = 10^6$.

K	$S_0 = 100$		$S_0 = 105$	
	Estimate$\pm s_N$	Exact	Estimate$\pm s_N$	Exact
90	13.7715±0.0173	13.81883	19.0374±0.0207	19.04025
100	9.3901±0.0147	9.42438	13.4581±0.0181	13.45927
110	5.9558±0.0120	5.97929	8.9005±0.0152	8.90134

Table 2 The Monte Carlo unbiased estimates of α-Quantile call option prices for various K and σ under Kou's model are tabulated for $\alpha = 0.2$. The parameters used are $S_0 = 100$, $T = 1$, $r = 0.05$, $\lambda = 3$, $p = 0.6$, $\eta_+ = 34$, $\eta_- = 34$, and $N = 10^6$.

K	$\sigma = 0.2$		$\sigma = 0.3$	
	Estimate$\pm s_N$	Exact	Estimate$\pm s_N$	Exact
90	6.9982±0.0074	6.98492	6.7290±0.0092	6.72912
100	2.0793±0.0043	2.08466	2.6993±0.0060	2.69358
110	0.3666±0.0018	0.37724	0.8643±0.0034	0.86545

Table 3 The Monte Carlo biased estimates of proportional step down call and put prices under the CEV model using the Brownian bridge interpolation method are tabulated for various values of K. The parameters used are $S_0 = 100$, $T = 1$, $r = 0.1$, $\delta = 2.5$, $\beta = -0.5$, $\rho = 0.5$, $L = 90$, $\Delta t = 0.05$, and $N = 10^6$.

K	Step calls		Step puts	
	Estimate $\pm s_N$	Exact	Estimate $\pm s_N$	Exact
90	20.9939±0.0009	20.9939	2.0416±0.0006	2.0382
100	14.8192±0.0006	14.8172	4.1988± 0.0010	4.1938
110	9.8621±0.0004	9.8600	7.5750±0.0017	7.5689

options taken from [6]. We can observe the perfect agreement between the Monte Carlo estimates and the exact values.

Monte Carlo *biased* estimates of proportional step down option prices under the CEV model were also computed. This was done using the exact CEV path sampling algorithm together with the Brownian bridge approximation or the piecewise linear path interpolation. To reduce the variance, the estimator of a standard European option price was used as a control variate. The Monte Carlo estimates of option prices are compared with the analytical estimates obtained in [8]. Recall that the origin is an absorbing boundary for the CEV model. If the asset price process hits the zero boundary before the maturity date, then the asset goes to bankruptcy and a derivative on the asset becomes worthless. Thus, the payoff function is given by

$$f^{\pm}_{\text{step}}(A^{L,\pm}_T(\mathbf{S}), S_T) = \mathrm{e}^{-\rho A^{L,\pm}_T(\mathbf{S})} f(S_T)\mathbf{1}_{T<\tau_0}.$$

The estimates were obtained from averaging over $N = 10^6$ samples, with a time step of $\Delta t = 0.05$. These approximate prices are given in Table 3, for a range of K, and with $\rho = 0.5$, $L = 90$ and $T = 1$. The CEV model parameters used in all simulations were $\delta = 2.5$, $\beta = -0.5$, and $r = 0.1$.

The Brownian bridge interpolation and linear interpolation sampling methods were compared for various values of Δt. To do this, Monte Carlo estimates of the step call prices were computed using both of these sampling methods for a range of Δt and with $N = 5 \cdot 10^6$ trials. The results of these simulations are shown in Table 4. As is seen from the table, the Brownian bridge interpolation method works quite accurately even for $\Delta t = 0.5$ (i.e. a sample skeleton only consists of two points).

Table 4 The Monte Carlo biased estimates of proportional step call prices under the CEV model obtained with the use of the Brownian bridge approximation and linear interpolation methods are compared for decreasing time steps Δt. The parameter values used in the simulations are $S_0 = 100$, $T = 1$, $r = 0.1$, $\delta = 2.5$, $\beta = -0.5$, $\rho = 0.5$, $L = 90$, $K = 100$, $N = 5 \cdot 10^6$. The analytical estimate of the option price is 14.8172.

Δt	Bridge interpolation		Linear interpolation	
	Estimate$\pm s_N$	Time (s)	Estimate$\pm s_N$	Time (s)
0.5	14.8184±0.0003	19,160	14.8451±0.0004	7,105
0.25	14.8191±0.0003	38,815	14.7906±0.0004	15,685
0.1	14.8194±0.0003	78,770	14.7931±0.0003	41,835
0.05	14.8191±0.0006	142,150	14.8046±0.0003	76,560

Table 5 The Monte Carlo biased estimates of proportional step down put option prices under the CEV model using the Brownian bridge approximation are tabulated for various values of K. Also, the regular path sampling algorithm (a) is compared to the accelerated first hitting time sampling algorithm (b). The other parameters used are $S_0 = 100$, $T = 1$, $r = 0.1$, $\delta = 2.5$, $\beta = -0.5$, $\rho = 0.5$, $L = 90$, $\Delta t = 0.05$, and $N = 10^6$.

K	Estimate (a) $\pm s_N$	Estimate (b)$\pm s_N$	Exact
90	2.0394±0.0006	2.0407±0.0004	2.0382
100	4.1927±0.0010	4.1974±0.0008	4.1938
110	7.5616±0.0017	7.5730±0.0012	7.5689
	(Time is 28,430 s)	(Time is 26,875 s)	

Finally, the first hitting time method was used to price the proportional step down put option under the CEV model. These simulations used the exact CEV path sampling algorithm along with the Brownian bridge approximation method for $N = 10^6$ trials and with $\Delta t = 0.05$. The prices were computed for a range of K, and they are given in Table 5 along with their standard errors. As is seen from the table, the cost of the f.h.t. method is twice less than that of the regular algorithm. The cost of a MCM algorithm is defined as a product of the sample variance and the computational time.

6 Conclusions

In this paper, we study the simulation of occupation times of Brownian processes, jump diffusions, and state-dependent volatility diffusion models. An efficient algorithm for the exact sampling of occupation times of a Brownian bridge is presented. It is used for the exact simulation of occupation times for Kou's jump-diffusion model. We apply this method to pricing occupation time derivatives. Also, a similar algorithm is designed for pricing quantile options. The sampling method is efficient and can be extended to general Lévy processes. It works for any finite activity process provided that an exact path simulation algorithm is available. Infinite activity Lévy processes can be treated by replacing small jumps with a diffusion term (e.g., see [2]).

By using the Brownian bridge interpolation of a general diffusion process, we obtain an approximate sampling algorithm for occupation times of the CEV diffusion model. The prices of proportional step options are computed by the Monte Carlo method. The approach can be extended to other types of occupation time derivatives and also to other solvable diffusion models (e.g., see [7]).

Acknowledgements The first author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) for a Discovery Research Grant. We thank the two anonymous reviewers for their comments, which helped to improve the paper.

References

1. Andersen, L., Brotherton-Ratcliffe, R.: Exact Exotics. Risk **9**, 85–89 (1996)
2. Asmussen, S., Rosinski, J.: Approximations of small jumps of Lévy processes with a view towards simulation. Journal of Applied Probability **38**, 482–493 (2001)
3. Baldi, P.: Exact Asymptotics for the Probability of Exit from a Domain and Applications to Simulation. Annals of Probability **23**, 1644–1670 (1995)
4. Beghin, L., Orsingher, E.: On the Maximum of the Generalized Brownian Bridge. Lithuanian Mathematical Journal **39**, 157–167 (1999)
5. Borodin, A.N., Salminen, P.: Handbook of Brownian Motion – Facts and Formulae, 2nd edition, Birkhäuser, Basel, Boston, Berlin (2002)
6. Cai, N., Chen, N., Wan, X.: Occupation Times of Jump-Diffusion Processes with Double Exponential Jumps and the Pricing of Options. Mathematics of Operations Research **35**(2), 412–437 (2010)
7. Campolieti, G., Makarov, R.N.: On Properties of Some Analytically Solvable Families of Local Volatility Models. Mathematical Finance, to appear (2011)
8. Campolieti, G., Makarov, R.N., Wouterloot, K.: Pricing Step Options under Solvable Nonlinear Diffusion Models. Working paper (2011)
9. Cont, R., Tankov, P.: Financial modelling with jump processes. Chapman & Hall/CRC (2004)
10. Cox J.: Notes on option pricing I: Constant elasticity of variance diffusions. Working paper, Stanford University (1975). Reprinted in Journal of Portfolio Management **22**, 15–17 (1996)
11. Dassios A.: Sample quantiles of stochastic processes with stationary and independent increments. The Annals of Applied Probability **6**(3), 1041–1043 (1996)
12. Davydov, D., Linetsky, V.: Pricing Options on Scalar Diffusions: An Eigenfunction Expansion Approach **51**, 185–209 (2003)
13. Hooghiemstra, G.: On explicit occupation time distributions for Brownian processes. Statistics & Probability Letters **56**, 405–417 (2002)
14. Hugonnier, J.-N.: The Feynman-Kac formula and pricing occupation time derivatives. International Journal of Theoretical and Applied Finance **2**(2), 153–178 (1999)
15. Glasserman, P.: Monte Carlo methods in financial engineering. Springer-Verlag, New York (2004)
16. Kou, S.G.: A jump-diffusion model for option pricing. Management Science **48**(8), 1086–1101 (2002)
17. Kou, S.G., Wang H.: Option pricing under a double exponential jump diffusion model. Management Science **50**, 1178–1192 (2004)

18. Leung, S.L., Kwok, Y.K.: Distribution of occupation times for constant elasticity of variance diffusion and pricing of the α-quantile options. Quantitative Finance **7**, 87–94 (2006)
19. Linetsky, V.: Step Options. Mathematical Finance **9**, 55–96 (1999)
20. Linetsky, V.: Lookback Options and Diffusion Hitting Times: A Spectral Expansion Approach. Finance and Stochastics **8**, 373–398 (2004)
21. Makarov, R.N., Glew, D.: Exact Simulations of Bessel Diffusions. Monte Carlo Methods and Applications **16**(3), 283–306 (2010)

A Global Adaptive Quasi-Monte Carlo Algorithm for Functions of Low Truncation Dimension Applied to Problems from Finance

Dirk Nuyens and Benjamin J. Waterhouse

Abstract We show how to improve the performance of the quasi-Monte Carlo method for solving some pricing problems from financial engineering. The key point of the new algorithm, coined "GELT", is an adaptive re-ordering of the point set so that the function is sampled more frequently in the regions where there is greater variation. The adaptivity only operates on the first few dimensions of the integrand and we show how to explicitly obtain the points of a digital sequence falling into boxes into these first few dimensions. This is effective as the problem is first transformed into having "low truncation dimension". In general it is assumed that finance problems have low effective dimension. In addition we make use of a so-called "sniffer function" to cope with the discontinuity in the integrand function. Numerical results with the new adaptive algorithm are presented for pricing a digital Asian option, an Asian option and an Asian option with an up-and-out barrier.

1 Introduction

We are interested in the pricing of contingent claims, which is of great interest in the area of mathematical finance. Typically, the claim is made on the uncertain future value of an asset such as a stock, commodity or exchange rate. The price of the asset S at time t is assumed to follow a stochastic differential equation

D. Nuyens
Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, 3001, Heverlee, Belgium
School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
e-mail: dirk.nuyens@cs.kuleuven.be

B.J. Waterhouse
School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
e-mail: ben.waterhouse@modelsolutions.net.au

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_34, © Springer-Verlag Berlin Heidelberg 2012

$$\mathrm{d}S(t) = a(S,t)\,\mathrm{d}t + b(S,t)\,\mathrm{d}W(t), \quad 0 < t \leq T,$$

with some initial price $S(0) = S_0$. We are interested in calculating the expected value of such a contingent claim. For example, the claim on a European call option with strike price K is $\max(S(T) - K, 0)$. In this case, the claim depends only on the final value of the asset, for other types the entire path may be important. Here we focus on "Asian" options where the claim is based upon the average price over time.

Most often the problem must be approximated numerically. The expected value problem may be formulated as an integration problem over the unit cube where numerical integration techniques such as Monte Carlo (MC) integration are often used. Monte Carlo integration refers to approximating the integral as

$$\int_{[0,1]^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \frac{1}{n}\sum_{k=0}^{n-1} f(\boldsymbol{x}_k), \tag{1}$$

where the points $\boldsymbol{x}_k \in [0,1]^s$ are chosen i.i.d. from the unit cube. The value of s is related to the number of time discretisations used in the problem. This can typically be in the hundreds or thousands.

Quasi-Monte Carlo (QMC) integration looks similar to MC integration except that the points $\boldsymbol{x}_k$ are chosen deterministically from the unit cube. In this paper we make use of a particular type known as digital (t,m,s)-nets and (t,s)-sequences where the number of points need not be fixed *a priori*. This is a desirable property as we will typically continue to add points until some error bound is achieved. These sequences are ordered in such a way that the unit cube is filled in a uniform way. However, many problems from mathematical finance involve integrands which are constant for large regions of the domain.

The key idea of this paper is to sample more frequently in regions which are "interesting" using a "sniffer function", where, crucially, the integrand needs to have low truncation dimension. Techniques for obtaining a reformulation of the original integral, known as "path construction methods", have been well studied in the literature and are still an active research topic. We will show the effect, in pictures, of a good transformation for our running example of a digital Asian option. Such a transformation is the first step in applying our adaptive algorithm.

The remainder of the paper is organised as follows. Problems typical of financial mathematics are discussed in Sect. 2. In Sect. 3 we examine the structure of digital (t,m,s)-nets and identify ways in which we can exploit their structure. The new adaptive algorithm is detailed in Sect. 4 and we present numerical results in Sect. 5.

2 Problems from Financial Mathematics

In this paper we consider contingent claims over stocks using the basic Black and Scholes model [1, 15]

$$\mathrm{d}S(t) = rS(t)\,\mathrm{d}t + \sigma S(t)\,\mathrm{d}W(t), \quad 0 < t \le T, \tag{2}$$

where r is the risk-free interest rate and σ is the volatility of the stock. One may allow r and σ to vary with time and the stock price, but for simplicity we shall consider them to be constant. Some may argue that the Black and Scholes model is a bad fit for reality, and they are right. However, here it simplifies matters and lets us focus on the story we want to tell. Nevertheless, the proposed method will also work in more advanced settings as long as one is able to transform the problem into one having low truncation dimension. More advanced path constructions than discussed in this paper can be used to that effect, see, e.g., [11].

It is well-known that the solution to (2), for a given Brownian motion $W(t)$, is

$$S(t) = S_0 \exp((r - \sigma^2/2)t + \sigma W(t)), \quad 0 < t \le T, \quad \text{where } S_0 = S(0). \tag{3}$$

2.1 Constructing an Asset Price Path

Our task now is to construct a discretised Brownian motion $W(t_j)$ for $j = 1, \dots, s$. To simplify the notation, we define the vector $\boldsymbol{w} = (W(t_1), \dots, W(t_s))$ containing the Brownian motion at times $t_1, \dots, t_s$. The vector $\boldsymbol{w}$ has mean zero and covariance matrix $\Sigma = \big(\min(t_i, t_j)\big)_{i,j=1}^{s}$. To construct the vector $\boldsymbol{w}$, we simply construct a vector $\boldsymbol{z} \sim N_s(\boldsymbol{0}, I)$ and note that $A\boldsymbol{z} \sim N_s(\boldsymbol{0}, \Sigma)$ if $AA^\top = \Sigma$. Given $\boldsymbol{w} = A\boldsymbol{z}$, the asset price at time t_j can be found using (3) to be

$$S(t_j) = S_0 \exp((r - \sigma^2/2)t_j + \sigma w_j).$$

There are several ways of constructing the Brownian path $\boldsymbol{w}$, or equivalently, of choosing the matrix A. Here we only make use of the three most straightforward methods: increment-by-increment (also called standard construction or Cholesky construction), Brownian bridge and PCA (a full eigenvector based decomposition). All of them are discussed in, e.g., [8] or [7]. Numerical results for a specific problem are shown in Fig. 2. We note that there are more advanced methods to construct the paths, see, e.g., [10, 11, 27]. A good path construction method is important in ensuring that the problem will be of low truncation dimension. However, from the point of view of the (plain) Monte Carlo method each of these construction methods is equivalent. It is only when using a quasi-Monte Carlo method that the choice of path construction can make a difference in the numerical approximation of the integral.

2.2 A Digital Asian Call Option

The typical pricing problem from financial engineering involves calculating the expected value of a contingent claim g whose value depends on the asset price.

For the discretised model we are considering, we assume that g depends on the asset price at times $t_1, t_2, \dots, t_s$. An example of this is a single-stock digital Asian call option with strike price K. The payoff of this claim is given by

$$g(S(t_1), \dots, S(t_s)) = 1_+ \left(\frac{1}{s} \sum_{j=1}^{s} S(t_k) - K \right), \tag{4}$$

where 1_+ represents the indicator function for the positive real line. That is, the value of the claim at time T is one if the arithmetic average of the price is greater than K and zero otherwise. The fact that the indicator function defines a jump in the payoff makes this problem harder than the typical Asian option that is often considered in QMC related literature.

The discounted value of this claim at time $t = 0$ is given by $e^{-rT}\,\mathbb{E}(g(\boldsymbol{w}))$. We calculate this expected value by integrating the value of the claim over all possible Brownian paths $\boldsymbol{w}$. This allows us to formulate the problem as

$$\begin{aligned} \mathbb{E}(g(\boldsymbol{w})) &= \int_{\mathbb{R}^s} g(\boldsymbol{w}) \, \frac{1}{(2\pi)^{s/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}\,\boldsymbol{w}^\top \Sigma^{-1} \boldsymbol{w}\right) \mathrm{d}\boldsymbol{w} \\ &= \int_{[0,1]^s} g(A\Phi^{-1}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} \end{aligned} \tag{5}$$

where $\Phi^{-1}(\boldsymbol{x}) = (\Phi^{-1}(x_1), \dots, \Phi^{-1}(x_s))^\top$ with $\Phi^{-1}(\cdot)$ denoting the inverse cumulative normal distribution function. It is important to notice here that evaluation of the inverse cumulative normal is by far the most expensive part in evaluating (4). We will make use of this observation in the new algorithm.

3 Quasi-Monte Carlo Point Sets

We approximate the s-dimensional integral over the unit cube with an n-point equal weight approximation of the form (1) using deterministically chosen points $\boldsymbol{x}_k$, for $k = 0, 1, \dots, n-1$ from a QMC sequence. It is well-known that the error of MC integration is $O(n^{-1/2})$, whereas for QMC integration the error is $O(n^{-1}(\log n)^s)$. Although the asymptotic bound is superior for QMC, for typically encountered values of n and s, the QMC bound is much weaker. Work by Sloan and Woźniakowski [23], and several following, demonstrated that if the importance of subsequent variables in the integrand diminishes sufficiently quickly then it is possible to achieve a rate of convergence arbitrarily close to $O(n^{-1})$.

Although most problems from finance are not covered by the theory from [23] this rate can often be achieved. See for example the numerical results in [4] which show $O(n^{-0.9})$ for an Asian option under the PCA path construction. For the more difficult digital Asian option considered here, we observe, in Fig. 2, a rate of approximately $O(n^{-0.7})$ using PCA path construction. In Sect. 5 we will see that the adaptive algorithm presented later in this paper will further improve upon this result.

3.1 Randomised QMC

One advantage of MC integration is that we may calculate an unbiased estimate of the standard error of the integration. Since the points in a QMC point set are correlated, we may no longer calculate this unbiased estimate. To get around this we randomise the QMC point set. For an overview of several randomization techniques we refer to [13]. Here we use *random shifts*, to be specified next, uniformly drawn from the s-dimensional unit cube. For each of these independent random shifts, $\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_M$, we then obtain a number of independent estimates for the integral, $Q_1, \ldots, Q_M$, where

$$Q_i := \frac{1}{n}\sum_{k=0}^{n-1} f(\boldsymbol{x}_k^{(i)}), \qquad \text{with } \boldsymbol{x}_k^{(i)} \text{ the shifted version of point } \boldsymbol{x}_k \text{ by shift } \boldsymbol{\Delta}_i.$$

The approximation to the integral is now taken as the average over the M independent n-point approximations

$$\overline{Q} := \frac{1}{M}\sum_{i=1}^{M} Q_i, \qquad \text{stderr}(\overline{Q}) = \sqrt{\frac{1}{M(M-1)}\sum_{i=1}^{M}\left(Q_i - \overline{Q}\right)^2}. \tag{6}$$

The total number of sample points used is then nM. The independent approximations, $Q_1, \ldots, Q_M$, can be used to calculate an unbiased estimate of the standard error for the approximation $\overline{Q}$ by the usual formula (6). Typically M is taken a small number, say 10 or 20, where more random shifts give a better approximation for the standard error. It is no use taking M much larger since one is only interested in the magnitude of the standard error.

The type of random shift considered in this paper is a *digital shift*, see, e.g., [5, 13].

Definition 1. Given an s-dimensional point set $P = \{\boldsymbol{x}_k\}_k$ and a shift $\boldsymbol{\Delta} \in [0,1)^s$, we define the *digitally shifted point set* $P + \boldsymbol{\Delta}$ in base b by setting

$$P + \boldsymbol{\Delta} = \{\boldsymbol{y}_k\}_k, \qquad \text{where } \boldsymbol{y}_k = \boldsymbol{x}_k \oplus_b \boldsymbol{\Delta},$$

where $\oplus_b$ is digitwise addition, in base b, modulo b, applied componentwise.

3.2 Digital Nets and Sequences

In this paper we are interested in a type of QMC point set known as a (t, m, s)-net in base b (where base 2 is the most practical choice). Such a (t, m, s)-net in base b could be a set of b^m points taken from a (t, s)-sequence in base b. We shall see that these sequences and nets have some attractive and exploitable properties. For a full background to the theory of these nets see [5, 16]. First we need to define what is meant by the notion of an elementary interval in base b.

Definition 2. An *elementary interval* in base b is a half-open subset of the unit cube of the form

$$J(\boldsymbol{a}, \boldsymbol{h}) = \prod_{j=1}^{s} \left[a_j\, b^{-h_j}, (a_j + 1)\, b^{-h_j}\right), \quad \text{for all } h_j \geq 0 \text{ and } 0 \leq a_j < b^{h_j}.$$

Such an elementary interval $J(\boldsymbol{a}, \boldsymbol{h})$ has volume $b^{-\sum_j h_j}$. If such an elementary interval has exactly the expected number of points for a given point set, then that point set is called a (t, m, s)-net.

Definition 3. A (t, m, s)*-net in base* b is an s-dimensional point set with b^m points and which has in each elementary interval of volume b^{t-m} exactly the expected number of points $b^{t-m} b^m = b^t$.

The parameter t, $0 \leq t \leq m$, is sometimes called (counterintuitively) the *quality parameter* of the net, where a smaller value of t is better. Obviously any point set in the unit cube is a (t, m, s)-net with $t = m$, since all the points fall in the unit cube, and one is then obviously interested in the smallest value of t possible.

One of the attractive features of, e.g., Sobol' points is that the opening dimensions are of particularly good quality (this is true for all popular low-discrepancy point sets). The Sobol' sequence yields $(0, m, 2)$-nets with m being any positive integer. (For a comprehensive list of minimal values of t, given m, s and b, see http://mint.sbg.ac.at/ and [22].) In other words, taking any initial sequence of the Sobol' sequence of size 2^m, then one will find one point in each of the elementary intervals of volume 2^{-m} in the first two dimensions, i.e., $t = 0$.

Another important property is that a (t, m, s)-net in base b will remain a (t, m, s)-net, with exactly the same t-value, following any digital shift in base b. This is easy to see from Definitions 1 and 3, since the digital shift is a bijection from $\mathbb{Z}_b$ to $\mathbb{Z}_b$ for each of the digits in the base b expansion of the point. Thus also for a shifted Sobol' sequence the observation from the previous paragraph is true.

A particular set of these elementary intervals *in the first two dimensions* of volume 2^{-m}, for m a multiple of 2, are the square boxes with sides $2^{-m/2}$. Observing the left-hand point set in Fig. 1, we see that 16 digitally-shifted Sobol' points, which form a $(0, 4, 2)$-net, can be divided into a 4×4 grid of equi-sized squares with one point lying in each box. In the right-hand point set of Fig. 1, we increased the number of points to $64 = 4 \times 16$. Now there are 4 points in each box of the 4×4

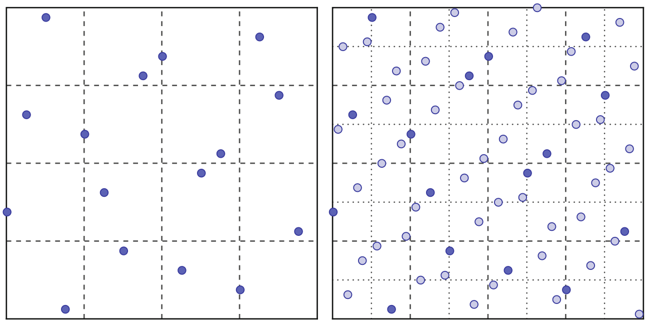


Fig. 1 First two dimensions of a digitally-shifted Sobol' sequence. *Left*: first 16 (= 2^4) points, *right*: first 64 (= 2^6) points.

grid and these grids can now be further subdivided to form an 8 × 8 grid. This will form the basis of the subdivision process in the adaptive algorithm.

Although this subdivision process seems trivial in two dimensions, the cost is exponential in the number of dimensions, i.e., 2^s. Furthermore, there are no $(0, m, 3)$-sequences in base 2 and one then has to resort to $(0, s)$-sequences in some higher base $b > 2$, e.g., the Faure sequence [6]. The subdivision in each dimension will then be b times instead of just 2, i.e., b^s subdivisions. The proposed algorithm in Sect. 4 can handle all that, but it will become less and less interesting as the base increases. Alternatively, for $t \neq 0$, one may choose to increase the number of points at a higher pace. From Definition 3 we note that as we take b^m points from a (t, s)-sequence in base b, and $t - m$ is a multiple of s, i.e., $t - m = -\nu s$, then we find b^t points in each elementary box with sides $b^{-\nu} \times \cdots \times b^{-\nu}$.

3.3 QMC on the Digital Asian Call Option

We use the following parameters for the digital option used throughout the paper:

$$T = 2, \qquad s = 256, \qquad \sigma = 23\%, \qquad r = 5\%, \qquad S_0 = 1, \qquad K = 1.$$

To obtain an error estimate we take the approach of Sect. 3.1 and therefore the Sobol' point set was digitally shifted ten times. In Fig. 2 we see the price of the option and the standard error as the number of points in the Sobol' point set is increased. All three different methods of factorising the matrix $\Sigma = AA^\top$ from Sect. 2.1 are shown as well as the plain Monte Carlo result. It is clear from this figure that the PCA method performs best for this particular problem, with the Brownian bridge method the next best, followed by the standard construction. In each case, the same Sobol' point set and the same digital shifts were used. All three QMC methods easily outperform the MC method.

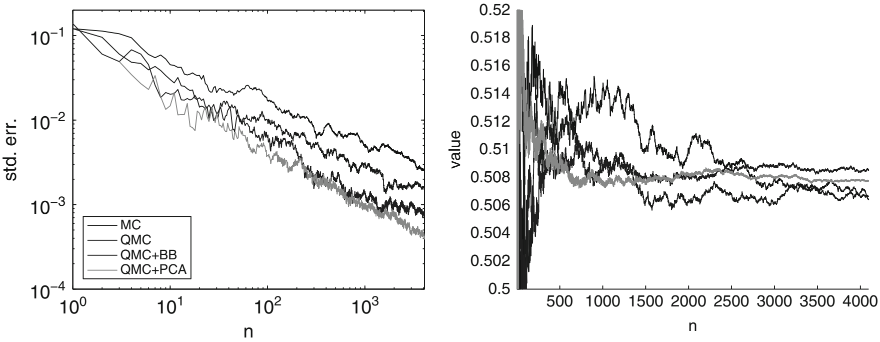


Fig. 2 Different path constructions on the digital Asian option problem. *Left*: the standard error (using ten shifts), the lines are in the same ordering as on the legend. One can determine that Monte Carlo (MC) performs like $O(n^{-0.5})$, quasi-Monte Carlo with the standard construction (QMC) has $O(n^{-0.55})$, whilst using Brownian bridge (QMC+BB) gives $O(n^{-0.64})$ and using (QMC+PCA) gives $O(n^{-0.71})$. *Right*: the convergence is illustrated by looking at the calculated value.

It should be noted that the PCA method will not be the best for every application. Work by Papageorgiou [19] and by Wang and Sloan [27] demonstrates that depending on the particular problem, the standard construction may even be the best choice for the matrix A. There also exist more advanced transformation methods, so-called linear transform (LT) methods, see, e.g., [10, 11], which will try to find the "optimal" linear transform for the problem at hand. While we will not discuss those methods further, it should be obvious that their usage could be advantageous, especially since they allow for more realistic pricing models than the standard lognormal model we employ here, cf. [11]. In the next section we give a heuristic explanation as to why particular problems perform better with a particular choice of the matrix A.

3.4 Low Truncation Dimension

One feature of QMC point sets is that the quality of the point set deteriorates as the dimension increases. That is, the minimal possible t-value of the (t, m, s)-net will increase as s is increased. We should therefore aim to construct our integration problem in such a way as to have *low truncation dimension*. We follow Caflisch *et al.*'s definition of truncation dimension [2].

For $f \in L^2([0,1]^s)$ there is an orthogonal ANOVA decomposition

$$f(\boldsymbol{x}) = \sum_{\mathfrak{u} \subseteq \mathscr{D}} f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}), \tag{7}$$

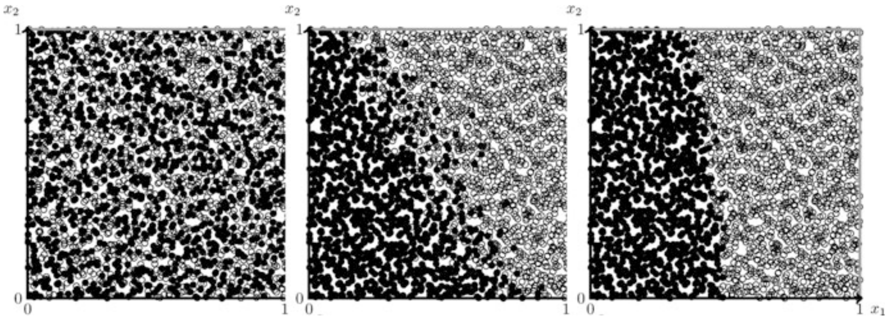


Fig. 3 Opening 2-dimensional projections (in $[0,1]^2$) of a digital Asian call option using three usual ways of path constructions for QMC. *Left*: Standard construction, *middle*: Brownian bridge construction, *right*: PCA construction.

with $\mathscr{D} = \{1, 2, \ldots, s\}$ and where the function $f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}})$ depends only on x_j if $j \in \mathfrak{u}$ and the variance of the function can be written as

$$\sigma^2(f) := I_s(f^2) - I_s(f)^2 = \sum_{\mathfrak{u} \subseteq \mathscr{D}} \sigma^2(f_{\mathfrak{u}}), \quad \sigma^2(f_{\mathfrak{u}}) = \int_{[0,1]^s} (f_{\mathfrak{u}}(\boldsymbol{x}_{\mathfrak{u}}))^2 \, \mathrm{d}\boldsymbol{x},$$

where $\sigma^2(f_\emptyset) = 0$. We next define the truncation dimension.

Definition 4. The *truncation dimension* of f is q if

$$\sum_{\mathfrak{u} \subseteq \{1,\ldots,q\}} \sigma_{\mathfrak{u}}^2 \geq p\,\sigma^2,$$

where p is an agreed upon constant chosen close to 1.

So, for $p = 0.99$ over 99% of the variance is captured by the first q variables. There is a similar concept known as the *superposition dimension* which will not be of interest to us in this paper. See [2] for an explanation of this.

If a problem has low truncation dimension, and if the opening few variables of the QMC point set are of superior quality to the subsequent dimensions, then we should expect (see [18]) to get a better result than if the problem did not have low truncation dimension. Furthermore, for an Asian option (non-digital) the truncation dimension after PCA is 2, after Brownian bridge it is 8 and for standard construction it is $0.8s$ [25, 26].

We now return to our example problem, the digital Asian option. While it is possible to estimate the truncation dimension, we gain more insight into the motivation for the algorithm by looking at some graphs. In Fig. 3 we plot the $(1,2)$-dimensional projection of a 256-dimensional point sampling of our digital Asian option example (using 256 Sobol' points with 10 random shifts). If the value of the integrand was 0 at a particular point, then that point is denoted with a full disc (darker areas). If the integrand took value 1, then it is denoted with an open disc (lighter areas).

We see in Fig. 3a, the standard construction, that, on the basis of the location of the first two dimensions of the 256-dimensional point, we have little idea as to whether the integrand will take the value 0 or 1. In Fig. 3b, which is the Brownian bridge construction, we gain a much clearer picture. On the basis of just the first two components of each point, we have a good idea as to whether or not the integrand takes value 0 or 1. However, there is a distinct "region of uncertainty". Note that even if for some simulated value of $(x_1, x_2, x_3, \ldots, x_s)$ the value of the function may be 1, then that does not mean that there cannot be another vector $(x_1, x_2, x'_3, \ldots, x'_s)$ where the value might be 0. We could loosely define the region of uncertainty in these 2-dimensional plots as the area where a change in the coordinates x_j for $j > 2$ could have a sudden change of the function value from 0 to 1 or *visa versa*. In this sense there is also a region of uncertainty in the 2-dimensional plot for the standard construction, but there it spans the whole area $[0, 1]^2$.

For the PCA construction in Fig. 3c we see that, based on just the first two components of each point, we can guess with very high probability the value that the integrand takes for the full 256-dimensional point. There is still a region of uncertainty which we shall refer to as the *interesting region*, however, it makes up just a very small proportion of the overall domain. We will describe an adaptive algorithm, where the adaptivity is only used in the first two dimensions, to avoid waisting samples on the constant part. Since we know QMC works well on the problem, the algorithm we propose is actually a reordering of the points such that they will sample more densely in the *interesting region*. Combined with a good stopping criterion this will lead to avoid sampling the constant area of the payoff.

4 A New Adaptive Algorithm for Low Truncation Dimension

A global adaptive algorithm breaks down the integration domain recursively in smaller subdomains, globally selecting subdomains for further refinement which have the largest estimated error contribution, see, e.g., [3]. Uniform subdivision of the integration domain has a cost which grows exponentially with the number of dimensions, e.g., one may choose to split an s-dimensional hypercube into 2^s smaller subcubes. Due to the exponential cost in the number of dimensions such an approach is only feasible for low dimensional functions (say $s \leq 5$). This means the 256 dimensions of our example problem are out of the question.

Alternative techniques for high-dimensional functions are based on the idea from HALF [24] and only split one dimension at a time "in half". In the Monte Carlo literature such an algorithm is known by the name MISER [20] where the splitting is based on the variance of the regions. VEGAS [14] is an alternative method that uses importance sampling based on an estimated probability density function which is constructed during an initial step of the algorithm. These algorithms have been adapted for usage with QMC in [21].

These approaches have difficulties when parts of the integrand are constant or when the integrand is really high-dimensional. E.g., the performance of MISER (using the GNU Scientific Library implementation) on our running example is identical to Monte Carlo; whilst VEGAS is unable to pass its initialization phase. Therefore we follow a completely different path by exploiting the low truncation dimension property of the function, allowing uniform subdivision in the first few dimensions, and by guiding the adaptive process by the introduction of a "sniffer function".

Assume the truncation dimension is denoted by q, and q is rather small (say 2 or 3). For the digital Asian option in fact we estimated $q = 2$, c.f. [25]. Furthermore we are given a (t, s)-sequence in base b for which, if we confine the sequence to only the first q dimensions, its t-value, denoted by t_q, is also rather small, say 0 or 1. For the Sobol' and Niederreiter sequences we have $t_2 = 0$. We now use the properties laid out in Sect. 3.2 and formalize what we need in the following proposition.

Proposition 1. *Given a (t, s)-sequence in base b, which, confined to the first q dimensions has a t-value of t_q, then the unit cube $[0, 1]^q$ can be subdivided into $b^{\nu q}$ congruent subcubes for which the first b^m points have exactly b^{t_q} points in each of these subcubes if $m - t_q = \nu q$ for $\nu \in \mathbb{N}$, i.e., if $m - t_q$ is a multiple of q.*

Proof. This is a direct consequence of Definitions 2 and 3.

Given such a sequence we propose Algorithm 1, which is a global adaptive algorithm which adaptively subdivides the first q dimensions. Next to MISER [20] and VEGAS [14] we propose to call this algorithm GELT, which are chocolate coins, as they are clearly less expensive than real gold coins. (Chocolate "gelt" is given to Jewish children for Chanukah, but a similar tradition exists in Belgium and The Netherlands for St. Nicholas, where "geld" is the Dutch word for money.)
Note that, in contrast with direct application of QMC, the sample points are now *not* equally weighted over the whole domain, but they are equally weighted with respect to the volume in which they occur, this as a direct consequence of splitting up the domain in smaller subdomains. Also note that in step 2(c)i we reuse the previous function values in the box.

An important consequence of the new algorithm is that, if we fix a preset maximum resolution R, it is still using exactly the same QMC points, but in a different ordering. This means that the trust one would have in applying QMC to the given problem is easily ported to the adaptive algorithm, since running it "till the end" of a $(t, Rq + t_q, s)$-net will give exactly the same numerical result (apart from rounding error); but in case of a low truncation dimension the algorithm will have converged dramatically faster as we will see.

Remark 1. Instead of the uniform subdivision scheme that we propose here, one could consider the approach from HALF [24]. This would allow the values of b, q and t_q to be larger, letting q grow to the normal values of dimensionality for HALF, VEGAS and MISER. (For $b \neq 2$ this algorithm would divide each dimension in b parts instead of 2.) E.g., in [21] numerical tests go up to 30 dimensions, although

Algorithm 1 GELT (Global adaptive reordering for low truncation dimension)

```
0. Input
   – an $s$-dimensional integrand function with (small) truncation dimension $q$,
   – a $(t, s)$-sequence in base $b$, with (small) $t$-value $t_q$ in the first $q$ dimensions.
1. Initialize
   – regionlist $= \big[$(box: $[0, 1]^q$, resolution: $0$, importance: $+\infty$)$\big]$
       (as a priority queue),
   – global approximation $Q = 0$,
   – error estimate $E = +\infty$.
2. Repeat the following as long as convergence criterion not met
   a. Remove the regions w $\in$ regionlist which have the largest importance.
   b. For all these regions w $\equiv$ (box: $B$, resolution: $\nu$, importance: $V$), split $B$ into
      $b^q$ congruent subcubes $B_i$ of $q$-dimensional volume $b^{-(\nu+1)q}$.
   c. Repeat the following for each such subcube $B_i$.
      i. Using the $b^{t_q}$ points in subcube $B_i$, from the first $b^{t_q+(\nu+1)q}$ points of the sequence,
         calculate:
         – the local approximation $Q_i^{(j)}$ for each of the random shifts, $j = 1, \ldots, M$,
             by evaluating only the new points;
         – and, based on the above calculations, the importance $V_i$ of this box $B_i$,
             by using the sniffer function (see Sect. 4.1).
      ii. Calculate the local approximation over all shifts: $Q_i = M^{-1} \sum_{j=1}^{M} Q_i^{(j)}$.
      iii. Add (box: $B_i$, resolution: $\nu + 1$, importance: $V_i$) to the list.
   d. Update the global approximation $Q$ and error estimate $E$.
3. Return global approximation $Q$ and error estimate $E$.
```

it is noted there that the QMC MISER algorithm proposed there should not be used in more than 5 dimensions, as plain QMC then seems to work better.

4.1 The Sniffer Function: Detecting Interesting Boxes

Foremost the adaptive QMC algorithm has to be able to detect interesting boxes. However, recall that our example function is discontinuous (typically in finance there is a discontinuity in the function itself or in the derivatives), as such the typical technique of estimating the variance in each box as a measure of "importance" is easily fooled. E.g., the discontinuity could be located close to the sides of the box where it is easily missed by sampling. To circumvent missing out on the discontinuity we propose the usage of a *sniffer function* to somehow smear out the information.

We now explain our main idea: since the dominating cost is the generation of multivariate normal samples and since evaluating the payoff function is negligible compared to this, cf. (4) and (5), it is relatively cheap to reuse the generated paths with a modified payoff function which we construct in such a way as to reveal the proximity of the discontinuity. In this way this so-called *"sniffer function"* will give an indication of the importance of a given box for further refinement, even in the case when all sample points in the box would have the same value.

Several approaches are possible for constructing such a sniffer function. The most straightforward approach takes a smoothed version of the original payoff function and then uses the variance of this sniffer function as an indicator to detect interesting regions. For this approach one can make direct use of the random shifting, calculating the variance per box. A more immediate approach uses the derivative of the smoothed payoff function as the sniffer function and then use its values directly as an indication of importance (and then we do not rely on random shifting).

Recall that the payoff function of our running example, the digital Asian call option, has an abrupt discontinuity, cf. (4). We can write this payoff in terms of the Heaviside step function H:

$$g(S(t_1), \dots, S(t_s)) = H(\overline{S} - K), \qquad \text{where } \overline{S} := \frac{1}{s} \sum_{j=1}^{s} S(t_k).$$

Using a well known smooth approximation for H and then differentiating we get

$$H_k(x) := \frac{1}{1 + \exp(-2kx)}, \quad \text{and} \quad D_k(x) := \frac{\mathrm{d}}{\mathrm{d}x} H_k(x) = \frac{2k \exp(-2kx)}{(1 + \exp(-2kx))^2},$$

where larger values of k give better approximations. The numerical example for the digital Asian option in Sect. 5 uses $D_k(x)/(2k)$ as the sniffer function with a value of $k = 20\nu$, where we scale k with the resolution ν of the box. As we also use random shifting we take the maximum value encountered and the sniffer value is then scaled w.r.t. the volume of the box. Calculating the value of the sniffer function is indeed an extra cost, but compared to constructing the path (and thus evaluating the inverse cumulative normal) this cost is negligible.

4.2 Localising Points and Shifted Points

The last important ingredient for the adaptive algorithm is the ability to generate the points inside a given box. To localise points of a digital (t, s)-sequence in base b which fall into the interesting boxes in the first q-dimensions we can use many approaches: (1) a brute force search through the generated points, (2) preprocess the sequence and form a hierarchical data set of indices, see, e.g., [12] for a similar approach w.r.t. the Halton sequence, or (3) directly solve a system of congruences governed by the generating matrices of the digital sequence to determine the point indices, see also [9] in this volume. Here we are interested in the last option as this is the most efficient one.

An s-dimensional digital sequence in base b is generated by a set of s infinite dimensional *generating matrices* $C_j \in \mathbb{F}_b^{\infty\times\infty}$, $j = 1, \ldots, s$. We now consider the points up to m-digit precision. If we denote by $C_j^{m\times m}$ the principal submatrix starting at the left upper corner of dimension $m \times m$, then the jth component of the kth point of the digital (t, m, s)-net taken from this (t, s)-sequence is generated by the following matrix-vector product over $\mathbb{F}_b$:

$$\vec{x}_{k,j} = \begin{pmatrix} x_{k,j,1} \\ x_{k,j,2} \\ \vdots \\ x_{k,j,m} \end{pmatrix} = C_j^{m\times m}\,\vec{k} = C_j^{m\times m} \begin{pmatrix} k_0 \\ k_1 \\ \vdots \\ k_{m-1} \end{pmatrix}, \tag{8}$$

where we use the base b expansions $x_{k,j} = \sum_{i=1}^{m} x_{k,j,i}\, b^{-i}$ and $k = \sum_{i=0}^{m-1} k_i\, b^i$, and the notation $\vec{x}$ means to assemble the base b digits in a vector over the finite field as indicated.

Now suppose we want to generate s-dimensional points in a q-dimensional subcube of resolution ν (in the first q dimensions), anchored at $\boldsymbol{a}/b^{\nu} \in [0, 1)^q$:

$$B(\boldsymbol{a}, \nu) := \prod_{j=1}^{q} [a_j\, b^{-\nu}, (a_j + 1)\, b^{-\nu}) \times \prod_{j=q+1}^{s} [0, 1),$$
$$\text{where } 0 \le a_j < b^{\nu},\ j = 1, \ldots, q,$$

then the base b digits of the anchor indices a_j determine which indices k will fulfill this condition. Following from (8) and Proposition 1, we get a system of congruences

$$\begin{pmatrix} \vec{a}_1 \\ \vdots \\ \vec{a}_q \end{pmatrix} = \begin{pmatrix} C_1^{\nu\times q\nu+t_q} \\ \vdots \\ C_q^{\nu\times q\nu+t_q} \end{pmatrix} \vec{k}. \tag{9}$$

The solutions $\vec{k} \in \mathbb{F}_b^{q\nu+t_q}$ determine which indices k fall inside $B(\boldsymbol{a}, \nu)$.

For brevity we will now focus on two-dimensional localisation, i.e., $q = 2$, using the Sobol' sequence, i.e., $t_2 = 0$. (Exactly the same holds for the Niederreiter sequence as the generating matrices of the first two dimensions are the same.) Then, the generating matrix C_1 for the first dimension is just the identity matrix I_{∞}, resulting in a *radical inversion* in base 2, and the generating matrix C_2 is upper triangular. To find the one point in the first $2^{2\nu}$ points in a box $B(\boldsymbol{a}, \nu)$ we first

Table 1 The matrices B_ν^{-1} for solving (10) for dimension 2 of the Sobol' or Niederreiter sequence in base 2: each column is interpreted as the binary expansion of an integer with the least significant bits in the top rows; for reference the generating matrix is also given.

ν	Columns of B_ν^{-1}
1	1
2	1, 3
3	2, 6, 5
4	1, 3, 5, 15
5	8, 24, 27, 30, 17
6	4, 12, 20, 60, 17, 51
7	2, 6, 10, 30, 34, 102, 85
8	1, 3, 5, 15, 17, 51, 85, 255
9	128, 384, 387, 390, 393, 408, 427, 510, 257
10	64, 192, 320, 960, 325, 975, 340, 1020, 257, 771
11	32, 96, 160, 480, 544, 1632, 1455, 510, 514, 1542, 1285
12	16, 48, 80, 240, 272, 816, 1360, 4080, 257, 771, 1285, 3855
13	8, 24, 40, 120, 136, 408, 680, 2040, 2056, 6168, 6939, 7710, 4369
14	4, 12, 20, 60, 68, 204, 340, 1020, 1028, 3084, 5140, 15420, 4369, 13107
15	2, 6, 10, 30, 34, 102, 170, 510, 514, 1542, 2570, 7710, 8738, 26214, 21845
16	1, 3, 5, 15, 17, 51, 85, 255, 257, 771, 1285, 3855, 4369, 13107, 21845, 65535

$$\begin{aligned} C_2^{\infty\times 32} = (&1, 3, 5, 15, 17, 51, 85, 255, 257, 771, 1285, 3855, 4369, 13107,\\ &21845, 65535, 65537, 196611, 327685, 983055, 1114129, 3342387,\\ &5570645, 16711935, 16843009, 50529027, 84215045, 252645135,\\ &286331153, 858993459, 1431655765, 4294967295) \end{aligned}$$

trivially solve $\vec{a}_1 = I_\nu \vec{k}_1$ with $\vec{k}_1 \in \mathbb{Z}_2^\nu$. Each point which has an index $k \equiv k_1 \pmod{2^\nu}$ will fall into $[a_1 b^{-\nu}, (a_1 + 1) b^{-\nu})$ into the first dimension. To determine k_2 we next solve

$$\vec{a}_2 = C_2^{\nu\times 2\nu} \vec{k} = \begin{pmatrix} A_\nu & B_\nu \end{pmatrix} \begin{pmatrix} \vec{k}_1 \\ \vec{k}_2 \end{pmatrix},$$

or in other words, solve

$$\vec{a}_2 - A_\nu \vec{k}_1 = B_\nu \vec{k}_2. \tag{10}$$

Since we have a $(0, 2)$-sequence we know we can find exactly one point and thus B_ν is invertible. We can calculate the matrices B_ν^{-1} for $\nu = 1, 2, \ldots$ up front, just as one stores the generating matrices of the Sobol' sequence (e.g., as a list of integers). These numbers are given in Table 1. Solving for $\vec{k}_2$ costs no more than generating a two-dimensional point of the sequence (using bit instructions) and is thus negligible if one is generating a 256-dimensional point set. Digital shifts are easily handled in this method by subtracting (modulo b) the first ν digits of the shift from the particular anchor. The same inverses B_ν^{-1} can be used for all possible shifts.

5 Numerical Results

We test three different call options: a digital Asian option with parameters as given in Sect. 3.3, a standard Asian option with parameters as in [4] ($T = 1$, $s = 100$, $\sigma = 0.2$, $r = 0.1$, $S_0 = 100$, $K = 100$), and this same Asian option with an up-and-out barrier condition added (same parameters as before with a barrier at $B = 110$). For all tests we use PCA path construction for the QMC and GELT algorithms and report the standard error over ten random shifts. In Fig. 4 we show for each test a row of three columns. In the first column we show the projection (for one shift only) on $[0, 1]^2$ denoting the distribution of the zero payoffs (in the top small panels) and the positive payoffs (bottom small panels) as sampled by the sniffer function. In the middle column we show confidence intervals at a distance of 3.4 from the mean. The graph shows the whole range of n for MC and QMC+PCA. The new algorithm QMC+PCA+GELT does not need this many samples and so stops earlier. In the last column the convergence of the standard error is plotted as well as two reference lines: one with slope $-1/2$ and one with slope -1.

The sniffer functions for these different products can be constructed intuitively (although interesting on its own, no real effort was made to find out optimal sniffers). For the digital Asian call option we use the following sniffer function

$$s_1(x,k) = \frac{\exp(-2kx)}{(1+\exp(-2kx))^2}, \qquad x = \frac{\overline{S}-K}{K}, \qquad k = 20\nu,$$

where ν is the resolution of the box which has sides of length 2^ν. As this is a binary option it is sufficient to detect the discontinuity. In the top row of Fig. 4 we see a convergence of $O(n^{-0.85})$ for QMC+PCA+GELT, c.f. Fig. 2.

For the Asian call option we use the following sniffer function

$$s_2(x,k) = \frac{1}{1+\exp(-kx)}, \qquad x = \frac{\overline{S}-K}{K} + \frac{1}{\nu^2}, \qquad k = 20\nu^2.$$

As this sniffer function just gives constant importance to areas which have an assumed positive payoff (in contrast with the sniffer above for the digital Asian) we artificially move the boundary by a factor ν^{-2}. In Fig. 4 we notice the same convergence speed as for the QMC+PCA algorithm, but without the shaky convergence behavior. Furthermore, QMC+PCA+GELT practically always has a smaller standard error than QMC+PCA and whilst QMC+PCA underestimates the value of the payoff in a shaky way, the new algorithm much quicker reaches a stable value.

The Asian call option with up-and-out barrier does not really have the low truncation dimension that we would like. We first propose a sniffer function for the up-and-out barrier part:

$$s_3(x,k) = 1 - \frac{1}{1+\exp(-kx)}, \qquad x = \frac{s\overline{S}-0.95B}{B}, \qquad k = \nu.$$

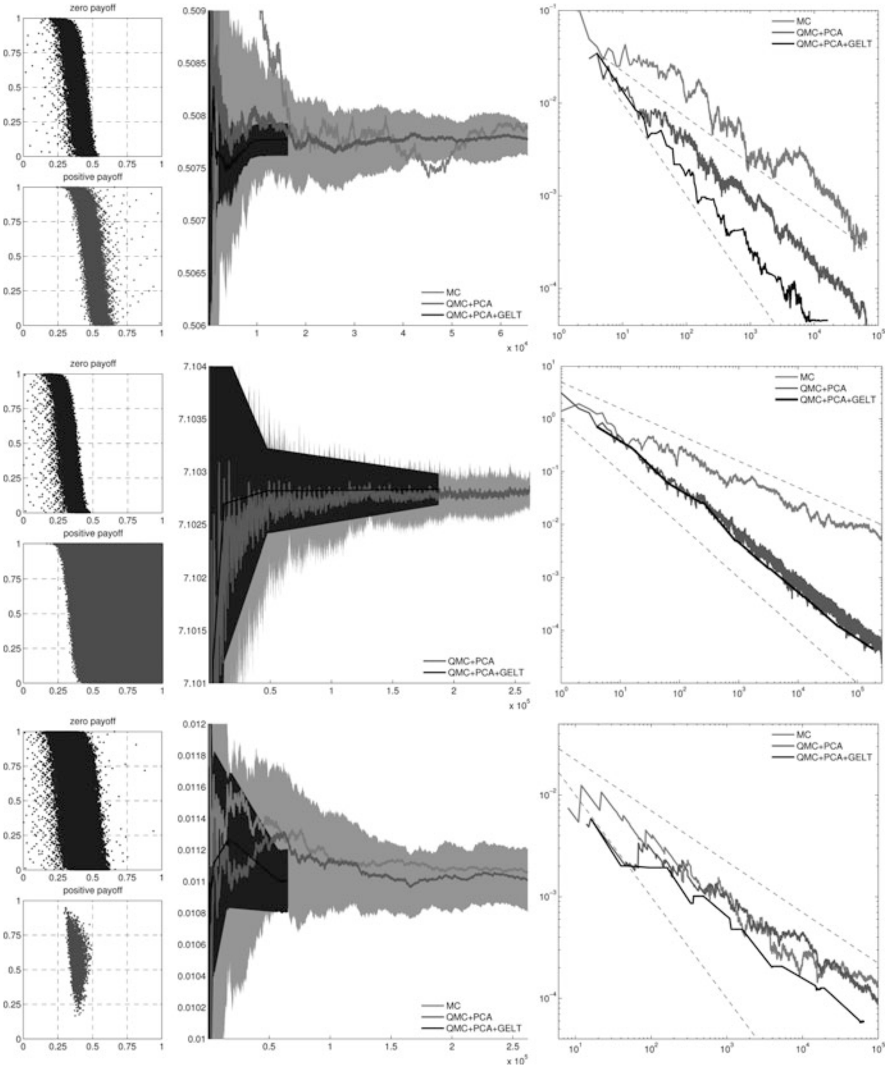


Fig. 4 *Top to bottom*: digital Asian, Asian and Asian with up-and-out barrier. *Left to right*: sniffer sampling (zero payoff/positive payoff); value and confidence intervals; standard error (ten shifts).

We use the same smoothing as for the Asian call which here only partially matches the shape that can be seen in the bottom row of Fig. 4, however we also see that this is sufficient to improve the convergence. This sniffer somehow distributes the barrier over the whole path where the 0.95 part artificially makes the area bigger.

For the complete sniffer for the Asian call option with up-and-out barrier we combine the two previous sniffers in such a way that if one of the two previous sniffers sees a box as unimportant the total sniffer will see it as unimportant:

$$s_4(x) = \left|(1 + s_2(x, 10\nu^2))(1 + s_3(x, \nu)) - 2\right| .$$

In Fig. 4 we clearly see that we need only a quarter of the samples to achieve a similar standard error as with the QMC+PCA or MC method.

Our calculated reference values for these examples are 0.50777 for the digital Asian call option, 7.1028 for the Asian call option and 0.011 for the Asian call with up-and-out barrier.

6 Conclusion

We presented GELT: a global adaptive algorithm based on reordering the points of a QMC sequence by means of a sniffer function for functions of low truncation dimension. The algorithm was demonstrated using examples from financial engineering in the Black & Scholes model for ease of presentation, but can as well be used in more advanced settings. We have shown that the algorithm performs better than the standard QMC+PCA approach while only having a minor overhead cost.

Acknowledgements The authors are grateful to Prof. Ian H. Sloan for useful discussions related to this paper and very much appreciated the careful comments and questions from the two anonymous referees. The first author is a fellow of the Research Foundation Flanders (FWO) and is grateful to the University of New South Wales where large parts of this paper were written; and therefore also thanks the Australian Research Council (ARC) for support.

References

1. F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
2. R. E. Caflisch, W. Morokoff, and A. B. Owen. Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *J. Comput. Finance*, 1(1):27–46, 1997.
3. R. Cools and A. Haegemans. Algorithm 824: CUBPACK: A package for automatic cubature; framework description. *ACM Trans. Math. Software*, 29(3):287–296, 2003.
4. R. Cools, F. Y. Kuo, and D. Nuyens. Constructing embedded lattice rules for multivariate integration. *SIAM J. Sci. Comput.*, 28(6):2162–2188, 2006.
5. J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, 2010.
6. H. Faure. Discrépance de suites associées à un système de numération (en dimension s). *Acta Arith.*, 41(4):337–351, 1982.
7. M. B. Giles, F. Y. Kuo, I. H. Sloan, and B. J. Waterhouse. Quasi-Monte Carlo for finance applications. *ANZIAM Journal*, 50:308–323, 2008.
8. P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, 2003.
9. L. Grünschloß, M. Raab, and A. Keller. Enumerating quasi-Monte Carlo point sequences in elementary intervals. In H. Woźniakowski and L. Plaskota, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*. Springer-Verlag, 2012. ibid.

10. J. Imai and K. S. Tan. A general dimension reduction technique for derivative pricing. *J. Comput. Finance*, 10(2):129–155, 2006.
11. J. Imai and K. S. Tan. An accelerating quasi-Monte Carlo method for option pricing under the generalized hyperbolic Lévy process. *SIAM J. Sci. Comput.*, 31(3):2282–2302, 2009.
12. A. Keller. Myths of computer graphics. In Niederreiter and Talay [17], pages 217–243.
13. P. L'Écuyer and C. Lemieux. Recent advances in randomized quasi-Monte Carlo methods. In M. Dror, P. L'Écuyer, and F. Szidarovszki, editors, *Modeling Uncertainty: An Examination of Its Theory, Methods, and Applications*, pages 419–474. Kluwer Academic, 2002.
14. P. G. Lepage. A new algorithm for adaptive multidimensional integration. *J. Comput. Phys.*, 27(2):192–203, 1978.
15. R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
16. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Number 63 in Regional Conference Series in Applied Mathematics. SIAM, 1992.
17. H. Niederreiter and D. Talay, editors. *Monte Carlo and Quasi-Monte Carlo Methods 2004*. Springer-Verlag, 2006.
18. A. B. Owen. Necessity of low effective dimension. Technical report, Dept. of Statistics, Stanford University, 2002.
19. A. Papageorgiou. The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration. *J. Complexity*, 18(1):171–186, 2002.
20. W. H. Press and G. R. Farrar. Recursive stratified sampling for multidimensional Monte Carlo integration. *Computers in Physics*, 4(2):190–195, 1990.
21. R. Schürer. Adaptive quasi-Monte Carlo integration based on MISER and VEGAS. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 393–406. Springer-Verlag, 2004.
22. R. Schürer and W. C. Schmid. MINT: A database for optimal net parameters. In Niederreiter and Talay [17], pages 457–469.
23. I. H. Sloan and H. Woźniakowski. When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *J. Complexity*, 14(1):1–33, 1998.
24. P. van Dooren and L. de Ridder. An adaptive algorithm for numerical integration over an n-dimensional cube. *J. Comput. Appl. Math.*, 2(3):207–217, 1976.
25. X. Wang and I. H. Sloan. Why are high-dimensional finance problems often of low effective dimension? *SIAM J. Sci. Comput.*, 27(1):159–183, 2005.
26. X. Wang and I. H. Sloan. Efficient weighted lattice rules with applications to finance. *SIAM J. Sci. Comput.*, 28(2):728–750, 2006.
27. X. Wang and I. H. Sloan. Quasi-Monte Carlo methods in financial engineering: An equivalence principle and dimension reduction. *Operations Res.*, 59(1):80–95, 2011.

Random and Deterministic Digit Permutations of the Halton Sequence*

Giray Ökten, Manan Shah, and Yevgeny Goncharov

Abstract The Halton sequence is one of the classical low-discrepancy sequences. It is effectively used in numerical integration when the dimension is small, however, for larger dimensions, the uniformity of the sequence quickly degrades. As a remedy, generalized (scrambled) Halton sequences have been introduced by several researchers since the 1970s. In a generalized Halton sequence, the digits of the original Halton sequence are permuted using a carefully selected permutation. Some of the permutations in the literature are designed to minimize some measure of discrepancy, and some are obtained heuristically.

In this paper, we investigate how these carefully selected permutations differ from a permutation simply generated at random. We use a recent genetic algorithm, test problems from numerical integration, and a recent randomized quasi-Monte Carlo method, to compare generalized Halton sequences with randomly chosen permutations, with the traditional generalized Halton sequences. Numerical results suggest that the random permutation approach is as good as, or better than, the "best" deterministic permutations.

Introduction

The Halton sequences are arguably the best known low-discrepancy sequences. They are obtained from one-dimensional van der Corput sequences which have a

* This material is based upon work supported by the National Science Foundation under Grant No. DMS 0703849.

G. Ökten · M. Shah · Y. Goncharov
Department of Mathematics, Florida State University, Tallahassee, FL, 32306-4510, USA
e-mail: okten@math.fsu.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_35, © Springer-Verlag Berlin Heidelberg 2012

simple definition easy to implement. The nth term of the van der Corput sequence in base b, denoted by $\phi_b(n)$, is defined as follows: First, write n in its base b expansion:

$$n = (a_k \cdots a_1 a_0)_b = a_0 + a_1 b + ... + a_k b^k,$$

then compute

$$\phi_b(n) = (0.a_0 a_1 \cdots a_k)_b = \frac{a_0}{b} + \frac{a_1}{b^2} + ... + \frac{a_k}{b^{k+1}}. \tag{1}$$

The Halton sequence in the bases $b_1, ..., b_s$ is $(\phi_{b_1}(n), ..., \phi_{b_s}(n))_{n=1}^{\infty}$. This is a uniformly distributed mod 1 (u.d. mod 1) sequence (see Kuipers and Niederreiter [11] for its definition) if the bases are relatively prime. In practice, b_i is usually chosen as the ith prime number.

One useful application of the Halton sequences (in general, low-discrepancy sequences) is to numerical integration. The celebrated Koksma–Hlawka inequality states,

Theorem 1. *If f has bounded variation $V(f)$ in the sense of Hardy and Krause over* $[0,1]^s$*, then, for any $x_1, ..., x_N \in [0,1)^s$, we have*

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(x_n) - \int_{[0,1)^s} f(x)dx \right| \leq V(f) D_N^*(x_i). \tag{2}$$

For the definition of bounded variation in the sense of Hardy and Krause, see Niederreiter [10]. The term $D_N^*(x_i)$, called the star discrepancy of vectors $x_1, ..., x_N$ in $[0,1)^s$, is defined as follows: For a subset S of $[0,1)^s$, let $A_N(S)$ be the number of vectors x_i that belong to S, and let $\lambda(S)$ be the s-dimensional Lebesgue measure of S.

Definition 1. The star discrepancy of vectors $x_1, ..., x_N \in [0,1)^s$ is

$$D_N^*(x_i) = \sup_S \left| \frac{A_N(S)}{N} - \lambda(S) \right|$$

where S is an s-dimensional interval of the form $\prod_{i=1}^{s} [0, \alpha_i)$, and the supremum is taken over the family of all such intervals. If the supremum is taken over intervals of the form $\prod_{i=1}^{s} [\alpha_i, \beta_i)$, then we obtain the so-called (extreme) discrepancy.

The star discrepancy of the Halton sequence, or any low-discrepancy sequence, is $O(N^{-1}(\log^s N))$. This fact, together with the Koksma–Hlawka inequality, lay the foundation of the quasi-Monte Carlo integration.

There is a well-known defect of the Halton sequence: in higher dimensions when the base is larger, certain components of the sequence exhibit very poor uniformity. This phenomenon is sometimes described as *high correlation between higher bases*. Figure 1, which plots the first 500 Halton vectors in bases 227 and

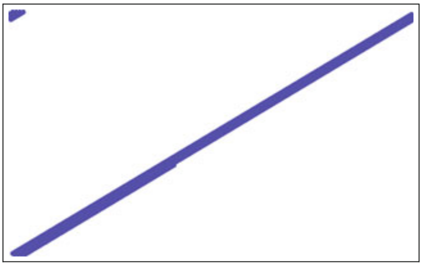


Fig. 1 The first 500 Halton vectors in bases 227 and 229.

229 (corresponding to 49th and 50th prime numbers) illustrate this high correlation. Similar plots have been reported by several authors in the past.

Observing this deficiency of the Halton sequence, Braaten and Weller [2] offered a remedy by generalizing the Halton sequence by using appropriately chosen permutations to scramble the digits in Eq. 1. Let σ_{b_i} be a permutation on the digit set $\{0, ..., b_i - 1\}$, and generalize Eq. 1 as

$$\phi_{b_i}(n) = \frac{\sigma_{b_i}(a_0)}{b_i} + \frac{\sigma_{b_i}(a_1)}{b_i^2} + ... + \frac{\sigma_{b_i}(a_k)}{b_i^{k+1}} \tag{3}$$

and define the Halton sequence in bases $b_1, ..., b_s$ as $(\phi_{b_1}(n), ..., \phi_{b_s}(n))_{n=1}^{\infty}$. Halton sequences generalized in this way are called generalized Halton, or scrambled Halton sequences. Here we will use the term *digit permuted Halton sequences*. Another generalization that allows different permutations for the different digits in Eq. 3 is also discussed in the literature; see, for example, Faure and Lemieux [6].

Since the publication of Braaten and Weller [2], several authors introduced different permutations to scramble the digits of the Halton sequence; see, for example, [1, 3–6, 8, 17–19]. Some of these permutations were obtained using heuristics, such as [8] and [18], and some others were obtained by searching for the optimal permutations that minimize the discrepancy of the one-dimensional or two-dimensional projections of the Halton sequence, such as [2–6].

As we will elaborate further in Sect. 1, most authors cited above use a numerical approach to compare various digit permuted Halton sequences and we will follow the same methodology. Before we get into more details, let us entertain a simple question: Do these digit permuted Halton sequences avoid the phenomenon of *high correlation between higher bases* (see Fig. 1), which was a defect of the Halton sequence? To answer this, we pick four permuted sequences; (1) permutations by Chi et al. [4], which were obtained by searching for best linear permutations that minimize correlations, (2) permutations by Faure [5], which were obtained by minimizing the discrepancy of one-dimensional projections, (3) permutations by Faure and Lemieux [6], which were obtained by considering both one and two-dimensional projections, and (4) permutations by Kocis and Whiten [8], which were obtained heuristically. Figure 2 plots the first 500 digit permuted Halton vectors in bases 71 & 229 using the Chi, Mascagni, Warnock (CMW) permutation, 191

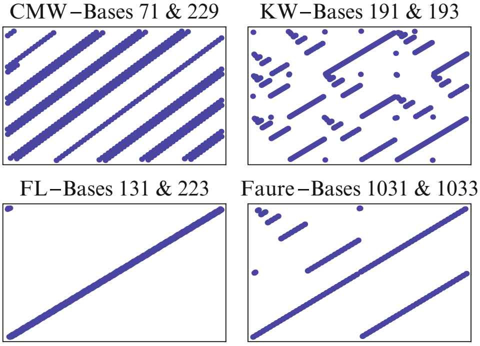


Fig. 2 The first 500 vectors from digit permuted Halton sequences.

& 193 using the Kocis and Whiten (KW) permutation, 131 & 223 using the Faure and Lemieux (FL) permutation, and 1,031 & 1,033 using the Faure permutation. Note that 1,033 is the 174th prime number and a dimension as large as 174 is not uncommon, for example, in financial applications.

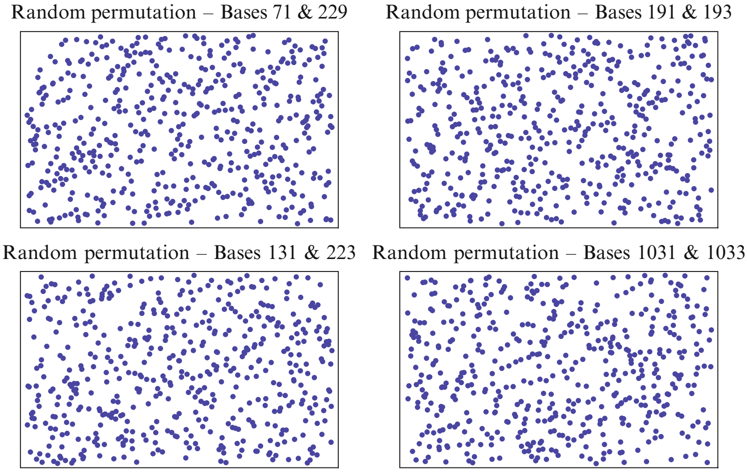
Figure 2 suggests that the digit permuted Halton sequences are also prone to the same deficiency of the Halton sequence. The bases used in the above plots were obtained by a computer search, and there are several other projections for each case that have similarly poor behavior. In Sects. 2 and 3, we will go further than a visual inspection, and compare digit permuted Halton sequences by their star discrepancy, and the error they produce in numerical integration.

In this paper we want to investigate the following question: What if we pick the permutation σ_{b_i} in Eq. 3, simply at random, from the space of all permutations? How would this approach, which we call *random digit permuted Halton sequence*, compare with the existing deterministic digit permuted Halton sequences? Perhaps a quick test for this idea would be to plot its vectors that correspond to the same bases we considered in Fig. 2.

Inspecting Fig. 3, we do not see a visual correlation we can speak of. Moreover, the same computer search program that we used to detect correlations in digit permuted Halton sequences did not detect similar correlations for any bases for the random digit permuted Halton sequence. On the other hand, one might wonder if these plots are too "pseudorandom like". The rest of the paper is devoted to comparing random digit permuted sequences with their deterministic counterparts.

1 Methodology

There are two main approaches to decide whether a given low-discrepancy sequence is better than another: theoretical, and empirical. The conventional theoretical approach computes upper bounds for the star discrepancy of the sequences, and chooses the one with the smaller upper bound. The star discrepancy of N vectors of an $s-$dimensional low-discrepancy sequence is bounded by $c_s(\log N)^s + O$



Random permutation – Bases 71 & 229

Random permutation – Bases 191 & 193

Random permutation – Bases 131 & 223

Random permutation – Bases 1031 & 1033

Fig. 3 First 500 vectors from randomly permuted Halton sequences.

$((\log N)^{s-1})$ where c_s is a constant that depends on the dimension s. The theoretical approach compares different sequences by their corresponding c_s values. There are two disadvantages of this approach. The first disadvantage is that since the upper bound for the star discrepancy becomes very large as s and N get larger, comparing the star discrepancy of different sequences by comparing the upper bounds they satisfy becomes meaningless when these upper bounds are several orders of magnitude larger than the actual star discrepancy.

The second disadvantage is that we do not know how tight the known bounds are for the constant c_s. For example, the Halton sequence used to be considered as the worst sequence among Faure, Sobol', Niederreiter, and Niederreiter-Xing sequences, based on the behavior of its c_s value. However, recent error bounds of Atanassov [1] imply significantly lower c_s values for the Halton sequence. In fact, a special case of these upper bounds, which apply to a digit permuted Halton sequence introduced by Atanassov [1], has lower c_s values than the Faure, Sobol', Niederreiter, and Niederreiter-Xing sequences. For details see Faure and Lemieux [6].

There are two empirical approaches used in the literature to compare low-discrepancy sequences. The first one is to apply the sequences to test problems with known solutions, and compare the sequences by the exact error they produce. The test problems are usually chosen from numerical integration, as well as various applications such as particle transport theory and computational finance. Numerical results are sometimes surprising. For example, even though the digit permuted Halton sequence by Atanassov [1] has the best known bounds for its star discrepancy, after extensive numerical results, Faure and Lemieux [6] conclude that several other digit permuted sequences (Chi et al. [4], Kocis and Whiten [8]) generally perform as well as the one by Atanassov [1] and Faure and Lemieux [6].

The second empirical approach is to compute the discrepancy of the sequence numerically. The star discrepancy is difficult to compute, but a variant of it, the L_2- discrepancy, is somewhat easier. In some papers, the L_2- discrepancy is used to compare different sequences. We will discuss a drawback of this approach in the next section.

In this paper, we will use the empirical approach to compare various digit permuted Halton sequences including the random digit permutation approach. Since it is not very practical to compare *all* digit permuted sequences, we will proceed as follows: Faure and Lemieux [6], after extensive numerical results, recommend the permutations by Atanassov and, Faure and Lemieux, and also report that permutations by Kocis and Whiten and Chi et al. generally perform well. We will use these sequences except the one by Atanassov in our numerical results. We will also consider the permutation by Faure [5], which was used successfully in previous numerical studies of the authors (Goncharov et al. [7]), and the permutation by Braaten and Weller [2]. The standard Halton sequence, and the permutation by Vandewoestyne and Cools [18], will be included in the numerical results as benchmarks.

Our empirical approach has two parts. We will compare the selected digit permuted sequences by computing lower and upper bounds for their star discrepancy, for some relatively small choices for sample size N, using a recent genetic algorithm developed by Shah [14] and an algorithm developed by Thiémard [15]. For larger sample sizes, however, we observed that computing meaningful bounds for the star discrepancy becomes intractable, and thus we will compare the sequences by the statistical error (possible by a randomization of the sequences we will discuss later) they produce when used in numerical integration. In our numerical results we do not consider the efficiency of the sequences. If we assume that the permutations are known and precomputed, as it would be the case in a practical implementation, then there is no significant difference between the computing times of various digit permuted Halton sequences.

The test problem we will consider from numerical integration is estimating the integral of

$$f(x_1, ..., x_s) = \prod_{i=1}^{s} \frac{|4x_i - 2| + a_i}{1 + a_i} \tag{4}$$

in $[0, 1)^s$. The exact value of the integral is one, and the sensitivity of the function to x_i quickly decreases as a_i increases. This function was first considered by Radovic et al. [21] and used subsequently by several authors.

2 Computing the Discrepancy

A modified version of the star discrepancy, which is easier to compute, is the L_2-star discrepancy:

Table 1 T_N^* and lower bounds for D_N^* of 16-dimensional digit permuted Halton vectors.

N	T_N^*		Lower Bounds for D_N^*	
	BW	REV	BW	REV
50	13.5×10^{-4}	2.00×10^{-4}	0.295	0.404
100	7.01×10^{-4}	1.77×10^{-4}	0.261	0.356
200	3.64×10^{-4}	1.53×10^{-4}	0.152	0.268

Definition 2. The L_2-star discrepancy of vectors $x_1, ..., x_N \in [0,1)^s$ is

$$T_N^*(x_i) = \left[\int_{[0,1)^s} \left(\frac{A_N(S)}{N} - \lambda(S) \right)^2 d\alpha_1 ... d\alpha_s \right]^{1/2}$$

where $S = \prod_{i=1}^{s} [0, \alpha_i)$.

Similarly, we can define the L_2-extreme discrepancy, $T_N(x_i)$, by replacing the sup norm in the definition of extreme discrepancy (Definition 1) by the L_2−norm. There are explicit formulas to compute T_N^* and T_N of a finite set of vectors. However, the formulas are ill-conditioned and they require high precision; see Vandewoestyne and Cools [18] for a discussion.

Matoušek [9] (p. 529) points out to a more serious drawback of T_N^*: if the dimension s is high, and the number of points is relatively small, then any point set clustered around the vertex $(1, 1, ..., 1)$ of the s-dimensional cube has nearly the best possible L_2-discrepancy!

We now discuss a recent example where the L_2-discrepancies give misleading results. In Vandewoestyne and Cools [18], a new permutation for the Halton sequence, called the reverse permutation, was introduced. The authors compared several digit permuted Halton sequences by their T_N^* and T_N, in dimensions that varied between 8 and 32. They considered at most $N = 1{,}000$ vectors in their computations. They concluded that the reverse permutation performed as good, or better, than the other permutations, in terms of the L_2-discrepancies. For example, Fig. 9 on page 355 of [18] shows that T_N^* of the sixteen dimensional Halton vectors obtained by the reverse permutation is much lower than that of the Braaten and Weller permutation, as N varies between 1 and 1,000. We compute T_N^*, and lower bounds for D_N^*, for the Braaten and Weller permutation (BW) and the reverse permutation (REV), when $N = 50, 100, 200$, in Table 1. The lower bounds for D_N^* are computed using the genetic algorithm by Shah [14], which we will discuss in more detail later.[2]

[2] In the numerical results of Sect. 2.1 we will give interval estimates for star discrepancy; a lower bound using the genetic algorithm, and an upper bound using Thiémard's algorithm. In this table, we only report lower bounds since computing upper bounds with these parameters was expensive.

Table 2 Integration error for f.

N	REV	BW	REV/BW
100	434×10^{-5}	34.1×10^{-5}	12.7
200	138×10^{-5}	13.7×10^{-5}	10.0
300	47.4×10^{-5}	44.2×10^{-5}	1.1
400	113×10^{-5}	7.28×10^{-5}	15.5
500	18.2×10^{-5}	18.0×10^{-5}	1.0
600	17.2×10^{-5}	38.8×10^{-5}	0.4
700	66.5×10^{-5}	9.84×10^{-5}	6.8
800	37.2×10^{-5}	11.4×10^{-5}	3.3
900	8.93×10^{-5}	8.89×10^{-5}	1.0
1,000	25.8×10^{-5}	11.8×10^{-5}	2.2

Observe that although T_N^* values for the reverse permutation are lower than the Braaten and Weller permutation for each N, exactly the opposite is true for the lower bounds for D_N^*! Which one of these results indicate a better sequence in terms of numerical integration? Next, we compare these sequences by comparing the exact error they produce when used to integrate the function f with $s = 16$ (see (4)). Table 2 displays the absolute error against the sample size N. The choices we make for N match the values used in Fig. 9 of [18].

We observe that except for $N = 600$, the Braaten and Weller permutation error is less than or equal to the reverse permutation error. In fact, in almost all of the numerical results of this paper, the reverse permutation, together with the standard Halton sequence, gave the largest error among the digit permuted sequences.

2.1 Computing Lower Bounds for Star Discrepancy Using a Genetic Algorithm

Here we will discuss a recent genetic algorithm by Shah (see [13, 14]) that computes lower bounds for the star discrepancy. The parameters of the algorithm were determined so that the algorithm provides good estimates for the star discrepancy when applied to two types of examples. The first type of examples included a small number of low-discrepancy vectors and dimension, so that the exact star discrepancy could be computed using a brute force search algorithm. For example, the star discrepancy of the first 50 vectors of the 5-dimensional Halton sequence was computed using a brute force search algorithm. Then the genetic algorithm was run, independently, forty times to obtain forty estimates (lower bounds) for the star discrepancy. Thirty-eight of these estimates were in fact the exact discrepancy, and the remaining two were within 1.64% of the exact value.

The other type of examples Shah used to determine the algorithm parameters had larger number of vectors or dimension, and a brute force search was not practical. However, lower and upper bounds for the star discrepancy could be computed using an algorithm by Thiémard [15]. Shah used the examples and the bounds given in [15], and was able to show that the genetic algorithm consistently yielded discrepancy estimates within Thiémard's bounds.

Table 3 Lower & upper bounds for star discrepancy for different bases. Dimension is five.

D^*_{100}	Case A	Case B	Case C
Halton	$(0.110, 0.146)$	$(0.601, 0.643)$	$(0.961, 1.)$
Reverse	$(0.084, 0.130)$	$(0.401, 0.428)$	$(0.563, 0.581)$
Faure	$(0.097, 0.151)$	$(0.143, 0.186)$	$(0.185, 0.225)$
FL	$(0.115, 0.150)$	$(0.152, 0.193)$	$(0.109, 0.148)$
KW	$(0.100, 0.136)$	$(0.149, 0.179)$	$(0.124, 0.165)$
CMW	$(0.116, 0.152)$	$(0.261, 0.291)$	$(0.522, 0.556)$
Random	$(0.104, 0.152)$	$(0.146, 0.173)$	$(0.188, 0.202)$

In the next two tables, we compute lower bounds for the star discrepancy of the first 100 digit permuted Halton vectors, D^*_{100}, using the genetic algorithm. We also compute upper bounds for D^*_{100} using Thiémard's algorithm[3] [15]. For example, the first entry in Table 3, $(0.110, 0.146)$, states that the lower & upper bounds for D^*_{100} computed by the genetic algorithm and Thiémard's algorithm, were 0.110 and 0.146, respectively, for the Halton sequence (in bases given below in Case A). We consider the permutations by Vandewoestyne and Cools [18], Faure [5], Faure and Lemieux [6], Kocis and Whiten [8], Chi et al. [4], and the standard Halton sequence; these sequences are labeled as Reverse, Faure, FL, KW, CMW, and Halton, respectively, in the tables. We want to compare these digit permuted sequences with our proposed random digit permuted sequences, with respect to their star discrepancy. To do this, we generate forty sets of random permutations independently (one random permutation for each base), which gives forty random digit permuted Halton sequences. We then compute lower and upper bounds for the star discrepancy of the first 100 vectors of these sequences. The row "Random" displays the sample means of these bounds.

In Table 3, there are three cases labeled as A, B, and C. In each case, we compute D^*_{100} when the dimension of the sequence is five, however, different cases use different bases. In A, the bases of the Halton sequence are the first five prime numbers; $p_1, p_2, ..., p_5$ (p_i is the ith prime number). In B, the bases are $p_{14,}p_{20}, p_{27}, p_{33}, p_{39}$, and in C the bases are $p_{46,}p_{47}, p_{48}, p_{49}, p_{50}$. We would like to see how increasing the prime base affects the discrepancy.

When the prime bases and the dimension (which is five) are low, as in Case A, we do not expect to see the standard Halton sequence have poor star discrepancy, and the results support that. The star discrepancy intervals of the sequences are close. In Case B, we increase the prime bases, in a mixed way, and the results change considerably. Now Halton has the worst discrepancy, followed by Reverse,

[3] The complexity of Thiémard's algorithm grows at least as s/ε^s, where s is the dimension and ε is the parameter that specifies the difference between the upper and lower bounds for the star-discrepancy (see [16] for a proof of the result on complexity and [15] for empirical results on complexity). We were able to go as low as $\varepsilon = 0.05$ in Table 3, and $\varepsilon = 0.2$ in Table 4. The genetic algorithm gave tighter lower bounds than Thiémard's algorithm in computing times roughly one-fifth (Table 3) and one-fortieth (Table 4) of Thiemard's algorithm.

Table 4 Star discrepancy for different bases. Dimension is ten.

D^*_{100}	Case A	Case B	Case C	Case D
Halton	$(0.251, 0.387)$	$(0.769, 0.962)$	$(0.910, 1.000)$	$(0.860, 1.000)$
Reverse	$(0.244, 0.392)$	$(0.429, 0.569)$	$(0.485, 0.640)$	$(0.903, 0.927)$
Faure	$(0.157, 0.324)$	$(0.238, 0.395)$	$(0.209, 0.388)$	$(0.360, 0.555)$
FL	$(0.189, 0.348)$	$(0.216, 0.369)$	$(0.187, 0.332)$	$(0.317, 0.485)$
KW	$(0.171, 0.331)$	$(0.285, 0.451)$	$(0.212, 0.378)$	$(0.419, 0.573)$
CMW	$(0.184, 0.337)$	$(0.198, 0.364)$	$(0.548, 0.683)$	N/A
Random	$(0.182, 0.345)$	$(0.212, 0.373)$	$(0.259, 0.444)$	$(0.294, 0.437)$

and CMW. The permutations Faure, FL, KW, and Random are in good agreement. Further increasing the bases in Case C spreads out the values; FL gives the lowest star discrepancy, and KW, Faure and Random come next.

In Table 4 we do a similar analysis, but now the problem is slightly more difficult: the dimension of the vectors is 10. In Case A, the bases are the first ten primes, and all the discrepancy intervals overlap, although the lower bounds for Halton and Reverse are the highest. In Case B, C, and D, the bases are the ith prime numbers where $i \in \{11, 17, 21, 22, 24, 29, 31, 35, 37, 40\}$, $i \in \{41, 42, 43, 44, 45, 46, 47, 48, 49, 50\}$, and $i \in \{43, 44, 49, 50, 76, 77, 135, 136, 173, 174\}$. In Cases B and D, Halton and Reverse give the highest star discrepancy intervals, and in Case C, CMW joins them. Since permutations for CMW are available up to $p_{50} = 229$, no estimates are available in Case D. Looking at these interval estimates across each row, one notices that the random permutation yields intervals that gradually increase with bases, but slower, when compared to other permutations. In Case D, the Random permutation gives the lowest lower and upper bounds.

3 Applications

In this section we compare deterministic and random digit permuted sequences when they are applied to the numerical integration of

$$f(x_1, ..., x_s) = \prod_{i=1}^{s} (|4x_i - 2| + a_i) / (1 + a_i) .$$

In our numerical comparisons, we will proceed as follows: All digit permuted Halton sequences can be randomized by the random-start approach, which is a randomized quasi-Monte Carlo technique (see Ökten [12] and Wang and Hickernell [20]). This enables us to compute the root mean square error of estimates obtained by independently "random-starting" a given digit permuted Halton sequence. For the random permutation approach, we will apply the random-start randomization to one realization of a random permuted Halton sequence.

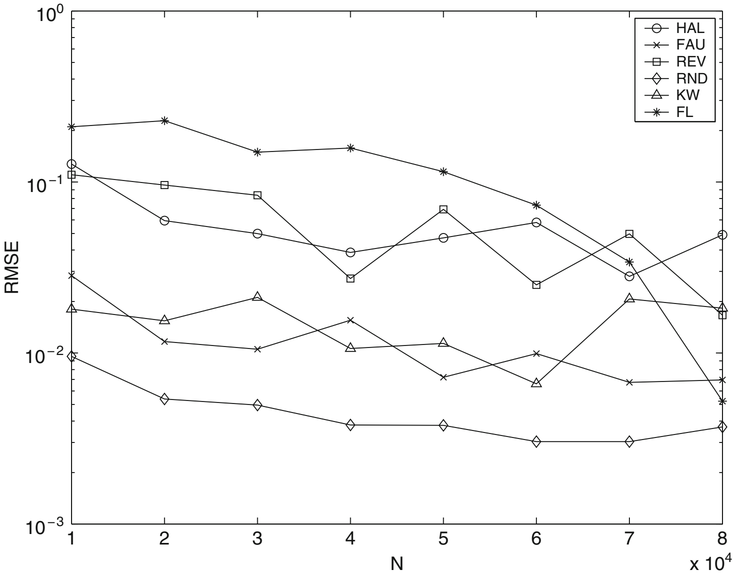


Fig. 4 Random digit permutation versus deterministic permutations. Case D.

The sensitivity of $f(x_1,...,x_s)$ to x_i depends inversely on the magnitude of the constant a_i. By appropriately choosing a_i, we can specify which components are more important, i.e., contribute more to the integral of the function. This enables us to test how well the underlying quasi-Monte Carlo sequence performs in different scenarios. For example, in Table 4, Case D, we observed that the lower bound for the star discrepancy of the random permutation approach was smaller than the lower bound for the other sequences. Case D corresponded to bases p_i where $i \in D = \{43, 44, 49, 50, 76, 77, 135, 136, 173, 174\}$. This result suggests that we might expect the random permutation approach perform relatively better in a numerical integration problem where the function heavily depends on its variables from the index set D. The test function f helps us to verify this hypothesis easily: we set $s = 10$, $a_i = 0$, and use prime bases that correspond to the indices from D in constructing the digit permuted Halton sequences. This test function can also be interpreted as a high dimensional function where the variables corresponding to indices D are the most important. Figure 4 plots the root mean square error (RMSE) of forty estimates when the Halton (HAL) sequence and the digit permuted sequences by Faure (FAU), Vandewoestyne and Cools (REV), Random (RND), Kocis and Whiten (KW), and Faure and Lemieux (FL) are randomized via the random-start method. The random permutation approach gives the lowest RMSE for all samples. FL gives the worst estimates (except for the last two samples), followed by HAL and REV. Permutations FAU and KW give close results that are better than FL, HAL, REV, except for the last sample.

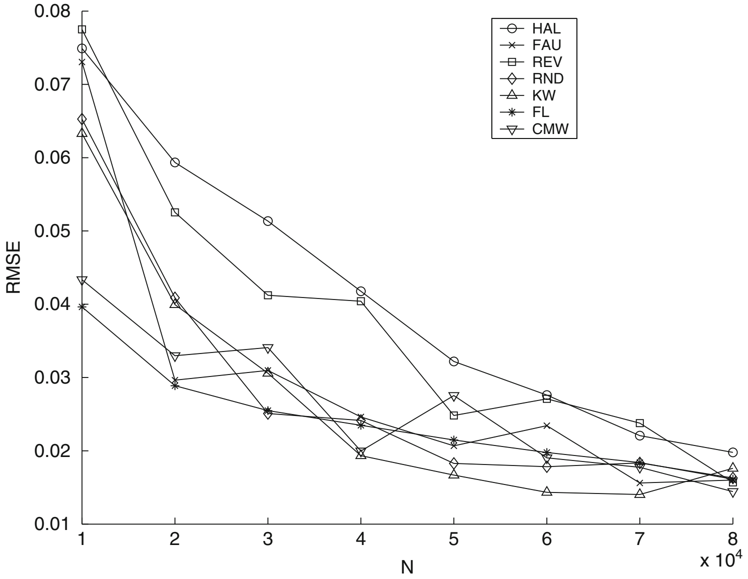


Fig. 5 Random digit permutations versus deterministic permutations. Mean dimension (in the truncation sense) of 3.52.

We next consider $s = 20$, and generate a set of 20 random constants a_i from $\{0, 1, 2\}$, conditional on obtaining a mean dimension larger than 3.5. For a definition of mean dimension see [22].

We obtained $a = \{0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 2, 1, 2, 0, 1, 0, 2, 0, 1, 0\}$, and a mean dimension (in the truncation sense [22]) of 3.52. Figure 5 plots the RMSE of forty estimates generated via the random-start method. HAL and REV has the worst overall performance. It is not easy to separate the other permutations in terms of error, except for the very first sample. The prime bases used to obtain the results in Fig. 5 were the first 20 prime numbers.

4 Conclusions

Deterministic permutations designed to improve the uniformity of the Halton sequence have been around since the 1970s. Although various numerical experiments have been used to show the benefits of these sequences over the Halton sequence, the simple question of how such a sequence compares with a randomly permuted sequence has not been addressed in the literature. We computed interval estimates for the star discrepancy, and used a test problem from numerical integration to compare randomly permuted Halton sequences with some selected deterministic sequences. We performed additional numerical experiments that are not reported here due to space limitations. Quite surprisingly, in the problems we considered, we have found that the random permutation approach was as good as, or better, than the "best" deterministic permutations.

Acknowledgements We thank Dr. Hongmei Chi for supplying us with the permuted Halton sequence code used in Chi et al. [4]. We also thank the anonymous referees for their helpful comments.

References

1. E. I. Atanassov, On the discrepancy of the Halton sequences, *Math. Balkanica, New Series* **18** (2004) 15–32.
2. E. Braaten, G. Weller, An improved low-discrepancy sequence for multidimensional quasi-monte Carlo integration, *Journal of Computational Physics* **33** (1979) 249–258.
3. H. Chaix, H. Faure, Discrepance et diaphonie en dimension un, *Acta Arithmetica* LXIII (1993) 103–141.
4. H. Chi, M. Mascagni, T. Warnock, On the optimal Halton sequence, *Mathematics and Computers in Simulation* **70** (2005) 9–21.
5. H. Faure, Good permutations for extreme discrepancy, *Journal of Number Theory* **42** (1992) 47–56.
6. H. Faure, C. Lemieux, Generalized Halton sequences in 2008: A comparative study, *ACM Transactions on Modeling and Computer Simulation* **19** (2009) 15:1–31.
7. Y. Goncharov, G. Ökten, M. Shah, Computation of the endogenous mortgage rates with randomized quasi-Monte Carlo simulations, *Mathematical and Computer Modelling* **46** (2007) 459–481.
8. L. Kocis, W. J. Whiten, Computational investigations of low-discrepancy sequences, *ACM Transactions on Mathematical Software* **23** (1997) 266–294.
9. J. Matoušek, On the L_2-discrepancy for anchored boxes, *Journal of Complexity* **14** (1998) 527–556.
10. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.
11. L. Kuipers and H. Niederreiter *Uniform Distribution of Sequences*, Dover Publications, Mineola, NY, 2006.
12. G. Ökten, Generalized von Neumann-Kakutani transformation and random-start scrambled Halton sequences, *Journal of Complexity* **25** (2009) 318–331.
13. M. Shah, A genetic algorithm approach to estimate lower bounds of the star discrepancy, Monte Carlo Methods and Applications, *Monte Carlo Methods Appl.* **16** (2010) 379–398.
14. M. Shah, Quasi-Monte Carlo and Genetic Algorithms with Applications to Endogenous Mortgage Rate Computation, Ph.D. Dissertation, Department of Mathematics, Florida State University, 2008.
15. E. Thiémard, An algorithm to compute bounds for the star discrepancy, *Journal of Complexity* **17** (2001) 850–880.
16. M. Gnewuch, Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy, *Journal of Complexity* **24** (2008), 154–172.
17. B. Tuffin, A New Permutation Choice in Halton Sequences, in: H. Niederreiter, P. Hellekalek, G. Larcher, and P. Zinterhof, editors, Monte Carlo and Quasi-Monte Carlo Methods 1996, Vol 127, Springer Verlag, New York, 1997, pp 427–435.
18. B. Vandewoestyne, R. Cools, Good permutations for deterministic scrambled Halton sequences in terms of L_2-discrepancy, *Journal of Computational and Applied Mathematics* **189** (2006) 341–361.

19. T. T. Warnock, Computational Investigations of Low-discrepancy Point Sets II, in: Harald Niederreiter and Peter J.-S. Shiue, editors, Monte Carlo and quasi-Monte Carlo methods in scientific computing, Springer, New York, 1995, pp. 354–361.
20. X. Wang, F. J. Hickernell, Randomized Halton sequences, *Mathematical and Computer Modelling* **32** (2000) 887–899.
21. I. Radovic, I. M. Sobol', R. F. Tichy, Quasi-Monte Carlo methods for numerical integration: Comparison of different low-discrepancy sequences, *Monte Carlo Methods Appl.* **2** (1996) 1–14.
22. R. E. Caflisch, W. Morokoff, A. B. Owen, Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension, *Journal of Computational Finance* **1** (1997) 27–46.

A Quasi Monte Carlo Method for Large-Scale Inverse Problems

Nick Polydorides, Mengdi Wang, and Dimitri P. Bertsekas

Abstract We consider large-scale linear inverse problems with a simulation-based algorithm that approximates the solution within a low-dimensional subspace. The algorithm uses Tikhonov regularization, regression, and low-dimensional linear algebra calculations and storage. For sampling efficiency, we implement importance sampling schemes, specially tailored to the structure of inverse problems. We emphasize various alternative methods for approximating the optimal sampling distribution and we demonstrate their impact on the reduction of simulation noise. The performance of our algorithm is tested on a practical inverse problem arising from Fredholm integral equations of the first kind.

1 Introduction

Many problems in computational science and engineering are characterized by experimental design, measurement acquisition, and parameter estimation or prediction. This process involves mathematical modeling of the physical systems pertinent to the observations, as well as estimation of unknown model parameters from the acquired measurements by formulating and solving an inverse problem. Quite often solving the inverse problem subject to measurement errors and model uncertainties becomes computationally prohibitive, particularly for high-dimensional parameter spaces and precise forward models [5].

In this paper we consider ill-posed inverse problems that upon discretization yield large systems of linear equations. Such problems formulated as Fredholm integral

N. Polydorides
EEWRC, The Cyprus Institute, Nicosia, Cyprus
e-mail: nickpld@cyi.ac.cy

M. Wang · D. P. Bertsekas
LIDS, MIT, Cambridge, MA, USA
e-mail: mwang@mit.edu; dimitrib@mit.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_36, © Springer-Verlag Berlin Heidelberg 2012

equations of the first kind typically arise in several areas of engineering and natural science including image processing, geophysical prospecting and wave scattering [11]. The main characteristic of these problems is that the integral operator that maps the model parameters to the observed data does not have a continuous inverse and thus a small amount of noise in the data may trigger an arbitrarily large variation in the estimated parameters. This inherent property of ill-posed problems is reflected also in the discrete problem setting causing the coefficients matrix of the respective linear system to be ill-conditioned or singular. Consider for example the integral equation

$$b(y) = \int_{x_1}^{x_2} \mathrm{d}x \, \alpha(x, y) f(x) + \eta(y)$$

to which we associate, through a numerical integration rule, the linear model

$$b = Af + \eta \tag{1}$$

where $A \in \Re^{m \times n}$ is a dense ill-conditioned matrix, $b \in \Re^m$ is the data vector, $f \in \Re^n$ is the discretization of the unknown function and $\eta \in \Re^m$ is some additive noise. In order to enforce stability in estimating f from noisy data b one may apply Tikhonov regularization, expressed as a penalized least-squares problem

$$\min_{f \in \Re^n} \|b - Af\|_\zeta^2 + \lambda \|f\|^2, \tag{2}$$

where $\zeta \in \Re^m$ is a known probability distribution with positive components and $\lambda \in \Re$ is a positive regularization parameter. This problem is shown to have a unique regularized solution f_t, obtained by solving the linear system

$$(A'ZA + \lambda I) f_t = A'Zb,$$

where $Z \in \Re^{m \times m}$ is the diagonal matrix based on ζ, I is the identity matrix and prime denotes transposition. The value of λ is chosen such that $(A'ZA + \lambda I)$ is full rank and well-conditioned for inversion [2]. When n or m is very large, computing f_t becomes challenging, hence we propose to approximate f_t within a low-dimensional subspace

$$S = \big\{ \Phi r \,|\, r \in \Re^s \big\},$$

where $\Phi \in \Re^{n \times s}$ is a matrix whose columns represent the s discrete basis functions spanning S. The type of basis functions can be arbitrary but we assume throughout that Φ has rank s. Our proposed methodology involves subspace approximation, Monte-Carlo simulation, regression, and most significantly, *only* low-dimensional vector operations, e.g. of order s. Let $\Pi : \Re^n \mapsto S$ be an orthogonal projection operator. By decomposing f to its orthogonal components, $f = \Pi f + (I - \Pi) f$, we have

$$b = A\big(\Pi f + (I - \Pi) f\big) + \eta = A \Pi f + \epsilon, \tag{3}$$

where the error term $\epsilon = A(I - \Pi)f + \eta$ encompasses the impact of subspace approximation and the additive noise. By representing Πf as Φr, and applying a Galerkin projection to S weighted by ζ, we obtain

$$c = Gr + z \tag{4}$$

where

$$c = \Phi' A' Z b, \qquad G = \Phi' A' Z A \Phi \qquad z = \Phi' A' Z \epsilon.$$

The new projected operator $G \in \Re^{s\times s}$ is now of moderate dimension but is typically still ill-conditioned and may not be invertible. Suppose that instead of evaluating G and c by performing the high-dimensional matrix products, we use estimators $\hat{G}$ and $\hat{c}$ obtained by stochastic simulation. In such case we formulate the linear model

$$\hat{c} = \hat{G}r + w, \quad \text{where} \quad w = z + (\hat{c} - c) + (G - \hat{G})r. \tag{5}$$

Then an approximate solution r^* can be computed from the regularized regression

$$\min_{r\in\Re^s} \|\hat{G}r - \hat{c}\|^2_{\Sigma^{-1}} + \lambda\|r - \bar{r}\|^2, \tag{6}$$

where $\Sigma \in \Re^{m\times m}$ is the noise covariance matrix of w and $\bar{r}$ is an initial guess on the solution. With minimal loss of generality we assume that η and ϵ are random variables with zero mean. The simulation-based regularized problem (6) admits the unique solution

$$\hat{r} = (\hat{G}'\Sigma^{-1}\hat{G} + \lambda I)^{-1}(\hat{G}'\Sigma^{-1}\hat{c} + \lambda\bar{r}), \tag{7}$$

although, because w is a function of r (cf. (5)), the noise covariance Σ is a function of the required solution. To overcome this problem one option is to evaluate a constant covariance based on a nominal r, such as the prior for example, yielding $\Sigma = \Sigma(\bar{r})$. Another possibility is a form of iterative regularized regression, whereby we iteratively estimate the optimal solution using an intermediate correction of $\Sigma(r)$ as

$$\hat{r}_{k+1} = \big(\hat{G}'\Sigma(\hat{r}_k)^{-1}\hat{G} + \lambda I\big)^{-1}\big(\hat{G}'\Sigma(\hat{r}_k)^{-1}\hat{c} + \lambda\bar{r}\big), \tag{8}$$

for $k \geq 0$ and $r_0 = \bar{r}$. The iteration was shown to converge locally to a fixed point of (8), provided that a matrix Euclidean norm of $\Sigma(r)$ is sufficiently small [19]. The estimation of $\hat{G}$, $\hat{c}$ and $\Sigma(\hat{r}_k)$ using stochastic simulation is addressed next.

2 Approximation Based on Simulation and Regression

Our approach is based on stochastic simulation. We note that there is a large body of work on the solution of linear systems using Monte Carlo methods, starting with a suggestion by von Neumann and Ulam, as recounted by Forsythe and Leibler [10], (see also Curtiss [6] and the survey by Halton [12]). For a thorough review of the methods including some important recent developments we refer the readers to the books by Asmussen et al. [1] and Lemieux [15].

Our approach differs from the works just mentioned in that it involves not only simulation, but also approximation of the solution within a low-dimensional subspace in the spirit of Galerkin approximation (see e.g. [5]). We also like to draw the distinction from Markov chain Monte Carlo methods used in the context of linear Bayesian estimation, where the *a posteriori* probability distribution is sampled using, for example, the Metropolis-Hastings or the Gibbs algorithms [14]. Our work is related to the approximate dynamic programming methodology that aims to solve forms of Bellman's equation of very large dimension by using simulation (see the books by Bertsekas and Tsitsiklis [3], and by Sutton and Barto [18]). This methodology has recently been extended to general square systems of linear equations and regression problems in a paper by Bertsekas and Yu [4], which served as a starting point for the present paper.

The use of simulation for linear algebra operations has also been adopted by Drineas et al. in a series of papers [7–9] in the context of randomized algorithms for massive dataset analysis. The authors propose sampling the entries of large matrices, in order to construct new sparser or smaller matrices that behave like the high-dimensional ones. In their analysis they consider products of several matrices where they randomly take samples according to an importance sampling distribution that relates to the Euclidean norms of the columns. In their work they make no assumptions on the matrices, as opposed to our methodology, which applies primarily to matrices of smooth structure like those arising from discretization of Fredholm kernels.

2.1 Markov Chain Monte Carlo Framework

In [4] the authors suggest generating a long finite sequence of indices $\{i_0, \dots, i_t\}$ according to a nominal probability distribution ξ and two sequences of transitions $\{(i_0, j_0), \dots, (i_t, j_t)\}$ and $\{(i_0, h_0), \dots, (i_t, h_t)\}$ according to some transition probabilities ρ_{ij} and ρ_{ih} respectively. This yields estimates of the low-dimensional G and c as

$$\hat{G} = \frac{1}{t+1} \sum_{p=0}^{t} \frac{\zeta_{i_p} a_{i_p j_p} a_{i_p h_p}}{\xi_{i_p} \rho_{i_p j_p} \rho_{i_p h_p}} \phi_{j_p} \phi'_{h_p}, \qquad \hat{c} = \frac{1}{t+1} \sum_{p=0}^{t} \frac{\zeta_{i_p} a_{i_p j_p} b_{i_p}}{\xi_{i_p} \rho_{i_p j_p}} \phi_{j_p}, \tag{9}$$

where a_{ij} denotes the (i, j)th component of A, and ϕ'_j is the jth row of Φ, assuming that $\rho_{ij} > 0$ whenever $a_{ij} > 0$. Apart from the estimators one obtains a sample-based estimator of the covariance given by

$$\Sigma(\hat{r}_k) = \frac{1}{t+1} \sum_{p=0}^{t} w_p w'_p = \frac{1}{t+1} \sum_{p=0}^{t} \big((G_p - \hat{G})\hat{r}_k + (\hat{c} - c_p)\big)\big((G_p - \hat{G})\hat{r}_k + (\hat{c} - c_p)\big)', \tag{10}$$

where each w_p can be viewed as a sample of w, while G_p and c_p denote the corresponding sample terms averaged to yield $\hat{G}$ and $\hat{c}$. For further discussion and a derivation of a confidence region for $\hat{r}_k$ obtained by introducing (9) and (10) into (8) we refer to [19]. For the needs of this work, we borrow an important result from [4], in the form of the following theorem.

Theorem 1. *As $t \to \infty$ we have $\hat{G} \to G$, and $\hat{c} \to c$ with probability 1, where $\hat{G}$ and $\hat{c}$ are given by (9).*

Proof. The proof is in [4].

Remark 1. Under the conditions of Theorem 1, if the eigenvalues of the sample-based covariance $\Sigma(\hat{r}_k)$ are bounded below by a positive scalar, then iteration (8) yields $\hat{r}_k \to r^*$ with probability 1, where Φr^* is the target high-dimensional regularized solution.

2.2 Variance Reduction by Importance Sampling

The central idea of our simulation method is to evaluate G and c as weighted averages of samples generated by a probabilistic mechanism. In this context, a critical issue is the reduction of the variance of the estimation errors $\hat{G} - G$ and $\hat{c} - c$. To achieve this goal we use importance sampling, which can be shown to yield estimators of minimal variance when an *optimal* probability distribution is used for generating the samples [12]. Let Ω be a discrete sample space, $\nu : \Omega \mapsto \Re$ be a function and $\{i_0, i_1, \dots, i_t\}$ be the sequence of samples generated from Ω independently according to distribution ξ. Then consider estimating the large sum $u = \sum_{i \in \Omega} \nu_i$ as

$$\hat{u} = \frac{1}{t+1} \sum_{p=0}^{t} \frac{\nu_{i_p}}{\xi_{i_p}},$$

and designing ξ so that the variance of $\hat{u}$ is minimized. If ν is nonnegative, the variance is

$$\text{var}\{\hat{u}\} = \frac{u^2}{t+1}\left(\sum_{\omega \in \Omega} \frac{\big(\nu(\omega)/u\big)^2}{\xi(\omega)} - 1\right),$$

from where it is now apparent that the choice $\xi^* = \nu u^{-1}$ is the optimal zero-variance sampling distribution. Note that the non-negativity of ν is not critical, for if ν admits negative values, it is trivial to decompose as $\nu = \nu^+ - \nu^-$ so that both ν^+ and ν^- are positive functions. In such a situation $\hat{u}$ is computed by estimating separately $u^+ = \sum_{i \in \Omega} \nu_i^+$ and $u^- = \sum_{i \in \Omega} \nu_i^-$. As is well known, calculating the optimal ξ^* is impractical since it requires knowledge of the unknown sum. However, designing a computationally tractable approximation $\hat{\xi}$ that nearly minimizes the L_1 norm $\|\xi - \nu u^{-1}\|_1$ can be shown to reduce the variance of $\hat{u}$. In the remaining part of this section we discuss some schemes for designing sampling distributions tailored to the data of the linear ill-posed inverse problems, so to achieve variance reduction.

2.2.1 Designing Importance Sampling Distributions with Polynomial Bases

We focus on estimating the (l,q)th entry of the symmetric, $s \times s$ matrix G and the lth element of vector c independently in an element by element fashion. Noticing that these can be expressed as high-dimensional sums (of dimensions n^3 and n^2 respectively)

$$G_{lq} = \phi_l' A' Z A \phi_q = \sum_{i=1}^{n} \zeta_i \Big(\sum_{j=1}^{n} a_{ij}\phi_{jl} \Big) \Big(\sum_{h=1}^{n} a_{ih}\phi_{hq} \Big), \tag{11}$$

$$c_l = \phi_l' A' Z b = \sum_{i=1}^{n} \zeta_i \Big(\sum_{j=1}^{n} a_{ij}\phi_{jl} \Big) b_i . \tag{12}$$

One may consider a sequence of independent *uniformly* distributed sample indices $\{(i_p, j_p, h_p)\}_{p=0}^{t}$ and $\{(i_p, j_p)\}_{p=0}^{t}$ from the spaces $[1,n]^3$ and $[1,n]^2$, and compute the Monte Carlo estimators

$$\hat{G}_{lq} = \frac{1}{t+1} \sum_{p=0}^{t} \frac{\zeta_{i_p} a_{i_p j_p} a_{i_p h_p} \phi_{j_p l} \phi_{h_p q}}{n^{-3}}, \quad \hat{c}_l = \frac{1}{t+1} \sum_{p=0}^{t} \frac{\zeta_{i_p} a_{i_p j_p} \phi_{j_p l} b_{i_p}}{n^{-2}}.$$

Alternatively, one may design an importance sampling distribution customized for G_{lq} as in (11).[1] In this case let the sample space be $\Omega = [1,n]^3$ and consider the function

$$\nu(i,j,h) = \zeta_i a_{ij} a_{ih} \phi_{jl} \phi_{hq},$$

assuming for simplicity that $\nu(i,j,h)$ is nonnegative. The aim here is to construct, in a computationally efficient manner, a sampling distribution $\hat{\xi}$ that approximates the optimal

$$\xi^*_{G_{lq}}(i,j,h) = \frac{\nu(i,j,h)}{G_{lq}}, \quad \text{where } G_{lq} = \sum_{i,j,h=1}^{n} \nu(i,j,h) = \sum_{i=1}^{n} \zeta_i \, \|a_i \phi_l\|_1 \, \|a_i \phi_q\|_1$$

and belongs to some family of relatively simple distribution functions. In the above a_i is the ith row of A and $\|a_i \phi_l\|_1$ is the L_1 norm of the Hadamard product of a_i and ϕ_l. As it now becomes apparent, ξ^* is not only high-dimensional and impractical

[1] Unless otherwise stated, from now on we deal exclusively with G_{lq}. A simplified analysis applies to c_l.

to compute, store and sample, but it also requires n-dimensional vector products and sums. Using Bayes' theorem and the conditional probability law the optimal distribution can be reformulated in a product form as

$$\xi^*(i,j,h) = \xi(h|i,j)\xi(i,j) = \xi(h|i,j)\xi(j|i)\xi(i), \tag{13}$$

where the marginal distributions are $\xi(i,j) = \sum_{h=1}^{n} \xi(i,j,h)$ and $\xi(i) = \sum_{j=1}^{n} \xi(i,j)$. We propose to approximate ξ^* by approximating the constituent sampling distributions

$$\xi(i) = \frac{\nu_{hj}(i)}{G_{lq}}, \quad \xi(j|i) = \frac{\nu_h(i|j)}{\sum_{i=1}^{n} \nu_h(i,j)}, \quad \xi(h|i,j) = \frac{\nu(i,j,h)}{\sum_{i,j=1}^{n} \nu(i,j,h)}, \tag{14}$$

corresponding to the functions

$$\nu_{hj}(i) = \sum_{j=1}^{n} \nu_h(i,j), \quad \nu_h(i,j) = \sum_{h=1}^{n} \nu(i,j,h), \quad \nu(i,j,h) = \zeta_i a_{ij} a_{ih} \phi_{jl} \phi_{hq}. \tag{15}$$

To accomplish this assume a low-dimensional discretization of the sampling space, for example a uniform cubical grid. For instance let $\Omega = \Omega^k \times \Omega^k \times \Omega^k$, where $\Omega^k = \cup_{i=1}^{K} \Theta_i$, and $\Theta_1, \ldots, \Theta_K$ are K connected disjoint subsets of $[1,n]$. Moreover, let $\psi_i : \Theta_i \to \Re$ be a polynomial function with support over Θ_i and let I_{Θ_i} denote a small nonempty set of points in Θ_i, for $i = 1, \ldots, K$. Then one can approximate ν by $\tilde{\nu}$ using ψ_i and samples of ν at I_{Θ_i}. If ψ_i is a constant function then I_{Θ_i} requires only one point, whereas if it is linear then two sample points are needed in each Θ_i and so on with higher degree polynomials. The advantage of using polynomial bases is that the approximate functions in (15) can be summed up to yield the probability distributions in (14) without element-wise summation, since the sums of discrete polynomial functions can be evaluated analytically. It is now easy to see that as the grid dimension grows, i.e. $K \to n$, then $\tilde{\nu} \to \nu$, so that the approximate ξ will converge to the optimum ξ^*, albeit with an increase of computational complexity. The suitability of the proposed importance sampling scheme relies predominantly on the ease of forming $\tilde{\nu}$ using a relatively small K so that $\|\tilde{\nu} - \nu\|_1$ is small and therefore so is $\|\hat{\xi} - \xi^*\|_1$. This is fundamentally due to the smooth structure of ν, a property that stems from the smooth structure of the model matrix A in (1) (resp. the Fourier series of the Fredholm kernel $\alpha : f \mapsto b$), which *always* holds true in linear ill-posed inverse problems [11]. Once $\hat{G}$ and $\hat{c}$ are estimated, the low-dimensional solution can be computed by (7) or (8). Moreover, since the components of G and c are estimated independently, one may view the samples of G as vectors in $\Re^{s^2}$ that are independent of the samples of c. Thus we can estimate the simulation error covariance by

$$\Sigma(\hat{r}) = \Sigma_c + \begin{bmatrix} \hat{r}' & 0 & \dots & 0 \\ 0 & \hat{r}' & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \hat{r}' \end{bmatrix} \quad \Sigma_G \quad \begin{bmatrix} \hat{r} & 0 & \dots & 0 \\ 0 & \hat{r} & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \hat{r} \end{bmatrix}, \tag{16}$$

where $\Sigma_c \in \Re^{s \times s}$ is the sample-based covariance of c and $\Sigma_G \in \Re^{s^2 \times s^2}$ is the sample-based covariance of G, which is given by

$$\Sigma_G = \begin{bmatrix} \mathrm{cov}(\hat{g}'_1, \hat{g}'_1) & \mathrm{cov}(\hat{g}'_1, \hat{g}'_2) & \dots & \mathrm{cov}(\hat{g}'_1, \hat{g}'_s) \\ \mathrm{cov}(\hat{g}'_2, \hat{g}'_1) & \mathrm{cov}(\hat{g}'_2, \hat{g}'_2) & \dots & \mathrm{cov}(\hat{g}'_2, \hat{g}'_s) \\ \ddots & \ddots & \dots & \ddots \\ \mathrm{cov}(\hat{g}'_s, \hat{g}'_1) & \mathrm{cov}(\hat{g}'_s, \hat{g}'_2) & \dots & \mathrm{cov}(\hat{g}'_s, \hat{g}'_s) \end{bmatrix},$$

where $\mathrm{cov}(\hat{g}'_i, \hat{g}'_j)$ is the sample covariance between the ith and jth rows of $\hat{G}$.

2.2.2 The Simulation Algorithm

The resulting importance sampling (IS) algorithm for estimating G_{lq} is summarized as follows:

1. Divide the sampling space $[1, n]$ into K disjoint intervals $\Theta_1, \dots, \Theta_K$.
2. Fix $d + 1$ points I_{Θ_i} in Θ_i, $i = 1, \dots, K$, with $d \geq 0$.
3. Choose the bases of dth order polynomial functions $\psi_i : \Theta_i \mapsto \Re$, $i = 1, \dots, K$.
4. Form the weights matrix $N \in \Re^{(d+1)K \times (d+1)K \times (d+1)K}$ by evaluating $\nu(i, j, h)$ at $I_{\Theta_i} \times I_{\Theta_j} \times I_{\Theta_h}$, for $i, j, h = 1, \dots, K$.
5. Sum N over the h-dimension to get $N_h \in \Re^{(d+1)K \times (d+1)K}$.
6. Sum N_h over the j-dimension to get $N_{hj} \in \Re^{(d+1)K}$.
7. For $p = 0, \dots, t$:
   a. Evaluate the sum $Q = \sum_{i=1}^{(d+1)K} |N_{hj}(i)|$, construct distribution $q(i) = |N_{hj}(i)|/Q$ and take sample s_i from $\cup_{i=1}^{K} I_{\Theta_i}$ according to distribution q.
   b. Let I_{Θ_l} be the set containing s_i, construct the distribution q_l over Θ_l by interpolating with the bases ψ. Sample i_p from Θ_l according to distribution q_l with probability P_{i_p}.
   c. Evaluate the sum $Q = \sum_{j=1}^{(d+1)K} |N_h(s_i, j)|$, and construct $q(j) = |N_h (s_i, j)|/Q$ and take sample s_j from $\cup_{i=1}^{K} I_{\Theta_i}$ according to q.
   d. Let I_{Θ_m} be the set containing s_j, and construct distribution q_m over Θ_m by interpolating. Sample j_p from Θ_m according to q_m with probability P_{j_p}.
   e. Evaluate the sum $Q = \sum_{h=1}^{(d+1)K} |N(s_i, s_j, h)|$, and construct $q(h) = |N(s_i, s_j, h)|/Q$ and take sample s_h from $\cup_{i=1}^{K} I_{\Theta_i}$ according to q.

   f. Let I_{Θ_n} be the set containing s_h, and construct distribution q_n over Θ_n by interpolating. Sample h_p from Θ_n according to q_n with probability P_{h_p}.
   g. Register sample (i_p, j_p, h_p) with probability $\xi(i_p, j_p, h_p) = P_{i_p} P_{j_p} P_{h_p}$ and evaluate $\nu_p = \zeta_{i_p} a_{i_p j_p} a_{i_p h_p} \phi_{j_p l} \phi_{h_p q}$.
   h. Evaluate pth sample mean:

      i. $\hat{G}_{lq} = \nu_p / \xi(i_p, j_p, h_p)$ if $p = 0$.
      ii. $\hat{G}_{lq} = \frac{p}{p+1} \hat{G}_{lq} + \frac{1}{p+1} \nu_p / \xi(i_p, j_p, h_p)$ if $p > 0$.

8. End sampling.
9. Evaluate t-sample variance var($\hat{G}_{lq}$).
10. Evaluate the total error covariance using (16).
11. Compute the solution approximation from (7) or (8).

3 Discrete Linear Inverse Problems

Linear ill-posed inverse problems typically occur in applications of image processing, emission tomography, wave diffraction, palaeo-climatology, and heat transfer, and are usually expressed in Fredholm integral equations. Discretizing these equations yields linear systems with ill-conditioned coefficient matrices. This is an inherent characteristic of ill-posed problems and has been analyzed in various publications, including [11] and [2] which emphasize its implications to the existence, uniqueness and stability of the solution. In particular, the condition number of the coefficient matrix obtained by discretization can be shown to increase with the dimension n, sometimes at an exponential rate in which case the problem is said to be heavily ill-posed.

In our development we have assumed the structure of the matrix A to be smooth, implying that neighboring entries have almost identical values. This property is due to the spectral properties of the Fredholm operators in consideration. Figure 1 illustrates this effect on a moderately sized discretized kernel $A \in \Re^{n \times n}$ with $n = 10^3$ for a problem arising from geophysics. A large-scale numerical study based on this model problem is investigated next.

3.1 *Test Example: Gravitational Prospecting*

Gravitational prospecting is a problem typically encountered in hydrocarbon exploration. Suppose a mass of density $f(\theta)$ is distributed on a circular ring O_i of radius r_i centered at the origin, where $0 \leq \theta \leq 2\pi$. Allow also a concentric circle O_o of radius r_o, with $r_o \ll r_i$ lying on the same plane, where the centrally directed component of gravitational force $b(\phi)$ is measured, for $0 \leq \phi \leq 2\pi$. According to the law of cosines the squared distance between a mass element situated on O_i at an angle θ and a point of O_o at ϕ is

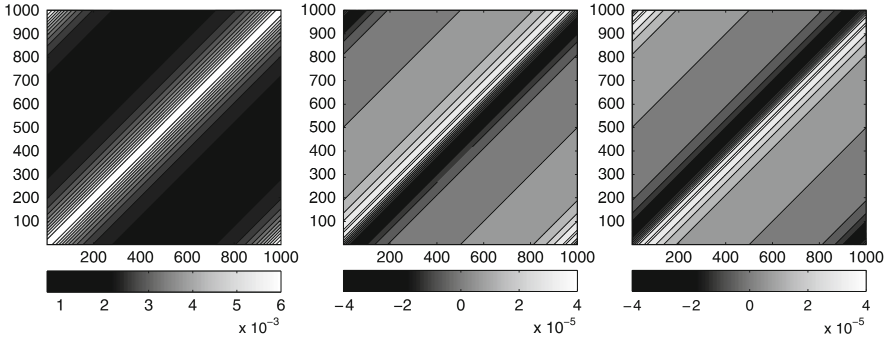


Fig. 1 Contour plots of the elements of A (*left*) for a smaller scale problem with $n = 10^3$ and its gradients in row index i (*middle*) and column index j (*right*) to indicate the smooth structure of the model matrix as manifested by the flat regions in the gradient plots. At dimension $n = 10^3$ the condition number of the A is 2.2×10^{20}.

$$\rho_t^2 = r_o^2 + r_i^2 - 2r_o r_i \cos(\theta - \phi),$$

while the angle χ formed between the normal component of the gravity force at ϕ and the line connecting that point to the gravitating element $f(\theta)\mathrm{d}\theta$ of the inner ring satisfies

$$\rho_t \cos(\chi) = r_o - r_i \cos(\theta - \phi).$$

In effect, the overall gravitational force exerted at the measuring angle ϕ is

$$b(\phi) = \gamma \int_0^{2\pi} \mathrm{d}\theta \frac{1}{2\rho_t^2} \cos(\chi) f(\theta),$$

where γ is the universal gravity constant. Taking for simplicity $\gamma = 1$, $r_o = 1$ and $r_i = 0.5$ yields the Fredholm equation

$$b(\phi) = \int_0^{2\pi} \mathrm{d}\theta \; \alpha(\phi, \theta) \; f(\theta), \qquad 0 \le \phi \le 2\pi,$$

with kernel

$$\alpha(\phi, \theta) = \frac{2 - \cos(\phi - \theta)}{\big(5 - 4\cos(\phi - \theta)\big)^{3/2}}.$$

The integral equation is discretized using a midpoint quadrature rule on a uniform grid with $n = 10^6$ points $\{\theta_i, \phi_j\}_{i,j=1}^n$ spanning over $[0, 2\pi] \times [0, 2\pi]$. To approximate the solution we choose a subspace spanned by an orthogonal basis of $s = 10^2$ piecewise constant functions, and consider reconstructing a subspace approximation of the regularized density f given data $b \in \Re^n$. To test the performance of the proposed scheme in reducing the simulation noise we run the Algorithm 2.2.2 for various t and K, each time using piecewise constant, linear

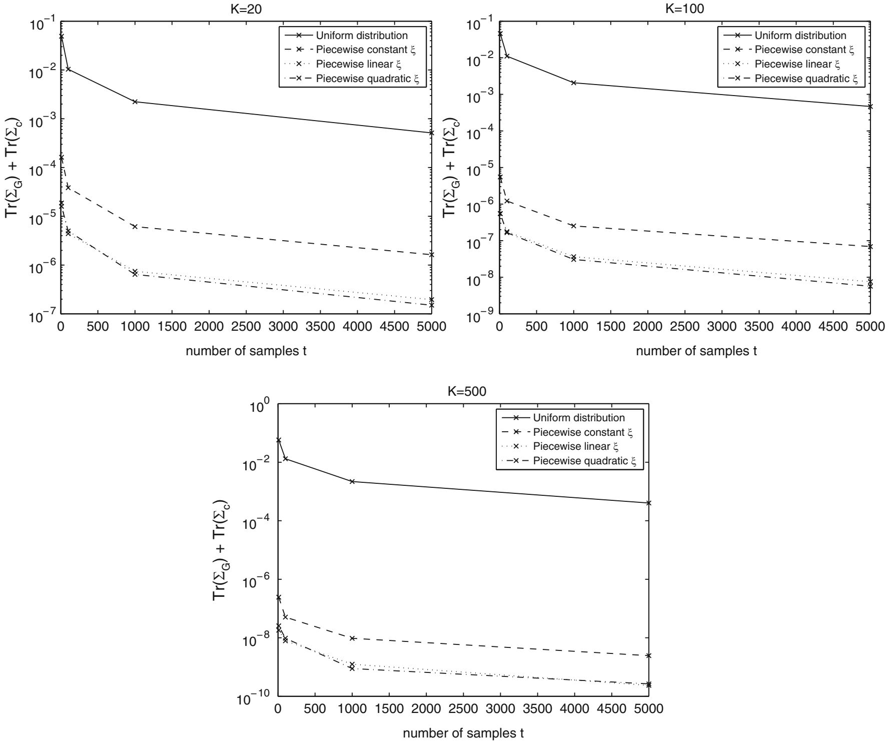


Fig. 2 Reduction in simulation noise $Tr(\Sigma_G) + Tr(\Sigma_c)$ with the number of acquired samples. From the *left* the cases with $K = 20, 100$ and 500 intervals, assuming $s = 10^2$ and $n = 10^6$. In each graph the *solid line* is with the naive Monte Carlo sampling (uniform distribution), the *dashed* for a piecewise constant approximation of the optimum IS distribution ξ^*, the *dotted* for a piecewise linear and the *dash-dotted* for a quadratic approximation of the optimum IS distribution. Notice that the simulation error reduces in increasing K and in implementing a higher-order approximation of the optimal IS distribution. In all cases the proposed scheme outperforms the naive Monte Carlo sampling in reducing the variance of the estimators.

and quadratic basis functions for approximating the optimal importance sampling distribution. The graphs of Fig. 2 illustrate the reduction of the simulation noise, quantified in terms of the sum of the traces of the two sample-based covariances as it is affected by t and K. Notice that $Tr(\Sigma_G) + Tr(\Sigma_c)$ reduces with increasing the number of samples and/or the degree of the polynomials ψ used in approximating the optimal distribution. The corresponding graphs obtained with uniform sampling are plotted for comparison in order to show the superiority of the importance sampling scheme.

In our numerical tests we choose not to add any measurement noise, i.e. $\eta = 0$. When solving ill-posed inverse problems with synthetic data the noise free case is considered as a "contrived" example as it allows for processing unrealistically precise data, (see for example Chap. 5 in [14]). On the other hand, there is a large

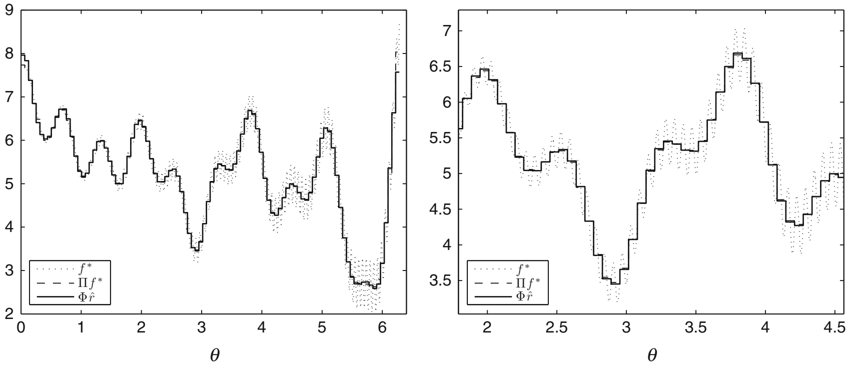


Fig. 3 *Left*, the true image f^*, its projection Πf^* in S spanned by $s = 10^2$ piecewise constant orthogonal basis functions, and the result of the simulated approximation $\Phi\hat{r}$. Angle $\theta \in [0, 2\pi]$ is discretized in $n = 10^6$ elements. To the right a more detailed view of the results for $2 \leq \theta \leq 4.5$. Notice that the curves of Πf^* and $\Phi\hat{r}$ are almost overlapping.

body of literature on how to adjust the regularization parameter λ in (6) so as to counteract the impact of noise and stabilize the solution. An extensive survey of such methods is presented in Chap. 5 of [13]. In particular, notice that the problem under consideration involves a square linear system of manageable dimension, where the data vector $\hat{c}$ and coefficients matrix $\hat{G}$ include simulation errors for which we can estimate their element-wise variance based on samples. Moreover, $\hat{G}$ is symmetric and ill-conditioned and the overall noise includes the subspace approximation error and any additive noise contained in the original data b. In this context, to choose λ we adopt the discrete Picard condition [17], which relies on the singular value decomposition of the low-dimensional $\Sigma(\hat{r}_k)^{-1/2}\hat{G}$, implemented after each iteration (8).

In this study we focus on demonstrating the performance of the Algorithm 2.2.2 in estimating G and c with reduced simulation error. In particular, our claim is that for $\eta = 0$ and a sufficiently small $\lambda > 0$, as the number of samples increases the recursive formula (7) will generate a solution $\hat{r}$ such that $\Phi\hat{r} \to \Phi r^*$. In turn, this relies on reducing the simulation noise as illustrated by the graphs of Fig. 2. Moreover, notice that in realistic experimental conditions, physical noise and measurement precision are likely to result in $\|\eta\| \gg Tr(\Sigma_G) + Tr(\Sigma_c)$. In this case the covariance of the overall noise in (5) will be predominantly determined by that of η.

The results presented in Fig. 3 have been obtained after implementing Algorithm 2.2.2 with $t = 2 \times 10^3$, $K = 10^2$, $n = 10^6$, $s = 10^2$, ψ piecewise linear, and $\lambda = 10^{-7}$. The figure shows for comparison the true solution f^* used to compute the data b, its subspace projection Πf^* and subspace approximation $\Phi\hat{r}$ as computed by introducing $\hat{G}$, $\hat{c}$, Σ_G, and Σ_c into (8) after only a single iteration. The similarity between Πf^* and $\Phi\hat{r}$ is indicative of the small variance in the estimated $\hat{G}$ and $\hat{c}$. The total computation time, almost exclusively dissipated in estimating the upper triangular part of $\hat{G}$ (5,150 entries) and the 10^2 entries of c was about 8.5 h on a 2.66 GHz quad processor computer with 4 GB RAM running Matlab [16].

4 Special Case: Underdetermined Problems

Quite often, practical limitations impose a limit to the amount of data that can realistically be measured to estimate a certain set of parameters. By contrast, there is always a quest for increasing the amount of information extracted from an inverse solution, e.g. in terms of its degrees of freedom or resolution. This mismatch in the dimensions of the parameter and data spaces evidently yields underdetermined inverse problems. These problems are addressed in the context of the minimum-norm Backus-Gilbert regularization method [2].

In dealing with severely underdetermined problems, one can implement our algorithm to estimate the components of the high-dimensional solution directly, without the need for subspace approximation. Assuming now that $A \in \Re^{s \times n}$ where s is reasonably small and n is very large by comparison, we may adapt the preceding methodology to estimate f_t from

$$(A'ZA + \lambda I) f_t = A'Zb. \tag{17}$$

Using the matrix inversion lemma [14], the solution can also be expressed as

$$f_t = A'(AA' + \lambda Z^{-1})^{-1} b,$$

which by contrast to (17) requires only the inversion of a low-dimensional matrix. Using this lemma, it is also easy to prove that if $(A'ZA + \lambda I)$ is well conditioned then so is $(AA' + \lambda Z^{-1})$. In such a case the s-dimensional matrix we seek to estimate by simulation is $G = AA'$, whose element G_{lq} we express as a finite sum of functions

$$G_{lq} = \sum_{i,j=1}^{n} \nu(i,j) = \sum_{i,j=1}^{n} a_{li} a_{qj}.$$

To obtain an approximation to the optimal importance sampling distribution for ν we work similar to the algorithm described above, essentially dividing the sampling space $[1, n]$ into K disjoint intervals $\Theta_1, \ldots, \Theta_K$, where we take a number of arbitrary points I_{Θ_i} and interpolate d-degree polynomial functions in each Θ_i. After t samples, the importance sampling yields the estimator given by

$$\hat{G}_{lq} = \frac{1}{t+1} \sum_{p=0}^{t} \frac{a_{li_p} a_{qi_p}}{\xi(i_p, j_p)}, \qquad l, q = 1, \ldots, s,$$

and its t-sample variance $\mathrm{var}(\hat{G}_{lq})$. Consequently the ith element of the solution can be computed by

$$f_i = a_i'(\hat{G} + \lambda Q^{-1})^{-1} b, \tag{18}$$

where $a_i \in \Re^s$ is the ith column of A and Q is the noise covariance encompassing the additive noise and the simulation error. Notice that since we do not use subspace approximation, the approximation error is essentially zero.

5 Conclusions and Future Directions

In this paper, we have considered the approximate solution of linear inverse problems within a low-dimensional subspace spanned by a given set of basis functions. We have proposed a simulation-based regularized regression approach that involves importance sampling and low-dimensional computation, and that relies on designing sampling distributions customized to the model matrices and basis functions spanning the subspace. We have elaborated on a few approaches for designing near-optimal sampling distributions, which exploit the continuous structure of the underlying models. The performance of our method has been evaluated with a number of numerical tests using a classical inverse problem. The computation experiments demonstrate an adequate reduction of simulation error after a relatively small number of samples and an attendant improvement in quality of the obtained approximate solution.

A central characteristic of our methodology is the use of low-dimensional calculations in solving high-dimensional problems. Two important approximation issues arise within this context: first the solution of the problem should admit a reasonably accurate representation in terms of a relatively small number of basis functions, and second, the problem should possess a reasonably continuous/smooth structure so that effective importance sampling distributions can be designed with relatively small effort. In our computational experiments, simple piecewise polynomial approximations have proved adequate, but other more efficient alternatives may be possible. We finally note that the use of regularized regression based on a sample covariance obtained as a byproduct of the simulation was another critical element for the success of our methodology with nearly singular problems.

Acknowledgements Research is supported by the Cyprus Program at MIT Energy Initiative, the LANL Information of Science and Technology Institute, and by NSF Grant ECCS-0801549.

References

1. Asmussen, S., and Glynn, P. W.: Stochastic Simulation. Springer, New York (2007)
2. Bertero, M., and Boccacci, P.: Introduction to Inverse Problems in Imaging. IoP, Bristol (2002)
3. Bertsekas, D.P., and Tsitsiklis, J.: Neuro-Dynamic Programming. Athena Scientific (1996)

4. Bertsekas D.P., and Yu, H.: Projected Equation Methods for Approximate Solution of Large Linear Systems. J. Comp. Appl. Math., **227**, 27–50 (2009)
5. Biegler, L., Biros, G., Ghattas, O., Heinkenschloss, M., Keyes, D., Mallick, B., Marzouk, Y., Tenorio, L., van Blomen Waanders, B., and Willcox, K. (eds.): Large-Scale Inverse Problems and Quantification of Uncertainty. Wiley, Chichester (2011)
6. Curtiss, J. H.: Monte Carlo Methods for the Iteration of Linear Operators. J. Math. Phys., **32(4)**, 209–232 (1953)
7. Drineas, P., Kannan, R., and Mahoney, M.W.: Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication. SIAM J. Comput. **36**, 132–157 (2006)
8. Drineas, P., Kannan, R., and Mahoney M.W.: Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix. SIAM J. Comput. **36**, 158–183 (2006)
9. Drineas, P., Kannan, R., and Mahoney M.W.:, Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition. SIAM J. Comput. **36**, 184–206 (2006)
10. Forsythe, G.E., and Leibler, R.A.: Matrix Inversion by a Monte Carlo Method. Math. Tabl. Aids to Comp., **6(38)**, 78–81 (1950)
11. Groetsch, C.W.: The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind. Pitman, London (1984)
12. Halton, J.H.: A Retrospective and Prospective Survey of the Monte Carlo Method. SIAM Review, **12(1)** (1970)
13. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. SIAM, Philadelphia (2010)
14. Kaipio, J., and Somersalo, E.: Statistical and Computational Inverse Problems. Springer, New York (2004)
15. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling, Springer, New York (2009)
16. Matlab, The Mathworks Ltd
17. O'Leary, D.P.: Near-optimal Parameters for Tikhonov and Other Regularization Methods. SIAM J. on Scientific Computing, **23(4)**, 1161–1171 (2001)
18. Sutton, R.S., and Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Boston (1998)
19. Wang, M., Polydorides, N., and Bertsekas, D.P.: Approximate Simulation-Based Solution of Large-Scale Least Squares Problems, Report LIDS-P-2819, MIT (2009)

In Search for Good Chebyshev Lattices

Koen Poppe and Ronald Cools

Abstract Recently we introduced a new framework to describe some point sets used for multivariate integration and approximation (Cools and Poppe, BIT Numer Math 51:275–288, 2011), which we called Chebyshev lattices. The associated integration rules are equal weight rules, with corrections for the points on the boundary. In this text we detail the development of exhaustive search algorithms for *good* Chebyshev lattices where the cost of the rules, i.e., the number of points needed for a certain degree of exactness, is used as criterium. Almost loopless algorithms are considered to avoid dependencies on the rank of the Chebyshev lattice and the dimension. Also, several optimisations are applied: reduce the vast search space by exploiting symmetries, lower the cost of the point set creation and minimise the cost of the degree verification. The concluding summary of the search results indicates that higher rank rules in general are better and that the blending formulae due to Godzina lead to the best rules within the class of Chebyshev lattice rules: no better rules have been found in the searches conducted in up to five dimensions.

1 Introduction

1.1 Motivation

Recently Clenshaw–Curtis integration was revived [14, 15]. This technique for 1-dimensional integration is based on a cosine mapped trapezoidal rule. The question that motivated this research is: can a similar transform applied to a

K. Poppe · R. Cools
Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
e-mail: Koen.Poppe@cs.kuleuven.be; Ronald.Cools@cs.kuleuven.be

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_37, © Springer-Verlag Berlin Heidelberg 2012

lattice rule be beneficial for multivariate integration and approximation. In [3] we presented the Chebyshev lattice rules as generalising framework for cubature rules suited for integration with Chebyshev weight function. This setting arises naturally when determining the coefficients of a Chebyshev approximation of a multivariate function. Because of the limited number of parameters, the Chebyshev lattice notation enables an exhaustive search in moderate dimensions. We are interested in *good* cubature rules in the sense that they require a small number of points, and thus function evaluations, to give exact results for the weighted integral of polynomials up to a certain degree. We hope that, as known (near-) optimal point sets fit into the framework, good cubature rules in moderate dimensions may also be found.

1.2 Classical Lattices

Lattice rules are a well known family of quasi-Monte Carlo methods for the approximation of s-dimensional integrals (see, e.g., [2, 13] and their references). They are based on point sets that can easily be described as a linear combination of $k \leq s$ generating vectors. More specifically they are based on so-called integration lattices: a discrete subset of the real space $\mathbb{R}^s$ that is closed under addition and subtraction that contains the integer points as a subset. Hence the point set can be described as

$$\left\{ \sum_{j=1}^{k} \frac{\ell_j \, \mathbf{z}_j}{d_j} : \ell_j \in \mathbb{Z}, d_j \in \mathbb{N}_0 \text{ and } \mathbf{z}_j \in \mathbb{Z}^s \text{ for } j = 1, \ldots, k \right\}, \tag{1}$$

where we use curly braces only to denote the *set* of points, $\mathbb{Z}$ is the set of integers and $\mathbb{N}_0 := \{1, 2, \ldots\}$. The associated lattice rule uses a set of points $\Lambda = \{\mathbf{x}\} \subset [0, 1)^s$, where the components of $\mathbf{x}$ consist of the fractional part of the lattice points, to approximates the integral of $f(\mathbf{x})$ on the domain $[0, 1]^s$

$$Q[f] := \frac{1}{N} \sum_{\mathbf{x} \in \Lambda} f(\mathbf{x}) \quad \approx \quad \int_{[0,1]^s} f(\mathbf{x}) \, \mathsf{d}\mathbf{x} =: I[f]. \tag{2}$$

Note that N, the number of points in (2), follows from the generating vectors, more specific $N = |\det \mathbf{G}\,|^{-1}$ where $\mathbf{G}$ is a rational generator matrix of the lattice (1) [7].

There are many quality criteria for lattice rules in use, see [2] for a survey. One of these is the *total trigonometric degree*. This is defined as the maximal $n = |\mathbf{h}| := \sum_{r=1}^{s} |h_r|$ for which all trigonometric functions $f(\mathbf{x}) = \prod_{r=1}^{s} \exp(2\pi\imath\, h_r x_r)$ are integrated exactly, i.e., (2) becomes an equality. This allows for a ranking of rules: *good lattice rules* have a small number of points N compared to other rules with the same trigonometric degree and thus require fewer function evaluations.

1.3 Chebyshev Lattices

Chebyshev lattices of rank k, as introduced in [3], are based on a cosine mapping of a classical lattice with the same k generating vectors $\mathbf{z}_j \in \mathbb{Z}^s$ and denominators $d_j \in \mathbb{N}_0$, a fixed offset vector $\mathbf{z}_\Delta \in \mathbb{Z}^s$ and denominator $d_\Delta \in \mathbb{N}_0$:

$$\chi := \left\{ \cos\left(\pi\left(\sum_{j=1}^{k} \frac{\ell_j\, \mathbf{z}_j}{d_j} + \frac{\mathbf{z}_\Delta}{d_\Delta}\right)\right) : \ell_j \in \mathbb{Z} \text{ for } j = 1, \ldots, k \right\}. \tag{3}$$

The point sets (3) were developed in the context of multivariate integration on hypercubes $C_s := [-1,1]^s$ using a Chebyshev approximation of the integrand. Evaluating the coefficients of the approximation leads to integrals with Chebyshev weight function $\omega(\mathbf{x}) := \pi^{-s} \prod_{r=1}^{s} (1-x_r^2)^{-\frac{1}{2}}$ that, due to hyperinterpolation theory [12], can be replaced with a suitable cubature rule in which we use the points from (3):

$$Q[f] := \sum_{\mathbf{x}\in\chi} w_{\mathbf{x}}\, f(\mathbf{x}) \quad \approx \quad \int_{C_s} f(\mathbf{x})\, \omega(\mathbf{x})\, \mathsf{d}\mathbf{x} =: I[f]. \tag{4}$$

To avoid a periodicity requirement, and inherently through the cosine mapping, χ possibly includes points on all boundaries. The cubature rule (4) is therefore an equal weight rule, with corrections for the points on the boundary. The weights are still known explicitly, i.e., it is not necessary to solve a system of equations to obtain their values. To correct for points on the boundary, a scaling of the weights is needed: in three dimensions, points on faces, edges and vertices will have weights proportional to $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$. Using the conditional function $\phi(\text{condition})$, which evaluates to 1 if the 'condition' is true, 0 otherwise, the weight $w_{\mathbf{x}}$ can be written as

$$w_{\mathbf{x}} := \frac{\tilde{w}_{\mathbf{x}}}{\tilde{W}}, \quad \text{where} \quad \tilde{w}_{\mathbf{x}} := 2^{-\sum_{r=1}^{s} \phi(|x_r|=1)} \quad \text{and} \quad \tilde{W} := \sum_{\mathbf{x}\in\chi} \tilde{w}_{\mathbf{x}}, \tag{5}$$

in which the normalisation factor $\tilde{W}$ ensures exactness of (4) for a constant function.

As with classical lattices, the quality of a cubature rule (4) can be expressed in terms of its degree, but in this weighted setting we use the *total algebraic degree*. If (4) is an equality for all polynomial functions $f(\mathbf{x}) = \prod_{r=1}^{s} x_r^{h_r}$, where $|\mathbf{h}| \leq n$, the cubature rule is said to have a degree n. Similar to classical lattice rules, *better* Chebyshev lattice rules require less points to attain a certain degree n.

It is important to note that, in contrast to the classical lattice rules, there is no closed relation between the parameters of the Chebyshev lattice and the number of points N. Also, due to the folding of the cosine mapping, it is not guaranteed that *good* classical lattice parameters lead to *good* Chebyshev lattice rules and vice versa.

We showed [3] that most cubature point sets for the integrals (4) with Chebyshev weight function can be written as a Chebyshev lattice rule. Godzina's *blending point set* [5], for example, an explicit s-dimensional point set with given degree n, leads to the following full rank ($k = s$) Chebyshev lattice rule for $n = 4\nu - 1$ ($\nu \in \mathbb{N}_0$):

$$\begin{aligned} \mathbf{z}_1 &= [\,1, 1, 1 \cdots 1, 1\,] \\ \mathbf{z}_2 &= [\,0, 2, 0 \cdots 0, 0\,] \\ &\;\vdots \qquad\quad \ddots \\ \mathbf{z}_s &= [\,0, 0, 0 \cdots 0, 2\,] \end{aligned} \qquad \text{with} \qquad \begin{cases} d_1, \ldots, d_s = d_\Delta = 2\nu, \\ \mathbf{z}_\Delta = [\,0, 1, 0, 1, \ldots]. \end{cases}$$

1.4 Computer Search

It is obvious that Chebyshev lattice rules are described by a fixed number of parameters. This search space depends on the rank k and the dimension s and is bounded: due to the periodicity of the cosine mapping, the components of the generating vectors can be reduced modulo their denominators. Without loss of generality, we consider all denominators equal, i.e., $d_1 = \cdots = d_k = d$ and this common denominator d can be seen as the third search parameter. Note that any given Chebyshev lattice can be reformulated with equal denominators by taking d equal to the least common multiple of all denominators $d_1, \ldots, d_k$. This simplifies bookkeeping but postpones specific rules to higher values of d and thus more expensive searches. For example, a rank-2 Chebyshev lattice with $d_1 = 5$ and $d_2 = 7$ will only be found when $d = 35$.

Each combination of parameters s, k and d leads to a different search space. Our aim is to find the best Chebyshev lattices, i.e., the rules that require the lowest number of points to attain a certain degree of accuracy. Therefore, the programs keep track of the best Chebyshev lattice parameters, i.e., the generating vectors, denominator and the number of points for each degree and replaces them only if a rule with less points is found. These search process concepts are illustrated in Listing 1.

It is obvious that the search from Listing 1 can be done in parallel for the number of dimensions s, the rank k and the denominator d. This will require some post-processing, to combine all the rules and find the 'globally' best ones, but can be solved easily by storing the rules into a small database and by performing simple queries to extract the best Chebyshev lattice rules.

Whenever possible, the search programs have been made invariant of the three search parameters. To illustrate this, consider the most rudimentary way to iterate over rank-1 generators in s dimensions using s nested loops, one for each component of the generating vector. This is straightforward but limits the applicability of the program to a fixed s. To avoid this dependency, we have been exploring *loopless* and *almost loopless* algorithms [4, 6] that produce the same result with only one or

Listing 1 Conceptual overview of the search for good Chebyshev lattice rules. Sects. 2–4 elaborate on the lines that are indicated in the algorithmic overview.

```
for $s = 2, 3, \ldots$
  catalog = [ ]
  for $k = 1, \ldots, s$
    for $d = 1, 2, \ldots$ until time budget for this $s$ exhausted
      for each $\{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ in the search space        ⤳ Section 2
        create the point set; let $N$ be the number of points         ⤳ Section 3
        determine the algebraic total degree $n$                      ⤳ Section 4
        if( is_empty(catalog[$n$]) || $N$ < catalog[$n$].$N$ ) then
          catalog[$n$] = $\{\mathbf{z}_1, \ldots, \mathbf{z}_k, d, N\}$
        end if
      end for
    end for
  end for
  report the rules from catalog
end for
```

two nested loops, independent of the number of dimensions. This way, one could rewrite the rank-1 search so that the number of dimensions is just an input parameter: no code must be changed to run the search for another number of dimensions. This is clearly less error prone than explicit loops and could arguably be even slightly faster.

In the following three sections, we provide more details on the actual search programs and optimisations that have been used in the highlighted steps from Listing 1. After that, actual search results in up to five dimensions are compared to known point sets. Section 6 concludes this paper.

2 Reducing the Search Space

2.1 Exploiting Symmetry

The search space for generating vectors of s-dimensional, rank-k Chebyshev lattices where the components of each j-th generating vector component belong to $\{0, 1, \ldots, d\}$, has $(d+1)^{ks}$ elements. However, symmetries can be exploited to reduce this huge search space. Due to the symmetry of C_s and ω, the components of a point set can be permuted without influencing the degree. The same reasoning allows us to reorder the components of the generating vectors, which of course is not limited to rank-1. The higher rank case, i.e., $k > 1$, can also exploit the invariance of the Chebyshev lattice with respect to the ordering of the k generating vectors.

Let us first focus on s-dimensional rank-1 Chebyshev lattices: consider the integer search space where all generating vectors $\mathbf{z}_1$ live in. The number of symmetries can be related to the number of distinct component values of this vector, here denoted by t. The most obvious example are the $d+1$ vectors that have only one independent component, thus $t = 1$. This case can be summarised with a parametric description $\mathbf{z}_1 = (\mathtt{A}, \dots, \mathtt{A})$. For $t > 1$, several parameter descriptions can be made, e.g., for $s = 3$ and $t = 2$ there is $\mathbf{z}_1 = (\mathtt{A}, \mathtt{A}, \mathtt{B})$ and $\mathbf{z}_1 = (\mathtt{A}, \mathtt{B}, \mathtt{B})$ in which the parameters $\mathtt{A} < \mathtt{B}$ It should be clear that if all vectors satisfying these descriptions are checked, it is no longer necessary to verify $(\mathtt{A}, \mathtt{B}, \mathtt{A})$, $(\mathtt{B}, \mathtt{A}, \mathtt{A})$, $(\mathtt{B}, \mathtt{B}, \mathtt{A})$ or $(\mathtt{B}, \mathtt{A}, \mathtt{B})$ because of the cubature rules degree's invariance with respect to coordinate permutations. This leads to an asymptotical reduction of $\frac{1}{s!}$ for $d \to \infty$, but note that the reduced space, with $\frac{1}{s!}(d + 1)^{ks}$ elements, still explodes for growing s, k or d.

We define the *unique* generating vectors as the sets of generating vectors that are in the reduced search space in which all of the above invariances have been excluded. To see how many of those generating vectors there are, we first observe that the entire search space in s dimensions can be decomposed as

$$(d + 1)^s = \sum_{t=1}^{s} \left(\underbrace{\binom{d+1}{t}}_{\mathrm{I}} \cdot \underbrace{\sum_{\mathbf{g}} \frac{s!}{\prod_{v=1}^{t} g_v!}}_{\mathrm{II}} \right). \tag{6}$$

In this, the I-part represents the number of generators one can construct with t-different parameters in $\{0, \dots, d\}$ and the II-part denotes the number of symmetries that should also be included to get the entire space (this is also the number of permutations of the multi-set with occurrence counts $\mathbf{g}$). Examples of the vectors $\mathbf{g}$ are given in Table 1 and a proof of (6) for $s \leq 4$ is given in the Appendix.

Using this formalism, the size of the search space for rank-1 rules can be written explicitly by taking only one symmetry setting per parameter description. This replaces the II-part in (6) by #$\mathbf{g}$, the number of $\mathbf{g}$ vectors, and leads to (see Appendix)

$$U_{s,d,k=1} = \sum_{t=1}^{s} \left(\binom{d+1}{t} \cdot \#\mathbf{g} \right) = \sum_{t=1}^{s} \left(\binom{d+1}{t} \cdot \binom{s-1}{s-t} \right). \tag{7}$$

For higher ranks, a similar approach can be followed, but note that coordinate changes need to be considered for all k generating vectors together. A similar parametric approach as above can be used to describe the vectors. In two dimensions, rank $k = 2$, this leads to the following sets of vectors where $\mathtt{A} < \mathtt{B} < \mathtt{C} < \mathtt{D}$:

$$\mathbf{Z} = \left[\frac{\mathbf{z}_1}{\mathbf{z}_2} \right] = \left[\frac{\mathtt{A}, \mathtt{A}}{\mathtt{A}, \mathtt{B}} \right], \left[\frac{\mathtt{A}, \mathtt{B}}{\mathtt{A}, \mathtt{C}} \right], \left[\frac{\mathtt{A}, \mathtt{B}}{\mathtt{B}, \mathtt{A}} \right], \left[\frac{\mathtt{A}, \mathtt{B}}{\mathtt{C}, \mathtt{A}} \right], \dots, \left[\frac{\mathtt{A}, \mathtt{B}}{\mathtt{C}, \mathtt{D}} \right]. \tag{8}$$

The second symmetry for higher ranks is the order of the k vectors. Obviously, permuting them does not change the Chebyshev lattice. Also, because the search focusses on a specific rank, sets of vectors with a linear dependency can be discarded.

Table 1 Examples of the descriptions of a s-dimensional grid with components in $\{0, \dots, d\}$ with their occurrence vector $\mathbf{g}$. The last column lists the specific values of part II in (6).

(a) $s = 1$

t	Descr.	$\mathbf{g}$	$\frac{s!}{\prod_{v=1}^{t} g_v!}$	$\sum_{\mathbf{g}} \frac{s!}{\prod_{v=1}^{t} g_v!}$
1	A	[1]	1	1

(b) $s = 2$

t	Descr.	$\mathbf{g}$	$\frac{s!}{\prod_{v=1}^{t} g_v!}$	$\sum_{\mathbf{g}} \frac{s!}{\prod_{v=1}^{t} g_v!}$
1	A,A	[2]	1	1
2	A,B	[1, 1]	2	2

(c) $s = 3$

t	Descr.	$\mathbf{g}$	$\frac{s!}{\prod_{v=1}^{t} g_v!}$	$\sum_{\mathbf{g}} \frac{s!}{\prod_{v=1}^{t} g_v!}$
1	A,A,A	[3]	1	1
2	A,A,B	[2, 1]	3	6
	A,B,B	[1, 2]	3	
3	A,B,C	[1, 1, 1]	6	6

(d) $s = 4$

t	Descr.	$\mathbf{g}$	$\frac{s!}{\prod_{v=1}^{t} g_v!}$	$\sum_{\mathbf{g}} \frac{s!}{\prod_{v=1}^{t} g_v!}$
1	A,A,A,A	[4]	1	1
2	A,A,A,B	[3, 1]	4	14
	A,A,B,B	[2, 2]	6	
	A,B,B,B	[1, 3]	4	
3	A,A,B,C	[2, 1, 1]	12	36
	A,B,B,C	[1, 2, 1]	12	
	A,B,C,C	[1, 1, 2]	12	
4	A,B,C,D	[1, 1, 1, 1]	24	24

The parametric representation of the generators provides us with a flexible and memory friendly way of avoiding symmetries without explicitly storing all visited generators. Creating the parametric descriptions requires only a moderate $\mathcal{O}\big((ks)^{ks}\big)$ memory locations. It is important to see that this number is independent of d: the descriptions can thus be calculated once and re-used for every value of the denominator (skipping descriptions where $t > k\,(d + 1)$).

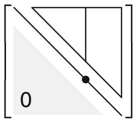
Our current programs use an almost loopless algorithm to generate the prescriptions for arbitrary rank k and dimension s and stores them on disk. The actual search then reads these descriptions one by one and uses a second almost loopless algorithm to generate the values for the parameters A, B, . . . Note that the loopless property dramatically simplifies this process, because the number of parameters t

is now just an input argument of the algorithm: t may vary between descriptions without adding any complexity to the program.

2.2 Hermite Normal Form

Following the discussions in ([13], Chap. 9), another way of avoiding spurious elements in the search space is by using the Hermite normal form. It is shown that integration lattices yield a full-rank matrix $\mathbf{H}$ which is derived from a generator matrix $\mathbf{Z} \in \mathbb{Z}^{s \times s}$ with on its rows the k generating vectors, extended with $s - k$ suitable unit vectors, using a unimodular transformation $\mathbf{U}$ so that $\mathbf{Z} = \mathbf{U}\,\mathbf{H}$. These Hermite matrices have a specific structure which can be exploited during the search process:

$$\mathbf{H} = \left[H_{i,j}\right] =$$



$$\text{where} \begin{cases} H_{i,j} = 0, & j < i, \\ 0 < H_{i,j} \le d, & j = i, \\ 0 \le H_{i,j} < H_{j,j}, & j > i. \end{cases} \qquad (9)$$

To enumerate all $\mathbf{H}$ matrices, only $\frac{s(s+1)}{2}$ parameters are needed instead of s^2 when considering all generator matrices $\mathbf{Z}$. Using the specific structure of $\mathbf{H}$, the number of elements in this search space is thus only

$$U_{s,d} = \prod_{r=1}^{s} \sum_{\delta=0}^{d} \delta^{r-1}. \qquad (10)$$

We have implemented a routine that iterates over all Hermite normal form matrices $\mathbf{H}$, given s and a denominator d. For this search, we loosened the fixed rank requirement and checked all sets of generating vectors *up to* a certain rank-k because this determines the number of parameters in $\mathbf{H}$.

3 Reducing the Point Set Creation Cost

When creating a Chebyshev lattice point set, duplicates must be avoided. This is somehow expensive because for a rank-k Chebyshev lattice with common denominator d, $\bar{N} = (d+1)^k$ s-dimensional points must be verified to be distinct. Using a balanced tree structure, a red-black tree in our programs, the number of scalar comparisons in this operation can be reduced from $\mathcal{O}(s\,\bar{N}^2)$ to $\mathcal{O}(s\,\bar{N}\log\bar{N})$, but further improvements are possible. With the common denominator d, it is easy to see that points from the Chebyshev lattice are derived from an integer vector $\mathbf{y}_\ell$. Rewriting the points from (3) as

$$\mathbf{x}_\ell = \cos\left(\frac{\pi}{d}\,\mathbf{y}_\ell\right) \qquad \text{with} \qquad \mathbf{y}_\ell := \sum_{j=1}^{k} \ell_j \mathbf{z}_j + \mathbf{z}_\Delta \tag{11}$$

clearly shows this. Moreover, due to the periodicity of the cosine, this integer vector $\mathbf{y}_\ell$ can be reduced, component by component, so that it still produces the same point $\mathbf{x}_\ell$. Using $\tilde{\mathbf{y}}_\ell$ for this reduced integer vector and 'minmod$_d$' for the element-wise reduction so that $\tilde{\mathbf{y}}_\ell \in [0, d]^s$, we can write the points as

$$\mathbf{x}_\ell = \cos\left(\frac{\pi}{d}\,\mathbf{y}_\ell\right) =: \cos\left(\frac{\pi}{d}\,\tilde{\mathbf{y}}_\ell\right) \quad \text{with} \quad \tilde{\mathbf{y}}_\ell := \operatorname{minmod}_d(\mathbf{y}_\ell)\,. \tag{12}$$

It suffices to compare s-dimensional integer vectors $\tilde{\mathbf{y}}_\ell$, instead of floating point vectors, to see whether a given point is already in the set. And, since d is rather small (less than a few hundred), it might be possible to compress the vectors even more. One way is to use integers with smaller ranges to represent $\tilde{\mathbf{y}}_\ell$ but combining the component values using a multi-radix notation proved to be even quicker. With sufficiently large integers, $\tilde{\mathbf{y}}_\ell$ can be stored as a scalar

$$\gamma_\ell = \sum_{r=1}^{s} \tilde{\mathbf{y}}_{\ell,r} (d+1)^r, \tag{13}$$

which is significantly faster in comparisons than the loop that is needed to go over all the components. Note that the tree also requires less memory as it only has to store γ_ℓ values to determine whether a point is a duplicate or not.

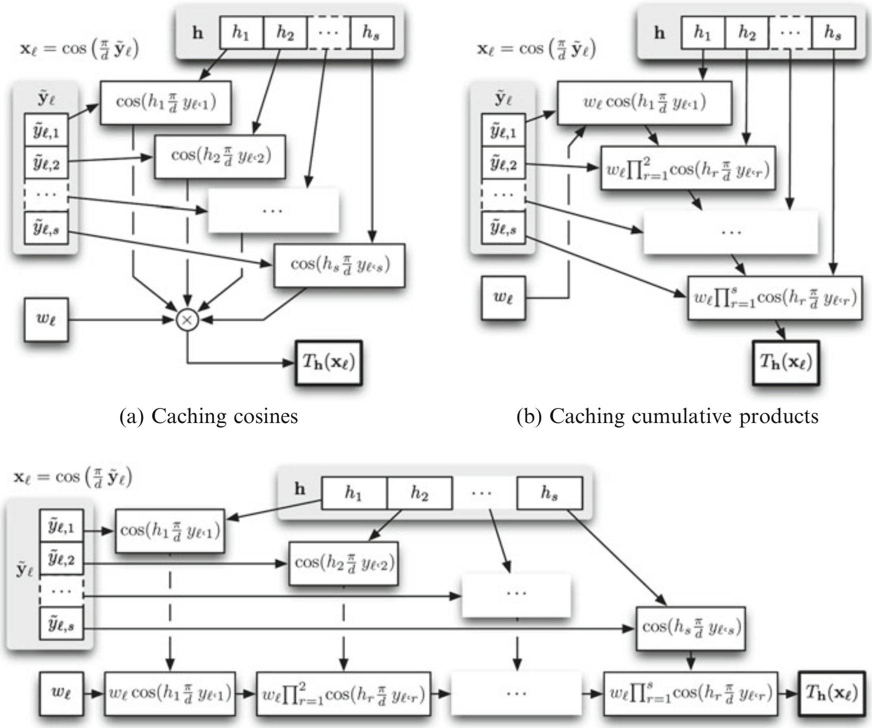
4 Reducing the Degree Verification Cost

4.1 Evaluation of the Basis Function

In order to check the degree of a given point set and weights, the cubature rule (4) is evaluated for polynomials with increasing degree. It seems that, without prior knowledge about the point set, not much can be done to accelerate this step: all polynomials for increasing degrees must be verified. However, using Chebyshev polynomials $T_{\mathbf{h}}(\mathbf{x}) := \prod_{r=1}^{s} \cos\left(h_r \arccos(x_r)\right)$ simplifies the evaluation: it is obvious that $\arccos(x_r)$ can be precomputed, but, because all points will originate from a Chebyshev lattice, using $\tilde{\mathbf{y}}_\ell$ from (12) leads to

$$T_{\mathbf{h}}(\mathbf{x}_\ell) = \prod_{r=1}^{s} \cos\left(h_r \arccos\left(\cos\left(\frac{\pi}{d}\, y_{\ell,r}\right)\right)\right) = \prod_{r=1}^{s} \cos\left(h_r \frac{\pi}{d}\, \tilde{y}_{\ell,r}\right) \tag{14}$$

and so the evaluation of the arccosine of the cosine can be avoided.


$\mathbf{x}_\ell = \cos\left(\frac{\pi}{d}\tilde{\mathbf{y}}_\ell\right)$
$\mathbf{h}$
h_1
h_2
$\cdots$
h_s
$\tilde{\mathbf{y}}_\ell$
$\tilde{y}_{\ell,1}$
$\tilde{y}_{\ell,2}$
$\cdots$
$\tilde{y}_{\ell,s}$
$\cos(h_1 \frac{\pi}{d} y_{\ell,1})$
$\cos(h_2 \frac{\pi}{d} y_{\ell,2})$
$\cos(h_s \frac{\pi}{d} y_{\ell,s})$
w_ℓ
$T_\mathbf{h}(\mathbf{x}_\ell)$
$w_\ell \cos(h_1 \frac{\pi}{d} y_{\ell,1})$
$w_\ell \prod_{r=1}^{2} \cos(h_r \frac{\pi}{d} y_{\ell,r})$
$w_\ell \prod_{r=1}^{s} \cos(h_r \frac{\pi}{d} y_{\ell,r})$

(a) Caching cosines

(b) Caching cumulative products

(c) Caching both cosines and cumulative products

Fig. 1 Three ways of reducing the evaluation complexity of $T_\mathbf{h}(\mathbf{x}_\ell)$ in (14). For the order in which the **h**'s are generated, variant (**b**) is the fastest, followed by (**a**). Variant (**c**) is slightly slower than (**b**) but requires more bookkeeping. For clarity, the diagrams show the evaluation in only one point. Evidently, this is vectorised over all points to benefit from pipelining.

For each degree n, all polynomials $T_\mathbf{h}$ with $|\mathbf{h}| = n$ must be verified, but *the order* in which this is done does not matter. If explicit loops are used, $\cos(h_r \frac{\pi}{d}\, \tilde{y}_{\ell,r})$ will only change when h_r changes. By storing these product terms for each h_r (see Fig. 1a), the computational complexity, in terms of the number of cosine evaluations, is $\mathscr{O}\big(\sum_{r=1}^{s} \binom{n}{r} N\big)$ for the evaluation for N points, instead of the $\mathscr{O}\big(s\binom{n}{s}N\big)$ a naive implementation would require.

Alternatively, the number of floating point operations can be reduced by storing cumulative products of the cosines, as illustrated in Fig. 1b. For example, if we know that only h_s changes, the repeated calculations of the cumulative product $\prod_{r=1}^{s-1} \cos(h_r \frac{\pi}{d}\, \tilde{y}_{\ell,r})$ from (14) can be cached easily.

These two ideas can also be combined, leading to the diagram in Fig. 1c. However, experiments for moderate dimensions ($s \le 10$) have shown that this third variant does not provide an additional decrease in execution time. Caching the cumulative product from Fig. 1b provides the fastest results and can be seen to be more cache friendly because of memory locality.

4.2 Generation of the Coefficients

This complexity can be reduced even further using an algorithm that generates the coefficients **h** in a way that minimises component changes because this increases the efficiency of the caches mentioned before. Of course, explicit loops already do this but as explained before we have been exploring almost loopless algorithms to avoid the dependency on the number of dimensions.

Our almost loopless algorithm, based on [4], generates all **h**'s for a certain degree n using only two component changes per **h**. This is optimal: less changes would violate the fixed degree assumption.

4.3 Special Note on $n = 1$

As exactness for a constant ($n = 0$) is guaranteed by the scaling of the weights, the first degree that must be checked is $n = 1$. Research has shown that a significant part of the generators fail for this degree. Using the fact that all s coefficients **h** with $|\mathbf{h}| = 1$ have only one non-zero element, this degree can be verified using only sN cosines in total (N is the number of points in the Chebyshev lattice) compared to s^2N cosines if these zero elements are not exploited. Unrolling for $n = 1$ speeds up some searches with up to a factor 3.

4.4 Incorporating Knowledge of the Point Set

Orthogonal to the above, another way to improve the complexity of the degree verification is the use of symmetry information. If it is known in advance that a point set equals its reflection around any $x_r = 0$, only even degrees must be checked in that direction (odd degrees are guaranteed). This proves very effective in reducing the number of polynomials in the verification, but for now symmetry information is not available a such. As the generating vectors are known, they do provide information about the symmetry. However, due to the folding-like operations of the element-wise cosine transform in the Chebyshev lattices, it is unclear how to derive this kind of symmetry properties directly.

5 Search Results

All results from the different search programs are summarised in Table 2. This shows that the best point sets do not improve the results of Godzina given at the end of Sect. 1.3. Therefore we measure the cost for the rules, i.e., the number of points, relative to those of Godzina as can be seen in Fig. 2. In up to four dimensions we

Table 2 Generating vectors corresponding to the Chebyshev lattices from Fig. 2 without offset vector ($\mathbf{z}_\Delta = \mathbf{0}$). Note that the results for $s = 2$ are excluded here, because they correspond to the Padua ($k = 1$) and Morrow-Patterson ($k = 2$) points described in [3]. The same applies for $k = s$ where the results correspond to Godzina's point set. In five dimensions, only rank-1 Chebyshev lattices and one rank-2 have been found so far.

(a) $s = 3, k = 1$

n	$\mathbf{z}_1$			d	N
1	1	1	1	1	2
3	6	10	15	30	18
5	12	15	20	60	30
7	20	28	35	140	60
9	30	35	42	210	84
11	42	66	77	462	168
13	56	63	72	504	180
15	72	88	99	792	270
17	90	99	110	990	330
19	110	130	143	1,430	462
21	132	143	156	1,716	546
23	156	204	221	2,652	819
25	182	195	210	2,730	840
27	210	238	255	3,570	1,080
29	240	255	272	4,080	1,224
31	272	304	323	5,168	1,530
33	306	323	342	5,814	1,710

(b) $s = 3, k = 2$

n	$\mathbf{z}_1$			$\mathbf{z}_2$			d	N
1	0	0	2	2	2	1	2	3
3	0	4	6	3	6	3	6	10
5	3	4	4	0	0	8	12	20
7	4	5	5	0	0	10	20	39
9	6	25	6	30	25	6	30	63
11	7	18	7	21	18	7	42	100

(c) $s = 4, k = 1$

n	$\mathbf{z}_1$				d	N
1	1	1	1	1	1	2
3	30	42	70	105	210	72
5	60	84	105	140	420	120
7	140	180	252	315	1,260	300
9	210	330	385	462	2,310	504
11	462	546	858	1,001	6,006	1,176
13	504	616	693	792	5,544	1,080

(d) $s = 4, k = 2$

n	$\mathbf{z}_1$				$\mathbf{z}_2$				d	N
1	0	0	0	2	2	2	2	1	2	3
3	0	6	4	0	3	3	6	2	6	20
5	3	4	4	9	3	12	4	3	12	52
7	4	5	5	12	4	15	5	4	20	117

(e) $s = 4, k = 3$

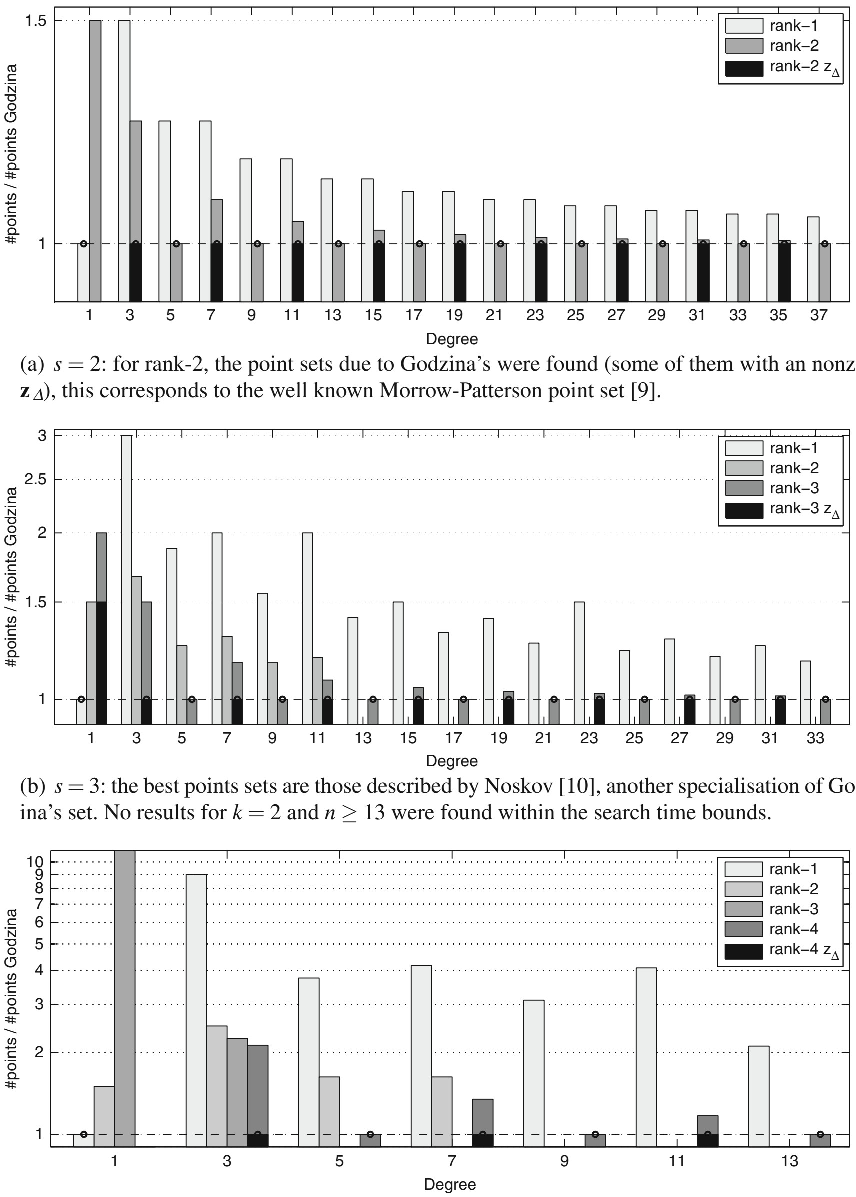
n	$\mathbf{z}_1$				$\mathbf{z}_2$				$\mathbf{z}_3$				d	N
1	1	0	0	1	0	1	0	0	0	0	1	0	2	27
3	3	3	3	2	0	6	0	4	0	0	6	4	6	18

(f) $s = 5, k = 1$

n	$\mathbf{z}_1$					d	N
1	1	1	1	1	1	1	2
3	210	330	462	770	1, 155	2, 310	432
5	420	660	924	1, 155	1, 540	4, 620	720
7	1, 260	1, 540	1, 980	2, 772	3, 465	13, 860	1, 800
9	2, 520	3, 080	3, 465	3, 960	5, 544	27, 720	3, 240
11	6, 006	7, 854	9, 282	14, 586	17, 017	102, 102	10, 584
13	5, 544	6, 552	8, 008	9, 009	10, 296	72, 072	7, 560

(g) $s = 5, k = 2$

n	$\mathbf{z}_1$					$\mathbf{z}_2$					d	N
1	0	0	0	0	2	2	2	2	2	1	2	3



(a) $s = 2$: for rank-2, the point sets due to Godzina's were found (some of them with an nonzero $\mathbf{z}_\Delta$), this corresponds to the well known Morrow-Patterson point set [9].

(b) $s = 3$: the best points sets are those described by Noskov [10], another specialisation of Godzina's set. No results for $k = 2$ and $n \geq 13$ were found within the search time bounds.

(c) $s = 4$: as with three dimensions, rank-k rules where $1 < k < s$ appear harder to find. Therefore, several degrees do not have a rank-2 and/or rank-3 rule. Godzina's set is also found here.

Fig. 2 Search results for up to four dimensions. As Godzina's point set describes the best Chebyshev lattices found so far, we show the relative number of points relative to the number in Godzina's set. The graphs show the number of points as function of the degree for different ranks. The *black bars* on *top* of the full-rank rules indicate results for non-zero offset vectors $\mathbf{z}_\Delta$. A *circle* indicates that the specific rule found, is identical to a point set due to Godzina.

have found point sets equivalent to those due to Godzina, those cases are indicated with a circle on the graphs. Note that although Godzina's point set is generally described with a full rank Chebyshev lattice (i.e., $k = s$), for a degree $n = 1$, the denominator $d = 2$ and thus the rank of the rule drops to 1.

Although we searched for Chebyshev lattices without offset, i.e., $\mathbf{z}_\Delta = \mathbf{0}$, Fig. 2 also includes results for the shifted Chebyshev lattices that were found by adding shifts to the previously known rules. Often, by adding a shift, the number of points could be reduced a little, while retaining the degree.

In two dimensions, the Morrow-Patterson points [9] are known to be *optimal*: they achieve the theoretical lower bound for the number of points due to Möller [8]. As shown in [3], these optimal set are a specialisation of Godzina's point set. The best two-dimensional rank-1 rules in correspond to the Padua points [1]. They are non-optimal and require more points to achieve the same degree as those by Morrow-Patterson.

For $s = 3$, the point set described by Noskov [10], again instances of Godzina's rules, are found to be best. It also appears that good rank-2 rules are rather hard to find: within the self imposed bounds on the search time, for corresponding denominators, both good rank-1 and good rank-3 rules were found. The rank-2 rules however, seem to require larger denominators and thus more time than allowed. From the incomplete results of Fig. 2b, a first general observation can be made: except for $n = 1$, higher rank rules require less points for the same degree. This actually encourages us to pursue high rank searches, although this requires iterating in a much larger space.

Results in four dimensions (see Fig. 2c) and preliminary ones for $s = 5$ (only shown in Table 2) support the observation that higher ranks require less points. Investigation of the actual point sets also shows a favour for grid-like structures. This is rather unwanted, because it corresponds to quickly growing number of points when the degree and the number of dimensions increases. Note however that this growth is intrinsic for the definition of the degree we have chosen here, so it cannot be eliminated completely.

6 Conclusion

We have presented several approaches to search for good Chebyshev lattices and detailed the implementation and optimisation using, amongst other, (almost) loopless algorithms and caching structures. Our searches provide computational evidence that Godzina's point set is 'optimal' within all Chebyshev lattices: no better point sets were found so far. They are of full rank and require less points than lower rank results, which also appears to be a general observation in this context. An advantage of these not so commonly known blending point sets and rank-1 Chebyshev lattices is the ability to use the fast Fourier transform while creating Chebyshev approximations of a function. This might lead to efficient software for interpolation and integration in moderate dimensions [11].

With these results, we will not pursue the search for good Chebyshev lattices, although future work could include worst-case-error based criteria and possibly other definitions of the polynomial degree. Such quality criteria are required anyway if one wants to extend the applicability to higher dimensions.

Acknowledgements 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 authors.

The authors acknowledge the support for this project by the Bijzonder Onderzoeksfonds of the Katholieke Universiteit Leuven.

Appendix: Proof of Identity (6) and (7)

The sum of a general function $G(\mathbf{g})$ over all $\mathbf{g}$'s is equivalent to $t-1$ nested sums

$$\sum_{\mathbf{g}} G(\mathbf{g}) \equiv \sum_{g_1=1}^{s-(t-1)} \sum_{g_2=1}^{\substack{s-(t-2)\\-g_1}} \sum_{g_3=1}^{\substack{s-(t-3)\\-g_1-g_2}} \cdots \sum_{g_{t-1}=1}^{\substack{s-1\\-\sum_{\mu=1}^{t-2} g_\mu}} G(\mathbf{g})\,\Big|_{g_t=s-\sum_{\mu=1}^{t-1} g_\mu}. \tag{15}$$

Therefore, part II from identity (6) can be rewritten into

$$\sum_{\mathbf{g}} \frac{s!}{\prod_{\nu=1}^{t} g_\nu!} = \sum_{g_1=1}^{s-(t-1)} \frac{1}{g_1!} \sum_{g_2=1}^{\substack{s-(t-2)\\-g_1}} \frac{1}{g_2!} \sum_{g_3=1}^{\substack{s-(t-3)\\-g_1-g_2}} \frac{1}{g_3!} \cdots \sum_{g_{t-1}=1}^{\substack{s-1\\-\sum_{\mu=1}^{t-2} g_\mu}} \frac{s!}{g_{t-1}!\left(s-\sum_{\mu=1}^{t-1} g_\mu\right)!}.$$

We verified (6) computationally for $s \in [1,20]$ and $d \in [1,100]$. The proofs for $s \in [2,4]$ ($s = 1$ is trivial), as used in this paper, are straightforward. The bold numbers correspond to part II of (6) and are taken from Table 1. Hence we obtain:

$$\begin{aligned}
\boxed{s=2}\quad & \tbinom{d+1}{1}\mathbf{1} + \tbinom{d+1}{2}\mathbf{2} \\
&= (d+1) + d(d+1) \\
&= d^2 + 2d + 1 = (d+1)^2,
\end{aligned}$$

$$\begin{aligned}
\boxed{s=3}\quad & \tbinom{d+1}{1}\mathbf{1} + \tbinom{d+1}{2}\mathbf{6} + \tbinom{d+1}{3}\mathbf{6} \\
&= (d+1) + 3d(d+1) + (d-1)d(d+1) \\
&= d^3 + 3d^2 + 3d + 1 = (d+1)^3,
\end{aligned}$$

$$\begin{aligned}
\boxed{s=4}\quad & \tbinom{d+1}{1}\mathbf{1} + \tbinom{d+1}{2}\mathbf{14} + \tbinom{d+1}{3}\mathbf{36} + \tbinom{d+1}{4}\mathbf{24} \\
&= (d+1) + 7d(d+1) + 6(d-1)d(d+1) + (d-2)(d-1)d(d+1) \\
&= d^4 + 4d^3 + 6d^2 + 4d + 1 = (d+1)^4.
\end{aligned}$$

Formula (7) can be proven easily for general s. Counting the $\mathbf{g}$'s corresponds to substituting $G(\mathbf{g}) \equiv 1$ in (15):

$$\sum_{\mathbf{g}} 1 = \sum_{g_1=1}^{s-(t-1)} \sum_{g_2=1}^{\substack{s-(t-2)\\-g_1}} \sum_{g_3=1}^{\substack{s-(t-3)\\-g_1-g_2}} \cdots \sum_{g_{t-1}=1}^{\substack{s-1\\-\sum_{\mu=1}^{t-2} g_\mu}} 1 = \prod_{\nu=1}^{t-1} \frac{s-(t-\nu)}{\nu} = \binom{s-1}{t-1}.$$

References

1. Caliari, M., De Marchi, S., Vianello, M.: Bivariate polynomial interpolation on the square at new nodal sets. Applied Mathematics and Computation **165**(2), 261–274 (2005)
2. Cools, R., Nuyens, D.: A Belgian view on lattice rules. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 3–21. Springer (2008)
3. Cools, R., Poppe, K.: Chebyshev lattices, a unifying framework for cubature with the Chebyshev weight function. BIT Numerical Mathematics **51**, 275–288 (2011)
4. Ehrlich, G.: Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. Journal of the ACM **20**(3), 500–513 (1973)
5. Godzina, G.: Dreidimensionale Kubaturformeln für zentralsymmetrische Integrale. Ph.D. thesis, Universität Erlangen-Nürnberg (1994)
6. Knuth, D.E.: Combinatorial Algorithms, *The Art of Computer Programming*, vol. 4 (2005–2009)
7. Lyness, J.N.: An introduction to lattice rules and their generator matrices. IMA Journal of Numerical Analysis **9**, 405–419 (1989)
8. Möller, H.: Lower bounds for the number of nodes in cubature formulae. Numerische Integration **45**, 221–230 (1979)
9. Morrow, C.R., Patterson, T.N.L.: Construction of algebraic cubature rules using polynomial ideal theory. SIAM Journal on Numerical Analysis **15**(5), 953–976 (1978)
10. Noskov, M.: Analogs of Morrow-Patterson type cubature formulas. Journal of Computation Mathematics and Mathematical Physics **30**, 1254–1257 (1991)
11. Poppe, K., Cools, R.: CHEBINT: Operations on multivariate Chebyshev approximations. https://lirias.kuleuven.be/handle/123456789/325973
12. Sloan, I.H.: Polynomial interpolation and hyperinterpolation over general regions. Journal of Approximation Theory **83**(2), 238–254 (1995)
13. Sloan, I.H., Joe, S.: Lattice Methods for Multiple Integration. Oxford Science Publications (1994)
14. Trefethen, L.N.: Is Gauss quadrature better than Clenshaw-Curtis? SIAM Review **50**(1), 67–87 (2008)
15. Trefethen, L.N., Hale, N., Platte, R.B., Driscoll, T.A., Pachón, R.: Chebfun version 3. Oxford University (2009). http://www.maths.ox.ac.uk/chebfun/

Approximation of Functions from a Hilbert Space Using Function Values or General Linear Information

Ralph Tandetzky

Abstract We give an example of a Hilbert space embedding $H \subset \ell_p$, $1 \le p < \infty$, whose approximation numbers tend to zero much faster than its sampling numbers. The main result shows that optimal algorithms for approximation that use only function evaluation can be more than polynomially worse than algorithms using general linear information.

1 The Problem

We consider the problem of approximating the embedding operator

$$I : H \to L_p(X, \mu)$$

from a Hilbert space H of functions on a measure space (X, μ) into the space $L_p(X, \mu)$, where all functionals $f \mapsto f(x)$, $H \to \mathbb{R}$ are continuous for $x \in X$. In order to simplify the formulas we will neglect writing μ all the time and we replace (X, μ) by X. To approximate the operator I we allow algorithms of the form

$$A_n(f) = \varphi(\alpha_1(f), \ldots, \alpha_n(f)),$$

where $\varphi : \mathbb{R}^n \to L_p(X)$ can be any continuous mapping and $\alpha_1, \ldots, \alpha_n$ are bounded linear functionals (i.e. general linear information) that can be chosen adaptively. That means, that α_i may depend on $\alpha_1(f), \ldots, \alpha_{i-1}(f)$. It is well-known that the best algorithms need no adaption and are linear as stated in [4]. Therefore, we can restrict ourselves to algorithms of the form

R. Tandetzky
Mathematisches Institut, University of Jena, Ernst-Abbe-Platz 2, Jena, 07743, Germany
e-mail: ralph.tandetzky@googlemail.com

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_38, © Springer-Verlag Berlin Heidelberg 2012

$$A_n(f) = \sum_{i=1}^{n} \alpha_i(f) h_i \qquad (\alpha_i \in H', h_i \in L_p(X))$$

without loss of generality. If we allow only function evaluations $f \mapsto f(x)$ instead of general bounded linear functionals, then linear algorithms are again optimal.

In order to measure how difficult the approximation problem is, when we have general information or only function evaluations, we define the approximation numbers $a_n(H \subset L_p(X))$ and the sampling numbers $g_n(H \subset L_p(X))$.

Definition 1. Let $H \subset L_p(X)$ be a Hilbert space of functions f on a set X, such that the function evaluations $x^* : H \to \mathbb{R}$, $f \mapsto f(x)$ are continuous for $x \in X$. The approximation numbers $a_n(H \subset L_p)$ and the sampling numbers $g_n(H \subset L_p)$ are defined as

$$a_n(H \subset L_p(X)) := \inf_{\substack{\alpha_1,\dots,\alpha_n \in H' \\ h_1,\dots,h_n \in L_p}} \sup_{\substack{f \in H \\ \|f\|_H \le 1}} \left\| f - \sum_{i=1}^{n} \alpha_i(f) h_i \right\|_p ,$$

$$g_n(H \subset L_p(X)) := \inf_{\substack{x_1,\dots,x_n \in X \\ h_1,\dots,h_n \in L_p}} \sup_{\substack{f \in H \\ \|f\|_H \le 1}} \left\| f - \sum_{i=1}^{n} f(x_i) h_i \right\|_p .$$

Remark 1. The approximation numbers $a_n(H \subset L_p(X))$ can be understood to be the worst case error (measured in $L_p(X)$) of the best algorithm that uses general linear information. On the other hand the sampling numbers $g_n(H \subset L_p(X)$ are the corresponding errors if only function evaluations are permitted. If it is clear, what embedding we are talking about, we will omit it in our notation. Obviously, $a_n \le g_n$ for every embedding $H \subset L_p(X)$.

For many spaces the approximation numbers are known or at least can be estimated very well, because they are analytically easy to handle. For the sampling numbers this is not the case, however, they are more interesting in numerical analysis, since computer algorithms can usually access function values, but not general linear functionals. In many applications the sampling numbers are almost as good as the approximation numbers. This leads to the question: "When are the sampling numbers roughly as good as the approximation numbers?" We want to ask this question more precisely in terms of the rate of convergence.

Definition 2. The order of convergence of a null sequence (c_n) is defined as

$$r(c_n) := \sup\{\beta \in \mathbb{R} : \lim_{n\to\infty} c_n n^\beta = 0\}.$$

For example the order of convergence of $n^{-a}(\log n)^b$ is a, for $a > 0$ and $b \in \mathbb{R}$. Now the question is whether for embeddings $H \subset L_p(X)$ it holds

$$r(a_n) = r(g_n).$$

We will summarize what is known in the following section.

2 What is Known?

As an improvement of [5], Kuo et al. [2] showed that for $p = 2$ and $r(a_n) > \frac{1}{2}$ it holds

$$r(g_n) \geq \underbrace{\frac{2r(a_n)}{2r(a_n)+1}}_{>1/2} r(a_n) > \frac{1}{2} r(a_n).$$

This is true for all continuous embeddings $H \subset L_2(X)$ for which all function evaluations are continuous. On the other hand, Hinrichs et al. [1] constructed embeddings with $r(a_n) = \frac{1}{2}$ and $r(g_n) = 0$, where again $p = 2$. This shows that there is a discontinuity at $r(a_n) = \frac{1}{2}$ for the worst possible rate of convergence of the sampling numbers. In this paper the result of Hinrichs et al. will be generalized from $p = 2$ to arbitrary $p \in [1, \infty)$.

A summary of convergence rates in various settings is given in [3].

3 Results

Theorem 1. *For each p with $1 \leq p < \infty$ there exists a Hilbert space $H \subset \ell_p$ with*

$$r(a_n(H \subset \ell_p)) = \min\left\{\frac{1}{p}, \frac{1}{2}\right\} \qquad \textit{and} \qquad r(g_n(H \subset \ell_p)) = 0\,.$$

The result of Hinrichs, Novak and Vybíral is a little bit more general than stated above. However, for the question of what convergence rate we get for the sampling numbers, if we know the convergence rate of the approximation numbers, this is the answer implied by their results. For $p = 2$ we get the same answer. For $p < 2$ the convergence rate of the approximation numbers does not change in our result, but stays $\frac{1}{2}$. For $p > 2$ the convergence rate is still positive but tends to 0 for $p \to \infty$. The methods used in the proof do not yield any particular result for $p = \infty$ that is non-trivial.

4 Proof

The proof can be divided into three steps. In the first step we consider finite-dimensional spaces in which the approximation numbers are much smaller than the sampling numbers. In a second step we prove a lemma that roughly states that if finite-dimensional examples exist, which have sufficiently large sampling numbers compared to the approximation numbers, then there are infinite-dimesional Hilbert space embeddings $H \subset \ell_p$ which have large sampling numbers. In the last step we choose parameters for the finite-dimensional spaces which match the assumptions in the lemma of the second step.

4.1 First Step: The Finite-Dimensional Case

Definition 3. Let $N \in \mathbb{N}$ and $\delta > \varepsilon > 0$. We define the Hilbert space

$$H_{N,\delta,\varepsilon} := \mathbb{R}^N$$

with the norm

$$\|x\|_{H_{N,\delta,\varepsilon}} := \sqrt{\frac{1}{\delta^2}(x,y)^2 + \frac{1}{\varepsilon^2}\|(x-(x,y)y\|_2^2},$$

where $y = N^{-1/2}(1,\dots,1) \in \mathbb{R}^N$. The term (x,y) is meant to be the scalar product in ℓ_2^N.

These spaces have small approximation numbers and large sampling numbers, if ε is small compared to δ and N is large. This is, what the following proposition states.

Proposition 1. *The approximation numbers and sampling numbers of the embedding $H_{N,\delta,\varepsilon} \subset \ell_p^N$ satisfy the inequalities*

$$\begin{aligned}
a_0 &\le \delta N^{1/p-1/2} + \varepsilon && \textit{for } p > 2,\\
a_0 &\le \delta N^{1/p-1/2} && \textit{for } p \le 2,\\
a_n &\le \varepsilon && \textit{for } n \ge 1, p \ge 2,\\
a_n &\le \varepsilon N^{1/p-1/2} && \textit{for } n \ge 1, p < 2,\\
g_n &\ge \frac{(N-n)^{1/p-1/2}}{\sqrt{\frac{1}{\delta^2} + \frac{n}{\varepsilon^2 N}}} && \textit{for all } n = 0,\dots,N \textit{ and } p \textit{ with } 1 \le p < \infty.
\end{aligned}$$

In order to show this, we recall the following result.

Lemma 1. *Let $H \subset L_p(X)$ be a Hilbert space of functions on a set X, such that for any fixed $x \in X$ the functional $x^* : H \to \mathbb{R}$, $f \mapsto f(x)$ is continuous. Then the approximation numbers and sampling numbers can be described by*

$$a_n(H \subset L_p(X)) = \inf_{\alpha_1,\ldots,\alpha_n \in H'} \sup_{\substack{f \in H\setminus\{0\} \\ \alpha_i(f)=0 \forall i}} \frac{\|f\|_p}{\|f\|_H},$$

$$g_n(H \subset L_p(X)) = \inf_{x_1,\ldots,x_n \in X} \sup_{\substack{f \in H\setminus\{0\} \\ f(x_i)=0 \forall i}} \frac{\|f\|_p}{\|f\|_H}.$$

This result is well-known and we will not prove it here.

Proof (Proof of Proposition 1). We use the notation $H := H_{N,\delta,\varepsilon}$ throughout the proof. We will first discuss the estimate for the approximation numbers starting with $n = 0$. For $p > 2$ and $x \in H$ we get from the monotonicity of the ℓ_p norms

$$\begin{aligned}\|x\|_p &\leq \|x - (x,y)y\|_p + \|(x,y)y\|_p \leq \|x-(x,y)y\|_2 + |(x,y)| \cdot \|y\|_p \\ &\leq \varepsilon\|x\|_H + \delta\|x\|_H N^{1/p-1/2} = (\varepsilon + \delta N^{1/p-1/2})\|x\|_H .\end{aligned}$$

Hence we get $a_0 = \|Id\|_{H \to \ell_p^N} \leq \varepsilon + \delta N^{1/p-1/2}$ for $p > 2$. On the other hand, for $p \leq 2$ and $x \in H$ we obtain from Hölders inequality[1]

$$\begin{aligned}\|x\|_p &\leq \|x\|_2 \|(1,\ldots,1)\|_{\frac{1}{1/p-1/2}} = \sqrt{\|(x,y)y\|_2^2 + \|x-(x,y)y\|_2^2} \cdot N^{1/p-1/2} \\ &\leq \delta \|x\|_H N^{1/p-1/2} .\end{aligned}$$

This shows $a_0 = \|Id\|_{H \to \ell_p^N} \leq \delta N^{1/p-1/2}$ for $p \leq 2$.

We continue with the case $n \geq 1$ and $p \geq 2$. It suffices to show $a_1 \leq \varepsilon$, since the sequence of approximation numbers is monotonically decreasing. We use the representation from Lemma 1. If $\alpha_1(x) := (x,y) = 0$, then

$$\|x\|_H = \sqrt{\frac{1}{\delta^2}(x,y)^2 + \frac{1}{\varepsilon^2}\|x-(x,y)y\|_2^2} = \frac{1}{\varepsilon}\|x\|_2 \geq \frac{1}{\varepsilon}\|x\|_p .$$

Hence $a_1 \leq \varepsilon$.

Now let us consider the case $n \geq 1$ and $p < 2$. Again, we use the formula from Lemma 1. We show the result for $n+1$ assuming $n \geq 0$. In order to do so, we define $n+1$ functionals $\alpha_0(x) := (x,y)$, $\alpha_i(x) = x_i$ for $i = 1,\ldots,n$. For an x with $\alpha_0(x) = 0,\ldots,\alpha_n(x) = 0$ we have

$$\begin{aligned}\|x\|_H &= \sqrt{\frac{1}{\delta^2}(x,y)^2 + \frac{1}{\varepsilon^2}\|x-(x,y)y\|_2^2} = \frac{1}{\varepsilon}\|x\|_2 \\ &\leq \frac{1}{\varepsilon}\|Id\|_{\ell_2^{N-n} \to \ell_p^{N-n}} \|x\|_p \leq \frac{1}{\varepsilon}(N-n)^{1/p-1/2}\|x\|_p .\end{aligned}$$

[1] We use the Hölder inequality in the form $\|(x_n y_n)\|_p \leq \|(x_n)\|_q \|(y_n)\|_r$, where $1/p = 1/q + 1/r$. In our case we have $q = 2$ and $r = \frac{1}{1/p-1/2}$.

The dimensions of ℓ_2^{N-n} and ℓ_p^{N-n} are $N - n$ because there are only $N - n$ coordinates of x that can be different from zero. Thus we obtain

$$a_{n+1} \leq \frac{1}{\varepsilon}(N-n)^{1/p-1/2} \qquad \text{for} \qquad n \geq 0.$$

Finally, let us investigate the sampling numbers. We can write the formula in Lemma 1 in the following way:

$$g_n = \inf_{\substack{\Lambda \subseteq \{1,\ldots,N\} \\ \#\Lambda = n}} \sup_{\substack{x \in H \setminus \{0\} \\ x_i = 0 \forall i \in \Lambda}} \frac{\|x\|_p}{\|x\|_H}.$$

For a given $\Lambda \subseteq \{1, \ldots, N\}$ with $\#\Lambda = n$ we define a concrete $x^\Lambda \in H$ to get the estimate for the g_n from below. We put

$$x_i^\Lambda := \begin{cases} 1 & \text{for } i \notin \Lambda, \\ 0 & \text{for } i \in \Lambda. \end{cases}$$

We get

$$\begin{aligned} \frac{\|x^\Lambda\|_p}{\|x^\Lambda\|_H} &= \frac{(N-n)^{1/p}}{\sqrt{\frac{1}{\delta^2}\frac{(N-n)^2}{N} + \frac{1}{\varepsilon^2}\left(\sum_{i \in \Lambda}\left(\frac{N-n}{N}\right)^2 + \sum_{i \notin \Lambda}\left(\frac{n}{N}\right)^2\right)}} \\ &= \frac{(N-n)^{1/p}}{\sqrt{\frac{(N-n)^2}{\delta^2 N} + \frac{n(N-n)^2 + (N-n)n^2}{\varepsilon^2 N^2}}} \geq \frac{(N-n)^{1/p-1/2}}{\sqrt{\frac{1}{\delta^2} + \frac{n}{\varepsilon^2 N}}}. \end{aligned}$$

This completes the proof.

4.2 Second Step: A Lemma

Lemma 2. *Let $p \in [1, \infty)$. If $p < 2$, then let $(C_M)_{M \in \mathbb{N}}$ be a sequence of positive numbers in $\ell_{\frac{1}{1/p-1/2}}$, otherwise put $C_M := 1$ for all $M \in \mathbb{N}$. Furthermore, let $(\kappa_M)_{M \in \mathbb{N}}$ and $(\lambda_M)_{M \in \mathbb{N}}$ be convergent sequence of real numbers with*

$$\kappa := \lim_{M \to \infty} \kappa_M > \lim_{M \to \infty} \lambda_M =: \lambda.$$

If for every $M \in \mathbb{N}^+$ there are an $N \in \mathbb{N}^+$ and an embedding of a Hilbert space $H_M \subset \ell_p^N$, such that

$$a_n(H_M \subset \ell_p^N) \le \frac{1}{(M+n)^{\kappa_M}} \qquad \textit{for } \underline{\textit{all}}\ n \in \{0,\dots,N\},$$

$$g_n(H_M \subset \ell_p^N) \ge \frac{1}{C_M n^{\lambda_M}} \qquad \textit{for } \underline{\textit{some}}\ n \in \{1,\dots,N\},$$

then there exists an embedding of a Hilbert space $H \subset \ell_p$ with

$$r(a_n(H \subset \ell_p)) \ge \kappa > \lambda \ge r(g_n(H \subset \ell_p)).$$

This space H is a weighted direct sum of infinitely many spaces H_M. If $p \ge 2$ then the direct sum is not weighted.

Proof. We define the sequence $(M_k)_{k\in\mathbb{N}}$, $(N_k)_{k\in\mathbb{N}}$ and $(n_k)_{k\in\mathbb{N}}$ inductively by the following properties:

1. $M_0 := 1$
2. $M_{k+1} = M_k + N_k$ for $k \in \mathbb{N}$.
3. For $k \in \mathbb{N}$ the embedding $H_{M_k} \subset \ell_p^{N_k}$ shall satisfy

$$a_n(H_{M_k} \subset \ell_p^{N_k}) \le \frac{1}{(M_k+n)^{\kappa_{N_k}}} \qquad \text{for all } n \in \{0,\dots,N_k\},$$

$$g_{n_k}(H_{M_k} \subset \ell_p^{N_k}) \ge \frac{1}{C_{M_k} n_k^{\lambda_{M_k}}}.$$

Such sequences $(M_k)_{k\in\mathbb{N}}$, $(N_k)_{k\in\mathbb{N}}$ and $(n_k)_{k\in\mathbb{N}}$ exist, but they are not uniquely defined. We select one such triple of sequence. Based on that, for every sequence $x = (x_n)_{n=1,2,\dots}$ of real numbers, we define

$$P_k x := (x_{M_k},\dots,x_{M_{k+1}-1}) \in H_{M_k} \qquad \text{and}$$

$$\|x\|_H := \sqrt{\sum_{k=0}^{\infty} C_{M_k}^{-2} \|P_k x\|_{H_{M_k}}^2}\,.$$

We put

$$H := \bigoplus_{k=0}^{\infty} H_{M_k} := \{x = (x_n)_{n=1,2,\dots} \subseteq \mathbb{R} : \|x\|_H < \infty\}.$$

Firstly, we will show that $r(a_n(H \subset \ell_p)) \ge \kappa$. Let us take any $n = 1,2,\dots$ and choose $k \in \mathbb{N}$ with $M_k \le n < M_{k+1}$. Now let us pick linear functionals $\tilde{\alpha}_{M_k},\dots,\tilde{\alpha}_n \in H'_{M_k}$ such that

$$\sup_{\substack{x\in H_{M_k}\setminus\{0\}\\ \tilde{\alpha}_i(x)=0\forall i}} \frac{\|x\|_p}{\|x\|_H} = a_{n-M_k}(H_{M_k}\subset \ell_p^{N_k})\,.$$

This is possible according to Lemma 1 because H_{M_k} is finite-dimensional. Furthermore, for $x\in H$ we define $\alpha_i(x):=x_i$ for $i=1,\dots,M_k-1$ and $\alpha_i(x):=\tilde{\alpha}_i(P_kx)$ for $i=M_k,\dots,n$. If $\alpha_1(x)=\cdots=\alpha_n(x)=0$ then for q with $q:=\infty$ for $p\ge 2$ and $1/q=1/p-1/2$ for $p<2$ we get

$$\begin{aligned}\|x\|_p &= \left(\sum_{i=0}^{\infty}\|P_ix\|_p^p\right)^{1/p} \le \left(\sum_{i=k}^{\infty} C_{M_i}^{-2}\|P_ix\|_p^2\right)^{1/2}\|(C_M)\|_q\\ &\le \Bigg(C_{M_k}^{-2}a_{n-M_k}(H_{M_k}\subset\ell_p^{N_k})^2\|P_kx\|_{H_{M_k}}^2\\ &\quad+\sum_{i=k+1}^{\infty}C_{M_i}^{-2}a_0(H_{M_i}\subset\ell_p^{N_i})^2\|P_ix\|_{H_{M_i}}^2\Bigg)^{1/2}\|(C_M)\|_q\\ &\le \left(\frac{C_{M_k}^{-2}}{n^{2\kappa_{M_k}}}\|P_kx\|_{H_{M_k}}^2+\sum_{i=k+1}^{\infty}\frac{C_{M_i}^{-2}}{M_i^{2\kappa_{M_i}}}\|P_ix\|_{H_{M_i}}^2\right)^{1/2}\|(C_M)\|_q\\ &\le \sup_{i\ge k}\frac{1}{n^{\kappa_{M_i}}}\|x\|_H\ \|(C_M)\|_q\,.\end{aligned}$$

Hence, by Lemma 1 we have

$$a_n(H\subset\ell_p)\le\sup_{i\ge k}\frac{\|(C_M)\|_q}{n^{\kappa_{M_i}}}\qquad\text{for } k \text{ such that } M_k\le n<M_{k+1}.$$

Since $\lim_{M\to\infty}\kappa_M=\kappa$, it follows that $r(a_n(H\subset\ell_p))\ge\kappa$.

Now we show the second estimate $r(g_n(H\subset\ell_p))\le\lambda$. We already know that

$$g_{n_k}(H_{N_k}\subset\ell_p^{N_k})\ge\frac{1}{C_{M_k}n_k^{\lambda_{M_k}}}\qquad\text{for } k\in\mathbb{N}.$$

We will now show that this implies a similar estimate for the sampling numbers of the embedding $H\subset\ell_p$. We obtain

$$\begin{aligned}g_{n_k}(H\subset\ell_p) &= \inf_{\substack{\Lambda\subset\{1,2,\dots\}\\ |\Lambda|=n_k}}\ \sup_{\substack{x\in H\setminus\{0\}\\ x_i=0\forall i\in\Lambda}}\frac{\|x\|_p}{\|x\|_H}\ge\inf_{\substack{\Lambda\subset\{1,\dots,N_k\}\\ |\Lambda|=n_k}}\ \sup_{\substack{x\in H_{N_k}\setminus\{0\}\\ (x)_i=0\forall i\in\Lambda}}\frac{\|x\|_2}{C_{M_k}^{-1}\|x\|_{H_{M_k}}}\\ &= C_{M_k}g_{n_k}(H_{M_k}\subset\ell_p^{N_k})\ge\frac{1}{n_k^{\lambda_{M_k}}}\,.\end{aligned}$$

We know that $\lambda_{N_k} \to \lambda$ for $k \to \infty$. Thus it remains to prove that $n_k \to \infty$ for $k \to \infty$. This, in turn, is evident from

$$\frac{1}{C_{M_k} n_k^{\lambda_{M_k}}} \leq g_{n_k}(H_{M_k} \subset \ell_p^{N_k}) \leq g_0(H_{M_k} \subset \ell_p^{N_k}) = a_0(H_{M_k} \subset \ell_p^{N_k}) \leq \frac{1}{M_k^{\kappa_{M_k}}}$$

$$\Longrightarrow \qquad n_k \geq M_k^{\kappa_{M_k}/\lambda_{M_k}} C_{M_k}^{-1/\lambda_{M_k}} \geq M_k \qquad \text{for large enough } k,$$

which completes the proof.

4.3 Final Step: Putting Everything Together

Having the first two steps, we are now able to prove Theorem 1. In order to do so, we take the finite-dimensional spaces from the first step and choose their parameters to fit the conditions of the lemma in the second step. We will have to make some case differentiations while we construct parameters that fulfill the conditions given in that lemma. Let us start with the sequence $(C_M)_{M\in\mathbb{N}}$. For $p < 2$ we pick any sequence of positive numbers in $\ell_{\frac{1}{1/p-1/2}}$, for example $C_M := 2^{-M}$. If $p \geq 2$ then we put $C_M := 1$ for all $M \in \mathbb{N}$. Furthermore, we choose $(\kappa_M)_{M\in\mathbb{N}}$ to be a sequence that converges to $\min\{1/p, 1/2\}$ from below, with $0 < \kappa_M < \min\{1/p, 1/2\}$ for all $M \in \mathbb{N}$. Furthermore, let $(\lambda_M)_{M\in\mathbb{N}}$ be a null sequence of positive numbers. Fix $M \in \mathbb{N}$. We put

$$H_M := H_{N,\delta,\varepsilon} \qquad \text{with}$$

$$\delta := \begin{cases} [M^{-\kappa_M} - (M+N)^{-\kappa_M}]\, N^{1/2-1/p} & \text{for } p \geq 2, \\ M^{-\kappa_M} N^{1/2-1/p} & \text{for } p < 2 \end{cases} \qquad \text{and}$$

$$\varepsilon := \begin{cases} (M+N)^{-\kappa_M} & \text{for } p \geq 2, \\ (M+N)^{-\kappa_M} N^{1/2-1/p} & \text{for } p < 2. \end{cases}$$

Note that we did not determine N yet. Therefore δ, ε and especially H_M still depend on N. For large enough N the inequality $\delta > \varepsilon$ is fulfilled as assumed in the definition of the $H_{N,\delta,\varepsilon}$ spaces. We will now continue the proof by showing the inequality for the approximation numbers required in Lemma 2 for every N (large enough, so that $\delta > \varepsilon$) and $n = 0, \ldots, N$. For $p \geq 2$ Proposition 1 yields

$$a_0(H_M \subset \ell_p^N) \leq \delta N^{1/p-1/2} + \varepsilon = M^{-\kappa_M} \qquad \text{and}$$

$$a_n(H_M \subset \ell_p^N) \leq \varepsilon = (M+N)^{-\kappa_M} \leq (M+n)^{-\kappa_M} \qquad \text{for } n = 1, \ldots, N.$$

This shows $a_n(H_M \subset \ell_p^N) \leq (M+n)^{-\kappa_M}$ for $n = 0, \dots, N$ for $p \geq 2$. For $p < 2$ this estimate is obvious, if we have an eye on Proposition 1. Hence we have proven the estimate for the approximation numbers required in Lemma 2 for all N large enough and $n = 0, \dots, N$. We will proceed by estimating the sampling numbers. Afterward we will select adequate N and n, for which the inequality for the sampling numbers required in Lemma 2 is satisfied. For every $n \in \mathbb{N}$ the sampling numbers can be estimated by

$$g_n(H_M \subset \ell_p^N) \geq \frac{(N-n)^{1/p-1/2}}{\sqrt{\frac{1}{\delta^2} + \frac{n}{\varepsilon^2 N}}}$$

according to Proposition 1. For $p \geq 2$ we get

$$\begin{aligned} g_n(H_M \subset \ell_p^N) &\geq \frac{(N-n)^{1/p-1/2}}{\sqrt{\frac{N^{2/p-1}}{[M^{-\kappa_M} - (M+N)^{-\kappa_N}]^2} + n(M+N)^{2\kappa_M} N^{-1}}} \\ &\overset{N\to\infty}{\sim} \frac{N^{1/p-1/2}}{\sqrt{N^{2/p-1}M^{2\kappa_M} + nN^{2\kappa_M-1}}} \\ &= \frac{1}{\sqrt{M^{2\kappa_M} + nN^{2\kappa_N-2/p}}} \overset{N\to\infty}{\longrightarrow} M^{-\kappa_M} . \end{aligned}$$

Here the $\sim$ symbol means, that the quotient of the right hand side and the left hand side converges to 1 for $N \to \infty$. Similarly, for $p < 2$ we obtain

$$\begin{aligned} g_n(H_N \subset \ell_p^N) &\geq \frac{(N-n)^{1/p-1/2}}{\sqrt{M^{2\kappa_M}N^{2/p-1} + nN^{-1}(M+N)^{2\kappa_M}N^{2/p-1}}} \\ &\overset{N\to\infty}{\sim} \frac{N^{1/p-1/2}}{\sqrt{M^{2\kappa_M}N^{2/p-1} + nN^{2\kappa_M-1}N^{2/p-1}}} \\ &= \frac{1}{\sqrt{M^{2\kappa_M} + nN^{2\kappa_M-1}}} \overset{N\to\infty}{\longrightarrow} M^{-\kappa_M} . \end{aligned}$$

Hence the lower bound for the sampling numbers g_n converges to $M^{-\kappa_M}$ in both cases. Now we choose an $n \in \mathbb{N}$ with $C_M^{-1}n^{-\lambda_M} \leq M^{-\kappa_M}$. Since the lower bound for the sampling numbers converges to $M^{-\kappa_M}$ for $N \to \infty$, there exists an $N \in \mathbb{N}$, such that

$$g_n(H_M \subset \ell_p^N) \geq C_M^{-1}n^{-\lambda_M}$$

can be accomplished. The claim of Theorem 1 follows from Lemma 2.

Acknowledgements I want to thank Erich Novak and Aicke Hinrichs for valuable discussions and ideas which lead to the results in this paper.

References

1. A. Hinrichs, E. Novak, J. Vybíral: Linear information versus function evaluations for L_2-approximation, J. Approx. Theory 152 (2008), pp. 97–107.
2. F. Y. Kuo, G. W. Wasilkowski, H. Woźniakowski: On the power of standard information for multivariate approximation in the worst case setting, J. Approx. Theory 158 (2009), pp. 97–125.
3. E. Novak, H. Woźniakowski: On the power of function values for the approximation problem in various settings.
4. J. F. Traub, H. Woźniakowski: A General Theory of Optimal Algorithms, Academic Press, New York, 1980.
5. G. W. Wasilkowski, H. Woźniakowski: On the power of standard information for weighted approximation, Found. Comput. Math. 1 (2001), pp. 417–434.

High Order Weak Approximation Schemes for Lévy-Driven SDEs

Peter Tankov

Abstract We propose new jump-adapted weak approximation schemes for stochastic differential equations driven by pure-jump Lévy processes. The idea is to replace the driving Lévy process Z with a finite intensity process which has the same Lévy measure outside a neighborhood of zero and matches a given number of moments of Z. By matching 3 moments we construct a scheme which works for all Lévy measures and is superior to the existing approaches both in terms of convergence rates and easiness of implementation. In the case of Lévy processes with stable-like behavior of small jumps, we construct schemes with arbitrarily high rates of convergence by matching a sufficiently large number of moments.

1 Introduction

Let Z be a d-dimensional Lévy process without diffusion component, that is,

$$Z_t = \gamma t + \int_0^t \int_{|y|\le 1} y\widehat{N}(dy, ds) + \int_0^t \int_{|y|>1} yN(dy, ds), \quad t \in [0, 1].$$

Here $\gamma \in \mathbb{R}^d$, N is a Poisson random measure on $\mathbb{R}^d \times [0, \infty)$ with intensity ν satisfying $\int 1 \wedge \|y\|^2\nu(dy) < \infty$ and $\widehat{N}(dy, ds) = N(dy, ds) - \nu(dy)ds$ denotes the compensated version of N. We study the case when $\nu(\mathbb{R}^d) = \infty$, that is, there is an infinite number of jumps in every interval of nonzero length a.s. Further, let X be an $\mathbb{R}^n$-valued adapted stochastic process, unique solution of the stochastic differential equation

P. Tankov
Laboratoire de Probabilités et Modeles Aléatoires, Université Paris Diderot – Paris 7,
Paris, France
e-mail: tankov@math.univ-paris-diderot.fr

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_39, © Springer-Verlag Berlin Heidelberg 2012

$$X_t = X_0 + \int_0^t h(X_{s-})dZ_s, \quad t \in [0,1], \tag{1}$$

where h is an $m \times d$ matrix.

In this article we are interested in the numerical evaluation of $E[f(X_1)]$ for a sufficiently smooth function f by Monte Carlo, via discretization and simulation of the process X. We propose new weak approximation algorithms for (1) and study their rate of convergence.

The traditional method to simulate X is to use the Euler scheme with constant time step

$$\hat{X}^n_{\frac{i+1}{n}} = \hat{X}^n_{\frac{i}{n}} + h\left(\hat{X}^n_{\frac{i}{n}}\right)\left(Z_{\frac{i+1}{n}} - Z_{\frac{i}{n}}\right).$$

This method has the convergence rate [6, 9]

$$|E[f(X_1)] - E[f\left(\hat{X}^n_1\right)]| \leq \frac{C}{n}$$

but suffers from two difficulties: first, for a general Lévy measure ν, there is no available algorithm to simulate the increments of the driving Lévy process and second, a large jump of Z occurring between two discretization points can lead to an important discretization error.

A natural idea due to Rubenthaler [11] (in the context of finite-intensity jump processes, this idea appears also in [2, 8]), is to approximate Z with a compound Poisson process by replacing the small jumps with their expectation

$$Z^\varepsilon_t := \gamma_\varepsilon t + \int_0^t \int_{|y|>\varepsilon} yN(dy, ds), \quad \gamma_\varepsilon = \gamma - \int_{\varepsilon<|y|\leq 1} y\nu(dy),$$

and then place discretization dates at all jump times of Z^ε.

The computational complexity of simulating a single trajectory using this method becomes a random variable, but the convergence rate may be computed in terms of the *expected* number of discretization dates, proportional to $\lambda_\varepsilon = \int_{|y|\geq\varepsilon} \nu(dy)$. When the jumps of Z are highly concentrated around zero, however, this approximation is too rough and the convergence rates can be arbitrarily slow.

In [7], the authors proposed a scheme which builds on Rubenthaler's idea of using the times of large jumps of Z as discretization dates but achieves better convergence rates. Their idea is, first, to approximate the small jumps of Z with a suitably chosen Brownian motion, in order to match not only the first but also the second moment of Z, and second, to construct an approximation to the solution of the continuous SDE between the times of large jumps. Similar ideas of Gaussian correction were recently used in [5] in the context of multilevel Monte Carlo methods for the problem (1). However, although diffusion approximation of small jumps improves the convergence rate, there are limits on how well the small jumps of a Lévy process can be approximated by a Brownian motion. In particular, the Brownian motion is a symmetric process, while a Lévy process may be asymmetric.

In this paper we develop new jump-adapted discretization schemes based on approximating the Lévy process Z with a finite intensity Lévy process Z^ε without diffusion part. Contrary to previous works, instead of simply truncating jumps smaller than ε, we construct efficient finite intensity approximations which match a given number of moments of Z. These approximations are superior to the existing approaches in two ways. First, given that Z^ε is a finite intensity Lévy process, the solution to (1) with Z replaced by Z^ε is easy to compute, either explicitly or with a fast numerical method, making it straightforward to implement the scheme. Second, by choosing the parameters of Z^ε in a suitable manner, one can, in principle, match an arbitrary number of moments of Z and obtain a discretization scheme with an arbitrarily high convergence rate.

The paper is structured as follows. In Sect. 2, we present the main idea of moment matching approximations and provide a basic error bound for such schemes. In Sect. 3, we introduce our first scheme which is based on matching 3 moments of Z and can be used for general Lévy processes. For Lévy processes with stable-like behavior of small jumps near zero, the scheme is shown to be rate-optimal. Finally, Sect. 4 shows how schemes of arbitrary order can be constructed by matching additional moments, in the context of one-dimensional Lévy processes with stable-like behavior of small jumps.

2 Moment Matching Compound Poisson Approximations

Let Z^ε be a finite intensity Lévy process without diffusion part approximating Z in a certain sense to be defined later:

$$Z_t^\varepsilon := \gamma_\varepsilon t + \int_0^t \int_{\mathbb{R}^d} y N^\varepsilon(dy, ds), \tag{2}$$

where N^ε is a Poisson random measure with intensity measure $dt \times \nu^\varepsilon$ such that $\lambda_\varepsilon := \nu^\varepsilon(\mathbb{R}^d) < \infty$.

In this paper we propose to approximate the process (1) by the solution to

$$d\hat{X}_t = h(\hat{X}_{t-}) dZ_t^\varepsilon, \quad \hat{X}_0 = X_0, \tag{3}$$

which can be computed by applying the Euler scheme at the jump times of Z^ε and solving the deterministic ODE $d\hat{X}_t = h(\hat{X}_t)\gamma_\varepsilon dt$ explicitly (or by a Runge–Kutta method[1]) between these jump times. The following proposition provides a

[1] In this paper, to simplify the treatment, we assume that the ODE is solved explicitly. Upper bounds on the additional error introduced by the Runge–Kutta method are given in [7, Proposition 7]. These bounds can be made arbitrarily small by taking a Runge–Kutta algorithm of sufficiently high order.

basic estimate for the weak error of such an approximation scheme. We impose the following alternative regularity assumptions on the functions f and h:

($\mathbf{H_n}$) $f \in C^n, h \in C^n$ $f^{(k)}$ and $h^{(k)}$ are bounded for $1 \leq k \leq n$ and $\int z^{2n} \nu(dz) < \infty$.

($\mathbf{H'_n}$) $f \in C^n$, $h \in C^n$, $h^{(k)}$ are bounded for $1 \leq k \leq n$, $f^{(k)}$ have at most polynomial growth for $1 \leq k \leq n$ and $\int |z|^k \nu(dz) < \infty$ for all $k \geq 1$.

Proposition 1. *Let Z and $\hat{Z}$ be Lévy processes with characteristic triplets $(0, \nu, \gamma)$ and $(0, \hat{\nu}, \hat{\gamma})$ respectively, and let X and $\hat{X}$ be the corresponding solutions of SDE* (1)*. Assume $\hat{\gamma} = \gamma$, $\hat{\nu} = \nu$ on $\{\|x\| > 1\}$, either* ($\mathbf{H_n}$) *or* ($\mathbf{H'_n}$) *for $n \geq 3$ and*

$$\int_{\mathbb{R}^d} x_{i_1} \ldots x_{i_k} \nu(dx) = \int_{\mathbb{R}^d} x_{i_1} \ldots x_{i_k} \hat{\nu}(dx), \quad 2 \leq k \leq n-1, \quad 1 \leq i_k \leq d. \tag{4}$$

Then

$$|E[f(\hat{X}_1) - f(X_1)]| \leq C \int_{\mathbb{R}^d} \|x\|^n |d\nu - d\hat{\nu}|,$$

where the constant C may depend on f, g, x and ν but not on $\hat{\nu}$.

Proof. To simplify notation, we give the proof in the case $m = d = 1$. Let $u(t, x) = E^{(t,x)}[f(X_1)]$. By Lemma 13 in [7], $u \in C^{1,n}([0, 1] \times \mathbb{R})$ and satisfies

$$\begin{aligned} &\frac{\partial u}{\partial t}(t, x) + \gamma \frac{\partial u}{\partial x}(t, x)h(x) + \int_{|y|>1} (u(t, x + h(x)y) - u(t, x))\, \nu\,(dy) \\ &\qquad + \int_{|y|\leq 1} \left(u(t, x + h(x)y) - u(t, x) - \frac{\partial u}{\partial x}(t, x)h(x)y\right) \nu(dy) = 0, \quad (5) \\ &u(1, x) = f(x). \end{aligned}$$

Applying Itô formula under the integral sign and using (5) and Lemma 11 in [7] (bounds on moments of $\hat{X}_t$) yields

$$\begin{aligned} &E[f(\hat{X}_1) - f(X_1)] = E[u(1, \hat{X}_1) - u(0, X_0)] \\ &= E\left[\int_0^1 \int_{\mathbb{R}} \left\{u(t, \hat{X}_t + h(\hat{X}_t)z) - u(t, \hat{X}_t) - h(\hat{X}_t)z\frac{\partial u}{\partial x}\right\} (d\hat{\nu} - d\nu)dt\right] \\ &\quad + E\left[\int_0^1 \int_{\mathbb{R}} \left\{u(t, \hat{X}_{t-} + h(\hat{X}_{t-})z) - u(t, \hat{X}_{t-})\right\} \hat{N}(dz, dt)\right] \\ &= E\left[\int_0^1 \int_{\mathbb{R}} \sum_{k=2}^{n-1} \frac{\partial^k u(t, \hat{X}_t)}{\partial x^k} h^k(\hat{X}_t)z^k(d\hat{\nu} - d\nu)dt + \text{remainder}\right], \\ &= E[\text{remainder}], \end{aligned}$$

where in the last line we used the moment matching condition (4) and the remainder coming from the Taylor formula can be estimated as

$$
\begin{aligned}
|\text{remainder}| &\le \int_0^1 \int_{\mathbb{R}} \sup_{0\le s\le 1} \left| \frac{\partial^n u(s,\hat{X}_s)}{\partial x^n} \right| |h(\hat{X}_t)|^n |z|^n |d\hat{\nu} - d\nu| dt \\
&\le C \sup_{0\le s\le 1} \left| \frac{\partial^n u(s,\hat{X}_s)}{\partial x^n} \right| \sup_{0\le s\le 1} |h(\hat{X}_s)|^n \int_{\mathbb{R}} |z|^n |d\hat{\nu} - d\nu|
\end{aligned}
$$

From the Lipschitz property of h and Lemma 13 in [7],

$$
\sup_{0\le s\le 1} \left| \frac{\partial^n u(s,\hat{X}_s)}{\partial x^n} \right| \sup_{0\le s\le 1} |h(\hat{X}_s)|^n \le C(1 + \sup_{0\le t\le 1} |\hat{X}_t|^p)
$$

for some $C < \infty$, where $p = n$ under $(\mathbf{H_n})$ and $p > n$ under $(\mathbf{H'_n})$. Following the arguments in the proof of Lemma 11 in [7], we get

$$
E[\sup_{0\le t\le 1} |\hat{X}_t|^p] \le C(1+|x|^p) \exp\left[c\left(|\bar{\gamma}|^p + \int_{\mathbb{R}} |z|^p \hat{\nu}(dz) + \left(\int_{\mathbb{R}} z^2 \hat{\nu}(dz) \right)^{p/2} \right) \right]
$$

for different constants C and c, where

$$
\bar{\gamma} = \hat{\gamma} + \int_{|z|>1} z\hat{\nu}(dz) = \gamma + \int_{|z|>1} z\nu(dz)
$$

by our assumptions. Since $\int_{\mathbb{R}} z^2 \hat{\nu}(dz) = \int_{\mathbb{R}} z^2 \nu(dz)$ by assumption, and

$$
\begin{aligned}
\int_{\mathbb{R}} |z|^p \hat{\nu}(dz) &\le \int_{|z|>1} |z|^p \hat{\nu}(dz) + \int_{|z|\le 1} |z|^2 \hat{\nu}(dz) \\
&= \int_{|z|>1} |z|^p \nu(dz) + \int_{|z|\le 1} |z|^2 \nu(dz),
\end{aligned}
$$

it is clear that $E[\sup_{0\le t\le 1} |\hat{X}_t|^p] \le C$ for some constant C which does not depend on $\hat{\nu}$. □

3 The 3-Moment Scheme

Our first scheme is based on matching the first 3 moments of the process Z. Let S^{d-1} be the unit sphere in the d-dimensional space, and $\nu(dr \times d\theta)$ be a Lévy measure on $\mathbb{R}^d$ written in spherical coordinates $r \in [0,\infty)$ and $\theta \in S^{d-1}$ and satisfying $\int_{[0,\infty)\times S^{d-1}} r^3 \nu(dr, d\theta) < \infty$. Denote by $\bar{\nu}$ the reflection of ν with respect to the origin defined by $\bar{\nu}(B) = \nu(\{x : -x \in B\})$. We introduce two measures on S^{d-1}:

$$\bar{\lambda}(d\theta) = \frac{1}{2}\int_{|r|\le\varepsilon} \frac{r^3}{\varepsilon^3}\,(\nu(dr,d\theta) - \bar{\nu}(dr,d\theta))$$

$$\lambda(d\theta) = \frac{1}{2}\int_{|r|\le\varepsilon} \frac{r^2}{\varepsilon^2}\,(\nu(dr,d\theta) + \bar{\nu}(dr,d\theta))\,.$$

The *3-moment scheme* is defined by

$$\nu_\varepsilon(dr,d\theta) = \nu(dr,d\theta)1_{r>\varepsilon} + \delta_\varepsilon(dr)\{\lambda(d\theta) + \bar{\lambda}(d\theta)\} \tag{6}$$

$$\gamma_\varepsilon = \gamma - \int_{[0,1]\times S^{d-1}} r\theta\,\nu_\varepsilon(dr,d\theta), \tag{7}$$

where δ_ε denotes a point mass at ε.

Proposition 2 (Multidimensional 3-moment scheme). *For every $\varepsilon > 0$, ν_ε is a finite positive measure satisfying*

$$\int_{\mathbb{R}^d} x_i x_j\,\nu(dx) = \int_{\mathbb{R}^d} x_i x_j\,\nu_\varepsilon(dx) \tag{8}$$

$$\int_{\mathbb{R}^d} x_i x_j x_k\,\nu(dx) = \int_{\mathbb{R}^d} x_i x_j x_k\,\nu_\varepsilon(dx), \quad 1 \le i,j,k \le d \tag{9}$$

$$\lambda_\varepsilon := \int_{\mathbb{R}^d} \nu_\varepsilon(dx) = \int_{\|x\|>\varepsilon} \nu(dx) + \varepsilon^{-2}\int_{\|x\|\le\varepsilon} \|x\|^2\nu(dx) \tag{10}$$

$$\int_{\mathbb{R}^d} \|x\|^4|d\nu - d\nu_\varepsilon| \le \int_{\|x\|\le\varepsilon} \|x\|^4\nu(dx) + \varepsilon^2\int_{\|x\|\le\varepsilon} \|x\|^2\nu(dx), \tag{11}$$

where the last inequality is an equality if $\nu(\{x : \|x\| = \varepsilon\}) = 0$.

Proof. The positivity of ν_ε being straightforward, let us check (8). Let $\{e_i\}_{i=1}^d$ be the coordinate vectors. Then,

$$\int_{\mathbb{R}^d} x_i x_j\,\nu_\varepsilon(dx) = \int_{[0,\infty)\times S^{d-1}} r^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\nu_\varepsilon(dr,d\theta)$$

$$= \int_{(\varepsilon,\infty)\times S^{d-1}} r^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\nu(dr,d\theta) + \int_{S^{d-1}} \varepsilon^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\{\lambda(d\theta) + \bar{\lambda}(d\theta)\}$$

$$= \int_{(\varepsilon,\infty)\times S^{d-1}} r^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\nu(dr,d\theta) + \int_{S^{d-1}} \varepsilon^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\lambda(d\theta)$$

$$= \int_{(0,\infty)\times S^{d-1}} r^2\langle\theta,e_i\rangle\langle\theta,e_j\rangle\nu(dr,d\theta) = \int_{\mathbb{R}^d} x_i x_j\,\nu(dx).$$

The other equations can be checked in a similar manner. □

Corollary 1. *Let $d = 1$. Then the 3-moment scheme can be written as*

$$\nu_\varepsilon(dx) = \nu(dx)1_{|x|>\varepsilon} + \lambda_+\delta_\varepsilon(dx) + \lambda_-\delta_{-\varepsilon}(dx)$$

$$\lambda_\pm = \frac{1}{2}\left\{\int_{|x|\leq\varepsilon}\frac{x^2}{\varepsilon^2}\nu(dx) \pm \int_{|x|\leq\varepsilon}\frac{x^3}{\varepsilon^3}\nu(dx)\right\}$$

Corollary 2 (Worst-case convergence rate). *Assume* $(\mathbf{H_4})$ *or* $(\mathbf{H_4'})$. *Then the solution $\hat{X}$ of* (3) *with the characteristics of Z^ε given by* (6)–(7) *satisfies*

$$|E[f(\hat{X}_1) - f(X_1)]| = o(\lambda_\varepsilon^{-1}).$$

as $\varepsilon \to 0$.

Proof. By Proposition 1 we need to show that

$$\lim_{\varepsilon\downarrow 0}\lambda_\varepsilon\int_{\mathbb{R}^d}\|x\|^4|d\nu - d\nu_\varepsilon| = 0.$$

By Proposition 2,

$$\begin{aligned}
&\lim_{\varepsilon\downarrow 0}\lambda_\varepsilon\int_{\mathbb{R}^d}\|x\|^4|d\nu - d\nu_\varepsilon| \\
&\leq \lim_{\varepsilon\downarrow 0}\left\{\int_{\|x\|>\varepsilon}\nu(dx) + \varepsilon^{-2}\int_{\|x\|\leq\varepsilon}\|x\|^2\nu(dx)\right\} \\
&\qquad\qquad \times\left\{\int_{\|x\|\leq\varepsilon}\|x\|^4\nu(dx) + \varepsilon^2\int_{\|x\|\leq\varepsilon}\|x\|^2\nu(dx)\right\} \\
&\leq 2\lim_{\varepsilon\downarrow 0}\varepsilon^2\left\{\int_{\|x\|>\varepsilon}\nu(dx) + \varepsilon^{-2}\int_{\|x\|\leq\varepsilon}\|x\|^2\nu(dx)\right\}\int_{\|x\|\leq\varepsilon}\|x\|^2\nu(dx) \\
&= 2\lim_{\varepsilon\downarrow 0}\varepsilon^2\int_{\|x\|>\varepsilon}\nu(dx)\int_{\|x\|\leq\varepsilon}\|x\|^2\nu(dx) \\
&\leq 2\int_{\mathbb{R}^d}\|x\|^2\nu(dx)\lim_{\varepsilon\downarrow 0}\varepsilon^2\int_{\|x\|>\varepsilon}\nu(dx) = 0,
\end{aligned}$$

where in the last line the dominated convergence theorem was used. □

In many parametric or semiparametric models, the Lévy measure has a singularity of type $\frac{1}{|x|^{1+\alpha}}$ near zero. This is the case for stable processes, tempered stable processes [10], normal inverse Gaussian process [1], CGMY [3] and other models. Stable-like behavior of small jumps is a standard assumption for the analysis of asymptotic behavior of Lévy processes in many contexts, and in our problem as well, this property allows to obtain a more precise estimate of the convergence rate. We shall impose the following assumption, which does not require the Lévy measure to have a density:

$(\mathbf{H}-\boldsymbol{\alpha})$ There exist $C > 0$ and $\alpha \in (0,2)$ such that[2]

$$l(r) \sim Cr^{-\alpha} \quad \text{as} \quad r \to 0 \tag{12}$$

where $l(r) := \int_{\|x\|>r} \nu(dx)$.

Corollary 3 (Stable-like behavior). *Assume* $(\mathbf{H}-\boldsymbol{\alpha})$ *and* $(\mathbf{H_4})$ *or* $(\mathbf{H_4'})$. *Then the solution* $\hat{X}$ *of* (3) *with the characteristics of* Z^ε *given by* (6)–(7) *satisfies*

$$|E[f(\hat{X}_1) - f(X_1)]| = O\left(\lambda_\varepsilon^{1-\frac{4}{\alpha}}\right).$$

Proof. Under $(\mathbf{H}-\boldsymbol{\alpha})$, by integration parts we get that for all $n \geq 2$,

$$\int_{\|x\|\leq r} \|x\|^n \nu(dx) \sim \frac{C\alpha}{n-\alpha} r^{n-\alpha} \quad \text{as } r \to 0.$$

Therefore, under this assumption,

$$\lambda_\varepsilon \sim \frac{2C}{2-\alpha}\varepsilon^{-\alpha} \quad \text{and} \quad \int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu_\varepsilon| \lesssim \frac{C\alpha(6-2\alpha)}{(2-\alpha)(4-\alpha)}\varepsilon^{4-\alpha} \quad \text{as } \varepsilon \to 0,$$

from which the result follows directly. □

Remark 1. It is interesting to compare the convergence rates for our jump-adapted moment matching scheme (6–7) with the convergence rates for the classical Euler scheme, known from the literature. In [6, 9], under assumptions similar to our $(\mathbf{H_4})$ or $(\mathbf{H_4'})$, it has been shown that the discretization error of the Euler scheme for Lévy-driven stochastic differential equations decays linearly with the computational effort. Since the worst-case error for our scheme decays like $o(\lambda_\varepsilon^{-1})$, our approach always outperforms the Euler scheme in this setting. Under stronger assumptions (similar to our $(\mathbf{H_8})$ or $(\mathbf{H_8'})$), [6, 9] give an expansion of the discretization error, making it possible to use the Romberg-Richardson extrapolation and obtain a quadratic convergence of the discretization error to zero. This is faster than our worst-case rate, but for stable-like processes with Blumenthal-Getoor index α, Corollary 3 shows that our approach outperforms the Euler scheme with Romberg extrapolation for $\alpha < \frac{4}{3}$. It is important to emphasize, that our scheme is usually much easier to implement than the Euler scheme because it does not require the simulation of increments of the process. If an approximate simulation algorithm

[2] Throughout this paper we write $f \sim g$ if $\lim \frac{f(x)}{g(x)} = 1$ and $f \lesssim g$ if $\limsup \frac{f(x)}{g(x)} \leq 1$.

of increments is used, the rate of convergence of the Euler scheme with Romberg extrapolation will in general be slower than quadratic [6]. Further comparisons of our scheme with the existing approaches are given in the numerical example below.

Rate-Optimality of the 3-Moment Scheme

From Proposition 1 we know that under the assumption ($\mathbf{H_4}$) or ($\mathbf{H_4'}$), the approximation error of a scheme of the form (2)–(3) can be measured in terms of the 4-th absolute moment of the difference of Lévy measures. We introduce the class of Lévy measures on $\mathbb{R}^d$ with intensity bounded by N:

$$\mathscr{M}^N = \{\nu \text{ Lévy measure on } \mathbb{R}^d,\ \nu(\mathbb{R}^d) \le N\}.$$

The class of Lévy measures with intensity bounded by λ_ε is then denoted by $\mathscr{M}^{\lambda_\varepsilon}$, and the smallest possible error achieved by any measure within this class is bounded from below by a constant times $\inf_{\nu' \in \mathscr{M}^{\lambda_\varepsilon}} \int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu'|$. The next result shows that as $\varepsilon \to 0$, the error achieved by the 3-moment scheme ν_ε differs from this lower bound by at most a constant multiplicative factor .

Proposition 3. *Assume* ($\mathbf{H} - \boldsymbol{\alpha}$) *and let* ν_ε *be given by* (6)*. Then,*

$$\limsup_{\varepsilon \downarrow 0} \frac{\int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu_\varepsilon|}{\inf_{\nu' \in \mathscr{M}^{\lambda_\varepsilon}} \int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu'|} < \infty.$$

Proof. Step 1. Let us first compute

$$\mathscr{E}_N := \inf_{\nu' \in \mathscr{M}^N} \int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu'|. \tag{13}$$

For $\nu' \in \mathscr{M}^N$, let $\nu' = \nu'_c + \nu'_s$ where ν'_c is absolutely continuous with respect to ν and ν'_s is singular. Then $\nu'_c(\mathbb{R}^d) \le N$ and

$$\int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu'| = \int_{\mathbb{R}^d} \|x\|^4 |d\nu - d\nu'_c| + \int_{\mathbb{R}^d} \|x\|^4 d\nu'_s.$$

Therefore, the minimization in (13) can be restricted to measures ν' which are absolutely continuous with respect to ν, or, in other words,

$$\mathscr{E}_N = \inf \int_{\mathbb{R}^d} \|x\|^4 |1 - \lambda(x)| \nu(dx),$$

where the inf is taken over all measurable functions $\lambda : \mathbb{R}^d \to \mathbb{R}^+$ such that $\int_{\mathbb{R}^d} \lambda(x)\nu(dx) \le N$. By a similar argument, one can show that it is sufficient to consider only functions $\lambda : \mathbb{R}^d \to [0, 1]$. Given such a function $\lambda(r, \theta)$, the spherically symmetric function

$$\hat{\lambda}(r) := \frac{\int_{S^{d-1}} \lambda(r,\theta)\nu(dr,d\theta)}{\int_{S^{d-1}} \nu(dr,d\theta)}$$

leads to the same values of the intensity and the minimization functional. Therefore, letting $\hat{\nu}(dr) := \int_{S^{d-1}} \nu(dr,d\theta)$,

$$\mathscr{E}_N = \inf_{0\leq\hat{\lambda}\leq 1} \int_0^\infty r^4(1-\hat{\lambda}(r))\hat{\nu}(dr) \quad \text{s. t.} \quad \int_0^\infty \hat{\lambda}(r)\hat{\nu}(dr) \leq N. \tag{14}$$

For every $e > 0$,

$$\mathscr{E}_N \geq \inf_{0\leq\hat{\lambda}\leq 1} \left\{ \int_0^\infty r^4(1-\hat{\lambda}(r))\hat{\nu}(dr) + e^4 \left(\int_0^\infty \hat{\lambda}(r)\hat{\nu}(dr) - N \right) \right\}.$$

The inf in the right-hand side can be computed pointwise and is attained by $\hat{\lambda}(r) = 1_{r>e} + \mu 1_{r=e}$ for any $\mu \in [0,1]$. Let $e(N)$ and $\mu(N)$ be such that

$$\hat{\nu}((e(N),\infty)) + \mu(N)\hat{\nu}(\{e(N)\}) = N.$$

Such a $e(N)$ can always be determined uniquely and $\mu(N)$ is determined uniquely if $\nu(\{e(N)\}) > 0$. It follows that $\hat{\lambda}(r) = 1_{r>e(N)} + \mu(N)1_{r=e(N)}$ is a minimizer for (14) and therefore

$$\mathscr{E}_N = \int_{\|x\|<e} \|x\|^4 \nu(dx) + (1-\mu)e^4\nu(\{x : \|x\| = e\}),$$

where e and μ are solutions of

$$\nu(\{x : \|x\| > e\}) + \mu\nu(\{x : \|x\| = e\}) = N.$$

Step 2. For every $\varepsilon > 0$, let $e(\varepsilon)$ and $\mu(\varepsilon)$ be solutions of

$$\nu(\{x : \|x\| > e(\varepsilon)\}) + \mu(\varepsilon)\nu(\{x : \|x\| = e(\varepsilon)\}) = \lambda_\varepsilon.$$

It is clear that $e(\varepsilon) \to 0$ as $\varepsilon \to 0$ and after some straightforward computations using the assumption $(\mathbf{H}-\boldsymbol{\alpha})$ we get that

$$\lim_{\varepsilon\to 0} \frac{e(\varepsilon)}{\varepsilon} = \left(\frac{2-\alpha}{2}\right)^{1/\alpha}.$$

Then,

$$\lim_{\varepsilon\downarrow 0}\frac{\int_{\mathbb{R}^d}\|x\|^4|d\nu - d\nu_\varepsilon|}{\mathscr{E}_{\lambda_\varepsilon}} = \lim_{\varepsilon\downarrow 0}\frac{\int_{\mathbb{R}^d}\|x\|^4|d\nu - d\nu_\varepsilon|}{\varepsilon^{4-\alpha}}\lim_{\varepsilon\downarrow 0}\frac{\varepsilon^{4-\alpha}}{e(\varepsilon)^{4-\alpha}}$$
$$\times\lim_{\varepsilon\downarrow 0}\frac{e(\varepsilon)^{4-\alpha}}{\int_{\|x\|<e(\varepsilon)}\|x\|^4\nu(dx)+(1-\mu(\varepsilon))e(\varepsilon)^4\nu(\{x:\|x\|=e(\varepsilon)\})}$$

Under $(\mathbf{H}-\boldsymbol{\alpha})$ the three limits are easily computed and we finally get

$$\lim_{\varepsilon\downarrow 0}\frac{\int_{\mathbb{R}^d}\|x\|^4|d\nu - d\nu_\varepsilon|}{\mathscr{E}_{\lambda_\varepsilon}} = (3-\alpha)\left(\frac{2}{2-\alpha}\right)^{4/\alpha}. \qquad (15)$$

□

Remark 2. The constant $(3-\alpha)\left(\frac{2}{2-\alpha}\right)^{4/\alpha} > 1$ appearing in the right-hand side of (15) cannot be interpreted as a "measure of suboptimality" of the 3-moment scheme, but only as a rough upper bound, because in the optimization problem (13) the moment-matching constraints were not imposed (if they were, it would not be possible to solve the problem explicitly). On the other hand, the fact that this constant is unbounded as $\alpha \to 2$ suggests that such a rate-optimality result cannot be shown for general Lévy measures without imposing the assumption $(\mathbf{H}-\boldsymbol{\alpha})$.

Numerical Illustration

We shall now illustrate the theoretical results on a concrete example of a SDE driven by a normal inverse Gaussian (NIG) process [1], whose characteristic function is

$$\phi_t(u) := E[e^{iuZ_t}] = \exp\left\{-\delta t\left(\sqrt{\alpha^2-(\beta-iu)^2}-\sqrt{\alpha^2-\beta^2}\right)\right\},$$

where $\alpha > 0$, $\beta \in (-\alpha,\alpha)$ and $\delta > 0$ are parameters. The Lévy density is given by

$$\nu(x) = \frac{\delta\alpha}{\pi}\frac{e^{\beta x}K_1(\alpha|x|)}{|x|},$$

where K is the modified Bessel function of the second kind. The NIG process has stable-like behavior of small jumps with $\nu(x) \sim \frac{const}{|x|^2}$, $x \to 0$ (which means that $(\mathbf{H}-\boldsymbol{\alpha})$ is satisfied with $\alpha = 1$), and exponential tail decay. The increments of the NIG process can be simulated explicitly (see [4, Algorithms 6.9 and 6.10]), which enables us to compare our jump-adapted algorithm with the classical Euler scheme.

For the numerical example we solve the one-dimensional SDE

$$dX_t = \sin(aX_t)dZ_t,$$

where Z is the NIG Lévy process (with drift adjusted to have $E[Z_t] = 0$). The solution of the corresponding deterministic ODE

$$dX_t = \sin(aX_t)dt, \quad X_0 = x$$

is given explicitly by

$$X_t = \theta(t;x) = \frac{1}{a}\arccos\frac{1+\cos(ax) - e^{2at}(1-\cos(ax))}{1+\cos(ax) + e^{2at}(1-\cos(ax))}$$

Figure 1 presents the approximation errors for evaluating the functional $E[(X_1 - 1)^+]$ by Monte-Carlo using the 3-moment scheme described in this section (marked with crosses), the diffusion approximation of [7, Sect. 3] (circles), the classical Euler scheme (diamonds) and the Euler scheme with Romberg extrapolation (triangles). The parameter values are $\alpha \approx 3.038$, $\beta = 1.6$, $\delta \approx 0.323$, $a = 5$ and $X_0 = 1$. For each scheme we plot the logarithm of the approximation error as function of the logarithm of the computational cost (time needed to obtain a Monte Carlo estimator with the standard deviation approximately equal to that of the Euler scheme estimator with 2,000 discretization points and 10^6 Monte Carlo paths). The approximation error is defined as the difference between the computed value and the value given by the Euler scheme with 2,000 discretization points. The curves are obtained by varying the truncation parameter ε for the two jump-adapted schemes and by varying the discretization time step for the Euler scheme.

The approximation error for the Euler scheme is a straight line with slope corresponding to the theoretical convergence rate of $\frac{1}{n}$. The graph for the 3-moment scheme seems to confirm the theoretical convergence rate of λ_ε^{-3}; the scheme is much faster than the others and the corresponding curve quickly drops below the dotted line which symbolizes the level of the statistical error. One advantage of the Euler scheme is that the discretization error can be expanded [9], which allows one to use the Romberg-Richardson extrapolation techniques. However, Fig. 1 shows that in the considered example our scheme remains competitive even if the convergence of the Euler scheme is accelerated with such an extrapolation technique.

4 High Order Schemes for One-Dimensional Stable-Like Lévy Processes

In this section, we develop schemes of arbitrary order for Lévy processes with stable-like behavior of small jumps. Throughout this section, we take $d = 1$ and let Z be a Lévy process with characteristic triplet $(0, \nu, \gamma)$ satisfying the following refined version of $(\mathbf{H} - \boldsymbol{\alpha})$:

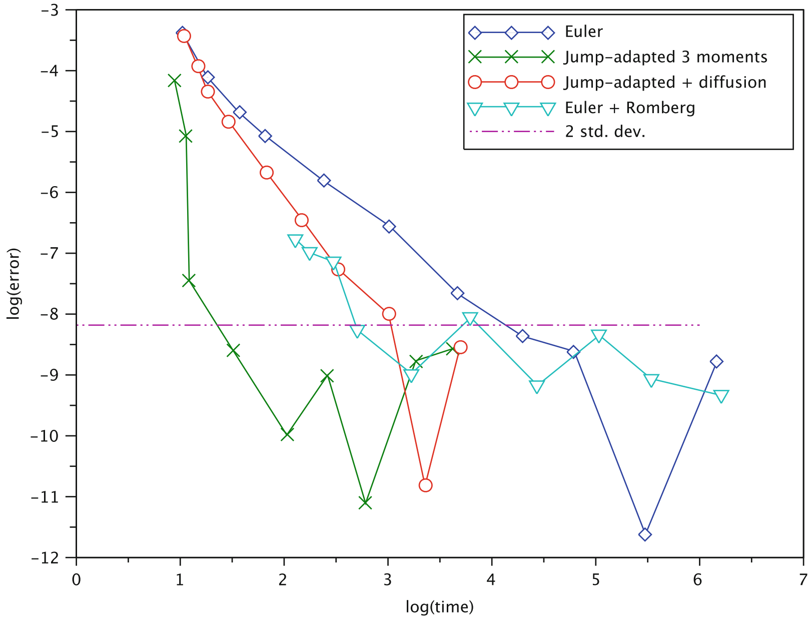


Fig. 1 Approximation errors for the 3-moment scheme (*crosses*), the scheme of [7, Sect. 3] (*circles*), the Euler scheme (*diamonds*) and the Euler scheme with Romberg extrapolation (*triangles*). The *horizontal dotted line* corresponds to the logarithm of the two standard deviations of the Monte Carlo estimator (the number of simulations for each numerical experiment was chosen to have approximately the same standard deviation for all points); everything that is below the *dotted line* is Monte Carlo noise.

(H′ − α) There exist, $c_+ \geq 0$, $c_- \geq 0$ with $c_+ + c_- > 0$ and $\alpha \in (0,2)$ such that

$$\int_\varepsilon^\infty \nu(dx) \sim c_+\varepsilon^{-\alpha} \quad \text{and} \quad \int_{-\infty}^{-\varepsilon} \nu(dx) \sim c_-\varepsilon^{-\alpha} \quad \text{as} \quad \varepsilon \downarrow 0$$

Introduce the probability measure

$$\mu^*(x) := \frac{(2-\alpha)|x|^{1-\alpha}(c_+ 1_{0\leq x\leq 1} + c_- 1_{-1\leq x\leq 0})}{c_+ + c_-}. \tag{16}$$

Let $n \geq 0$ and $\varepsilon > 0$. The high-order scheme for the stochastic differential equation (1) based on $n+2$ moments and truncation level ε is constructed as follows:

1. Find a discrete probability measure $\bar{\mu} = \sum_{i=0}^n a_i^* \delta_{x_i}$ with

$$\int_{\mathbb{R}} x^k \bar{\mu}(dx) = \int_{\mathbb{R}} x^k \mu^*(dx), \quad 1 \leq k \leq n, \tag{17}$$

such that $x_0 < x_1 < \cdots < x_n$, $x_i \neq 0$ for all i and $a_i^* > 0$ for all i.

2. Compute the coefficients $\{a_i^\varepsilon\}$ by solving the linear system

$$\sigma_\varepsilon^2 \sum_{i=0}^n a_i^\varepsilon x_i^k \varepsilon^k = \int_{|x|\le\varepsilon} x^{2+k}\nu(dx),\ k=0,\dots,n,\ \sigma_\varepsilon^2 = \int_{|x|\le\varepsilon} x^2\nu(dx).$$

3. The high-order scheme is defined by

$$\nu_\varepsilon(dx) = \nu(dx)1_{|x|>\varepsilon} + \sigma_\varepsilon^2 \sum_{i=0}^n \frac{a_i^\varepsilon \delta_{\varepsilon x_i}(dx)}{x_i^2\varepsilon^2}, \qquad \gamma_\varepsilon = \gamma - \int_{|z|\le 1} z\nu_\varepsilon(dz). \tag{18}$$

Remark 3. The first step in implementing the scheme is to solve the moment-matching problem (17) for measure μ^*. This problem will have, in general, an infinite number of solutions since there are many more unknowns than equations, and any of these solutions can be used to construct a high-order approximation scheme. An explicit solution for $n=3$ is given in Example 1.

Remark 4. It is easy to see that the measure

$$\nu_\varepsilon^*(dx) := (\sigma_\varepsilon^*)^2 \sum_{i=0}^n \frac{a_i^* \delta_{\varepsilon x_i}(dx)}{x_i^2\varepsilon^2}, \quad (\sigma_\varepsilon^*)^2 := \int_{|x|\le\varepsilon} x^2\nu^*(z)dz.$$

matches the moments of orders $2,\dots,n+2$ of $\nu^*(x)1_{|x|\le\varepsilon}$, where ν^* is the measure given by

$$\nu^*(x) = \frac{\alpha c_+ 1_{x>0} + \alpha c_- 1_{x<0}}{|x|^{1+\alpha}},$$

that is, ν^* satisfies the assumption $(\mathbf{H}'-\boldsymbol{\alpha})$ with equalities instead of equivalences. The idea of the method is to replace the coefficients $\{a_i^*\}$ with a different set of coefficients while keeping the same points $\{x_i\}$ to obtain a measure which matches the moments of $\nu(x)1_{|x|\le\varepsilon}$. Therefore, the points $\{x_i\}$ do not depend on the truncation parameter ε while the coefficients $\{a_i^\varepsilon\}$ depend on it.

Example 1. As an example we provide a possible solution of the moment matching problem for $n=3$, which leads to a 5-moment scheme (matching 3 moments of μ^* or 5 moments of the Lévy process). We assume that μ^* has mass both on the positive and the negative half-line: $c_+c_- > 0$.

The moments of μ^* are given by

$$m_k = \frac{2-\alpha}{k+2-\alpha}(\rho + (-1)^k(1-\rho)), \quad \rho := \frac{c_+}{c_+ + c_-}.$$

It is then convenient to look for the discrete measure matching the first 3 moments of μ^* in the form

$$\bar\mu = (1-\rho)(p\delta_{-\varepsilon_2} + (1-p)\delta_{-\varepsilon_1}) + \rho((1-p)\delta_{\varepsilon_1} + p\delta_{\varepsilon_2}), \tag{19}$$

where $p \in (0,1)$, $0 < \varepsilon_1 < \varepsilon_2$ are parameters to be identified from the moment conditions

$$(1-p)\varepsilon_1^k + p\varepsilon_2^k = \frac{2-\alpha}{k+2-\alpha}, \quad k = 1,2,3.$$

For the purpose of solving this system of equations, let $\mathscr{E}$ be a random variable such that $P[\mathscr{E} = \varepsilon_2] = p = 1 - P[\mathscr{E} = \varepsilon_1]$. From the moment conditions, we get:

$$\bar{\varepsilon} := E[\mathscr{E}] = \frac{2-\alpha}{3-\alpha}, \qquad \sigma^2 := \operatorname{Var}\mathscr{E} = \frac{(2-\alpha)}{(4-\alpha)(3-\alpha)^2}, \tag{20}$$

$$s := \frac{E[(\mathscr{E} - E[\mathscr{E}])^3]}{\sigma^3} = 2\frac{\alpha-1}{5-\alpha}\sqrt{\frac{4-\alpha}{2-\alpha}}. \tag{21}$$

On the other hand, the skewness s can be directly linked to the weight p:

$$s = \frac{1-2p}{\sqrt{p(1-p)}} \quad \Rightarrow \quad p = \frac{1}{2} - \frac{1}{2}\operatorname{sign}(s)\sqrt{\frac{s^2}{s^2+4}}, \tag{22}$$

and the parameters ε_1 and ε_2 can be linked to $\bar{\varepsilon}, \sigma$ and p:

$$\varepsilon_1 = \bar{\varepsilon} - \sigma\sqrt{\frac{p}{1-p}}, \quad \varepsilon_2 = \bar{\varepsilon} + \sigma\sqrt{\frac{1-p}{p}}. \tag{23}$$

The dependence of ε_1, ε_2 and p on α is shown in Fig. 2: it is clear from the graph that the constraints $p \in (0,1)$ and $0 < \varepsilon_1 < \varepsilon_2$ are satisfied for all $\alpha \in (0,2)$: therefore, Eqs. 19–23 define a four-atom probability measure which matches the first 3 moments of μ^*.

Proposition 4. *Let $\bar{\mu} = \sum_{i=0}^n a_i^* \delta_{x_i}$ be a solution of (17). There exists $\varepsilon_0 > 0$ such that for all $\varepsilon < \varepsilon_0$, ν_ε is a positive measure satisfying*

$$\int_{\mathbb{R}} x^k \nu(dx) = \int_{\mathbb{R}} x^k \nu_\varepsilon(dx), \quad 2 \le k \le n+2 \tag{24}$$

There exist positive constants C_1 and C_2 such that

$$\lambda_\varepsilon = \nu_\varepsilon(\mathbb{R}) \sim C_1\varepsilon^{-\alpha}, \qquad \int_{\mathbb{R}} |x|^{n+3}|d\nu - d\nu_\varepsilon| \lesssim C_2\varepsilon^{n+3-\alpha} \quad \text{as } \varepsilon \to 0.$$

Corollary 4. *Assume $(\mathbf{H_{n+3}})$ or $(\mathbf{H'_{n+3}})$. Then the solution $\hat{X}$ of (3) with characteristics of Z^ε given by (18) satisfies*

$$|E[f(\hat{X}_1) - f(X_1)]| = O\left(\lambda_\varepsilon^{1-\frac{n+3}{\alpha}}\right).$$

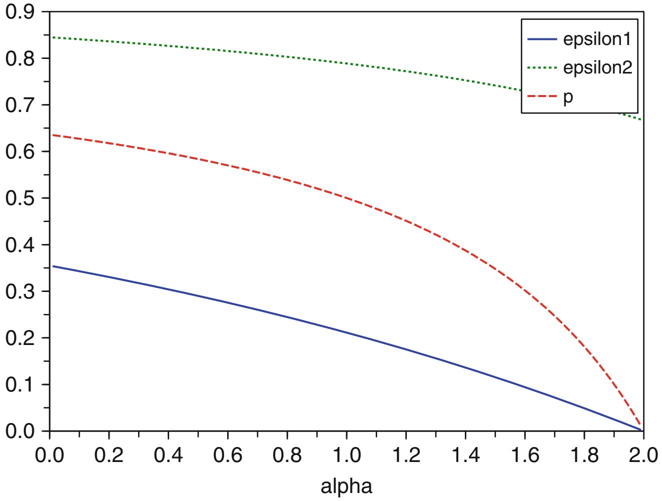


Fig. 2 Solution of the moment matching problem for 3 moments (see Example 1).

Proof (of Proposition 4). The moment conditions (24) hold by construction. Using integration by parts, we compute

$$\int_{|z|\leq\varepsilon} z^{2+k}\nu(dz) \sim \frac{(c_+ + (-1)^k c_-)\alpha\varepsilon^{2+k-\alpha}}{2+k-\alpha} \quad \text{as } \varepsilon \to 0 \text{ for } k \geq 0.$$

Therefore,

$$\lim_{\varepsilon\to 0}\frac{1}{\sigma_\varepsilon^2\varepsilon^k}\int_{|z|\leq\varepsilon} z^{2+k}\nu(dz) = \frac{(2-\alpha)(c_+ + (-1)^k c_-)}{(2+k-\alpha)(c_+ + c_-)} = \int_{\mathbb{R}} x^k \mu^*(dx).$$

Since the matrix $M_{ij} = (x_j)^i$, $0 \leq i \leq n$, $0 \leq j \leq n$ is invertible (Vandermonde matrix), this implies that $\lim_{\varepsilon\to 0} a_i^\varepsilon = a_i^*$. Therefore, there exists $\varepsilon_0 > 0$ such that for all $\varepsilon < \varepsilon_0$, $a_i^\varepsilon > 0$ for all i and ν_ε is a positive measure.

We next compute:

$$\nu_\varepsilon(\mathbb{R}) = \int_{|x|>\varepsilon}\nu(dx) + \frac{\sigma_\varepsilon^2}{\varepsilon^2}\sum_{i=0}^n \frac{a_i^\varepsilon}{x_i^2} \sim \int_{|x|>\varepsilon}\nu(dx) + \frac{\sigma_\varepsilon^2}{\varepsilon^2}\sum_{i=0}^n \frac{a_i^*}{x_i^2}$$

$$\sim \varepsilon^{-\alpha}(c_+ + c_-)\left\{1 + \frac{\alpha}{2-\alpha}\sum_{i=0}^n \frac{a_i^*}{x_i^2}\right\},$$

$$\int_{\mathbb{R}} |x|^{n+3}|d\nu - d\nu_\varepsilon| \le \int_{|x|\le\varepsilon} |x|^{n+3}d\nu + \sigma_\varepsilon^2\varepsilon^{n+1}\sum_{i=0}^{n} a_i^\varepsilon |x_i|^{n+1}$$

$$\sim \varepsilon^{n+3-\alpha}(c_+ + c_-)\left\{\frac{\alpha}{3+k-\alpha} + \frac{\alpha}{2-\alpha}\sum_{i=0}^{n} a_i^* |x_i|^{n+1}\right\}.$$

□

References

1. Barndorff-Nielsen, O.: Processes of normal inverse Gaussian type. Finance Stoch. **2**, 41–68 (1998)
2. Bruti-Liberati, N., Platen, E.: Strong approximations of stochastic differential equations with jumps. J. Comput. Appl. Math. **205**(2), 982–1001 (2007)
3. Carr, P., Geman, H., Madan, D., Yor, M.: The fine structure of asset returns: An empirical investigation. J. Bus. **75**(2), 305–332 (2002)
4. Cont, R., Tankov, P.: Financial Modelling with Jump Processes. Chapman & Hall / CRC Press (2004)
5. Dereich, S.: Multilevel Monte Carlo algorithms for Lévy-driven SDEs with Gaussian correction. Ann. Appl. Probab. **21**(1), 283–311 (2011)
6. Jacod, J., Kurtz, T.G., Méléard, S., Protter, P.: The approximate Euler method for Lévy driven stochastic differential equations. Ann. Inst. H. Poincaré Probab. Statist. **41**(3), 523–558 (2005)
7. Kohatsu-Higa, A., Tankov, P.: Jump-adapted discretization schemes for Lévy-driven SDEs. Stoch. Proc. Appl. **120**, 2258–2285 (2010)
8. Mordecki, E., Szepessy, A., Tempone, R., Zouraris, G.E.: Adaptive weak approximation of diffusions with jumps. SIAM J. Numer. Anal. **46**(4), 1732–1768 (2008)
9. Protter, P., Talay, D.: The Euler scheme for Lévy driven stochastic differential equations. Ann. Probab. **25**(1), 393–423 (1997)
10. Rosiński, J.: Tempering stable processes. Stoch. Proc. Appl. **117**, 677–707 (2007)
11. Rubenthaler, S.: Numerical simulation of the solution of a stochastic differential equation driven by a Lévy process. Stoch. Proc. Appl. **103**(2), 311–349 (2003)

High-Discrepancy Sequences for High-Dimensional Numerical Integration

Shu Tezuka

Abstract In this paper, we consider a sequence of points in $[0,1]^d$, which are distributed only on the diagonal line between $(0,\dots,0)$ and $(1,\dots,1)$. The sequence is constructed based on a one-dimensional low-discrepancy sequence. We apply such sequences to d-dimensional numerical integration for two classes of integrals. The first class includes isotropic integrals. Under a certain condition, we prove that the integration error for this class is $O(\sqrt{\log N}/N)$, where N is the number of points. The second class is called as Kolmogorov superposition integrals for which, under a certain condition, we prove that the integration error for this class is $O((\log N)/N)$.

1 Introduction

Low-discrepancy sequences (or quasi-Monte Carlo methods) have been widely used and successfully applied to high-dimensional (sometimes very high dimensions like 1,000 or more) numerical integration in the last two decades [3, 5]. The notion of discrepancy [4, 6, 10] originated from uniform distribution of sequences, a branch of analytic number theory, in early 1900s. Informally speaking, the lower (or smaller) discrepancy is, the more uniformly distributed are points in some domain.

On the other hand, the Kolmogorov superposition theorem tells that integration of any continuous high-dimensional function is written as the sum of one-dimensional integrations, and recently, research efforts (see, e.g., [1]) have been done to make the theorem numerically constructive. The theorem implies that any continuous high-dimensional function has "hidden" one-dimensional structure, whether it is

S. Tezuka
Faculty of Mathematics, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka-shi, Fukuoka-ken, 819-0395, Japan
e-mail: tezuka@math.kyushu-u.ac.jp

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_40, © Springer-Verlag Berlin Heidelberg 2012

explicit or not. Thus, exploiting such structure may lead us to show the existence of high-dimensional sequences that are not necessarily of low-discrepancy, whose convergence is faster than that of low-discrepancy sequences of the same dimensions.

In this paper, we consider a sequence of points in $[0, 1]^d$, which are distributed only on the diagonal line between $(0, \ldots, 0)$ and $(1, \ldots, 1)$, where the sequence is constructed from a one-dimensional low-discrepancy sequence. As it stands clearly, the points are extremely non-uniform in $[0, 1]^d$ and thus we call them *high-discrepancy sequences*. (Some relevant earlier results are found in [11, 12].) We apply such sequences to d-dimensional numerical integration for two classes of integrals. The first class includes *isotropic integrals*, and the second class is called as *Kolmogorov superposition integrals*. For these two classes of integrals, we prove that if we use appropriate high-discrepancy sequences for the integration, then the integration error becomes better than that of d-dimensional low-discrepancy sequences.

The organization of the paper is as follows: In Sect. 2, we recall isotropic integrals, and define the d-dimensional *isotropic high-discrepancy sequences*. Then, we prove that under a certain condition the integration error for this class of integrals is $O(\sqrt{\log N}/N)$, where N is the number of points. In Sect. 3, we first overview the Kolmogorov superposition theorem, which tells us that any continuous function on $[0, 1]^d$ can be represented by superposition of one-dimensional functions. Based on this theorem, we define Kolmogorov superposition integrals and then define the d-dimensional *Kolmogorov superposition high-discrepancy sequences*. We prove that under a certain condition the integration error for this class of integrals is $O((\log N)/N)$, where N is the number of points. In Sect. 4, we summarize software implementations of these two types of high-discrepancy sequences and give some numerical results. In the last section, we discuss the significance of the results.

2 High-Discrepancy Sequences for Isotropic Integrals

The following integral is called the d-dimensional isotropic integral:

$$I_d(h) = \int_{\mathbb{R}^d} h(||\mathbf{x}||)e^{-||\mathbf{x}||^2} d\mathbf{x}, \tag{1}$$

where $d \geq 1$ and $||x||$ denotes the Euclidean norm of $\mathbf{x}$ in $\mathbb{R}^d$. This kind of integral often appears in computational physics, and as practical examples, we know $h(x) = \cos(x)$ or $h(x) = \sqrt{1 + x^2}$. The research of the isotropic integral dates back to Keister [2], as well as Papageorgiou and Traub [9], more than 10 years ago. Although the integral can be reduced to one-dimensional by using spherical coordinates, they tackled (1) directly because it takes advantage of dependence on the norm automatically.

In numerical computations with d-dimensional low-discrepancy sequences, the domain, $\mathbb{R}^d$, of the integral is always transformed to the unit hypercube, $[0,1]^d$. Thus, we actually compute the following integral:

$$I_d(h) = \pi^{d/2} \int_{[0,1]^d} h\left(\sqrt{\frac{1}{2}\sum_{i=1}^{d}(\phi^{-1}(u_i))^2}\right) du_1 \cdots du_d,$$

where $\phi(x)$ is the standard normal distribution function.

We define an isotropic high-discrepancy sequence as follows:

Definition 1. An isotropic high-discrepancy sequence is defined as a sequence of points

$$P_n = (s_n, \ldots, s_n) \in [0,1]^d, \ n = 0, 1, 2, \ldots,$$

with $s_n = \phi(\chi^{-1}(v_n)/\sqrt{d})$, where $\chi(x)$, $(x \geq 0)$, is the chi-distribution function of degree d, and $v_n, n = 0, 1, 2, \ldots$, is a one-dimensional low-discrepancy sequence satisfying $D_N = c/N$, $(N = 1, 2, \ldots)$, with a constant $c \geq 1/2$, where D_N means the star discrepancy of the first N points.

As the definition makes clear, isotropic high-discrepancy sequences are distributed only on the diagonal line between $(0, \ldots, 0)$ and $(1, \ldots, 1)$. We should note that the sequence is constructed independently of $h(x)$, the integrand of isotropic integrals.

Based on the results of Papageorgiou [7], we obtain the following theorem:

Theorem 1. *For an isotropic integral, if the function $h(x)$ is absolutely continuous, $h'(x)$ exists almost everywhere, and* $\operatorname{ess\,sup}\{|h'(x)| : x \in \mathbb{R}\} \leq M$*, then the error of the numerical integration using an isotropic high-discrepancy sequence is given by $O(\sqrt{\log N}/N)$, where M is a constant and N is the number of points.*

Proof. If the points P_n, $n = 0, 1, \ldots$, in Definition 1 are used for the integration, we have

$$\begin{aligned}
Q_N(h) &= \frac{\pi^{d/2}}{N} \sum_{n=0}^{N-1} h\left(\sqrt{\frac{1}{2}\sum_{i=1}^{d}(\phi^{-1}(s_n))^2}\right) \\
&= \frac{\pi^{d/2}}{N} \sum_{n=0}^{N-1} h\left(\sqrt{\frac{d}{2}}\left|\phi^{-1}(s_n)\right|\right) \\
&= \frac{\pi^{d/2}}{N} \sum_{n=0}^{N-1} h\left(\sqrt{\frac{d}{2}}\left|\phi^{-1}\left(\phi\left(\frac{\chi^{-1}(v_n)}{\sqrt{d}}\right)\right)\right|\right) \\
&= \frac{\pi^{d/2}}{N} \sum_{n=0}^{N-1} h\left(\frac{\chi^{-1}(v_n)}{\sqrt{2}}\right).
\end{aligned}$$

From Theorem 2 of [7], we have

$$\pi^{-d/2}|I_d(h) - Q_N(h)| \leq C_1 \frac{\sqrt{\log N}}{N},$$

where C_1 is a constant dependent on d and M. Thus, the proof is complete. □

3 High-Discrepancy Sequences for Kolmogorov Superposition Integrals

The Kolmogorov superposition theorem tells us that for any integer $d \geq 1$, any continuous function $f(x_1, \ldots, x_d)$ on $[0,1]^d$ can be represented as a superposition of one-dimensional functions, i.e.,

$$f(x_1, \ldots, x_d) = \sum_{q=0}^{2d} g_q \left(\sum_{i=1}^{d} a_i \Psi_q(x_i) \right),$$

where $g_q(x), q = 0, 1, \ldots, 2d$, are functions determined depending on $f(x_1, \ldots, x_d)$, and $a_i, i = 1, \ldots, d$ are constants with $\sum_{i=1}^{d} a_i = 1$, determined independently of $f(x_1, \ldots, x_d)$. And $\Psi_i(x), i = 1, \ldots, d$, are monotone increasing and continuous function on $[0,1]$, determined independently of $f(x_1, \ldots, x_d)$. How to construct these functions and constants can be found in [1], which also includes more detail and latest information on this theorem.

Based on the theorem above, we define a Kolmogorov superposition integral as follows:

Definition 2. A Kolmogorov superposition integral is defined by

$$I_d(g) = \int_{[0,1]^d} g \left(\sum_{i=1}^{d} a_i \Psi(x_i) \right) dx_1 \cdots dx_d,$$

where $\Psi(x)$ is any monotone increasing function on $[0,1]$ which is continuous in $(0,1)$, and $a_1, \ldots, a_d$ are constants with $\sum_{i=1}^{d} a_i = 1$. And $g(x)$ is any function such that $|I_d(g)| < \infty$. Remark that $\Psi(x)$ can be $\pm\infty$ at the ends of the unit interval $[0,1]$.

We should note that isotropic integrals are not part of Kolmogorov superposition integrals, because $(\phi^{-1}(x))^2$ is not monotone increasing in $[0,1]$. We give a practical example of Kolmogorov superposition integral below.

Example 1. The option payoff function with maturity d days, the total time $T = d/365$, and the stock price being simulated daily has the following form:

$$f(x_1,\dots,x_d) = \max\left(S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{\frac{T}{d}}\sum_{j=1}^{d} x_j\right) - K, 0\right),$$

where S_0 the stock spot price, r the risk free interest rate, and σ annualized volatility. Therefore, this integral is written as the Kolmogorov superposition integral with

$$\begin{aligned} g(x) &= \max(a\exp(bx) - K, 0),\\ \Psi(x) &= \phi^{-1}(x),\\ a_1 &= \cdots = a_d = 1/d, \end{aligned}$$

where a and b are appropriate constants. According to Papageorgiou [8], many integrals that appear in finance have this form.

We now give the definition of a Kolmogorov superposition high-discrepancy sequence.

Definition 3. A Kolmogorov superposition high-discrepancy sequence is defined as a sequence of points

$$P_n = (s_n,\dots,s_n) \in [0,1]^d,\ n = 0,1,2,\dots,$$

with $s_n = \Psi^{-1}(p^{-1}(v_n))$, where $p(x)$ is the distribution function corresponding to $\sum_{i=1}^{d} a_i\Psi(x_i)$, and $v_n, n = 0,1,2,\dots$, is a one-dimensional low-discrepancy sequence.

The function $p(x)$ can be obtained either by repeatedly calculating convolutions or by using the product of the characteristic functions of probability distribution functions. As the definition makes it clear, the points of the Kolmogorov superposition high-discrepancy sequence are distributed only on the diagonal line between $(0,\dots,0)$ and $(1,\dots,1)$. We should note that the sequence is constructed independently of $g(x)$, the integrand of Kolmogorov superposition integrals. On the integration error, we obtain the following theorem:

Theorem 2. *Denote $\rho(x) = g(p^{-1}(x))$. For a Kolmogorov superposition integral, if the function $\rho(x)$ is of bounded variation, then the error of the numerical integration using a Kolmogorov superposition high-discrepancy sequence is given by $O((\log N)/N)$, where N is the number of points.*

Proof. If the points $P_n, n = 0,1,\dots$, in Definition 3 are used for the integration, we have

$$\left|\int_{[0,1]^d} g\left(\sum_{i=1}^{d} a_i\Psi(x_i)\right) dx_1\cdots dx_d - \frac{1}{N}\sum_{n=0}^{N-1} g\left(\sum_{i=1}^{d} a_i\Psi(s_n)\right)\right|$$

$$
\begin{aligned}
&= \left| \int_{-\infty}^{\infty} g(z)p'(z)dz - \frac{1}{N}\sum_{n=0}^{N-1} g(\Psi(s_n)) \right| \\
&= \left| \int_{0}^{1} g(p^{-1}(u))du - \frac{1}{N}\sum_{n=0}^{N-1} g(\Psi(\Psi^{-1}(p^{-1}(v_n)))) \right| \\
&= \left| \int_{0}^{1} \rho(u)du - \frac{1}{N}\sum_{n=0}^{N-1} \rho(v_n) \right| \\
&\leq V(\rho)D_N,
\end{aligned}
$$

where $V(\rho)$ is the variation of $\rho(x)$. Since $v_n, n = 0, 1, \ldots,$ is a one-dimensional low-discrepancy sequence, we have

$$|I_d(g) - Q_N(g)| \leq C_2 \frac{\log N}{N},$$

where C_2 is a constant generally dependent on d. Thus, the proof is complete. □

We should remark that the function $\rho(x)$ associated with the integral described in Example 1 is not of bounded variation. However, the results of Papageorgiou [8] imply that the integration error for the high-discrepancy sequence is $O(n^{-1+o(1)})$, where the asymptotic constant is independent of d.

4 Software Implementation

We summarize software implementations of the two types of high-discrepancy sequences discussed above. First, an implementation for isotropic high-discrepancy sequences (I-HDS) is given below.

[Software implementation of I-HDS]

Preprocessing:

By using the parameters, d, $\chi(x)$, and $\phi(x)$, compute I-HDS as

$$P_n = (s_n, \ldots, s_n) \in [0,1]^d, \; n = 0, 1, 2, \ldots,$$

with $s_n = \phi(\chi^{-1}(v_n)/\sqrt{d})$, where $v_n, n = 0, 1, 2, \ldots,$ is the van der Corput sequence. Then store it in the memory.

Main processing:

Once the integrand $h(x)$ is given, call N points of the I-HDS from the memory, and compute

$$\frac{\pi^{d/2}}{N}\sum_{n=0}^{N-1} h\left(\sqrt{\frac{d}{2}}\,\phi^{-1}(s_n)\right)$$

as an approximation to the integral. □

The next is an implementation for Kolmogorov superposition high-discrepancy sequences (KS-HDS).

[**Software implementation of KS-HDS**]

Preprocessing:

By using the parameters, d, $p(x)$, and $\Psi(x)$, compute KS-HDS as

$$P_n = (s_n, \dots, s_n) \in [0,1]^d,\ n = 0, 1, 2, \dots,$$

with $s_n = \Psi^{-1}(p^{-1}(v_n))$, where $v_n, n = 0, 1, 2, \dots,$ is the van der Corput sequence. Then store it in the memory.

Main processing:

Once the integrand $g(x)$ is given, call N points of the KS-HDS from the memory, and compute

$$\frac{1}{N}\sum_{n=0}^{N-1} g(\Psi(s_n))$$

as an approximation to the integral. □

For both implementations, if d is large, the functions $\chi(x)$ and $p(x)$ can be replaced by the normal distribution thanks to the central limit theorem.

In the following, we give results of numerical computations for the comparison among three methods: Monte Carlo methods, quasi-Monte Carlo methods (Sobol' sequences), and high-discrepancy sequences. The results are shown in Figs. 1 and 2, where the horizontal line indicates the number of points N, which is almost proportional to the computation time. The vertical line indicates the approximated value of the integral. Figure 1 shows the result for the 50-dimensional isotropic integral with $h(x) = \cos(x)$, where we denote Monte Carlo methods by 50Dim.MC, quasi-Monte Carlo methods by 50Dim.QMC, and high-discrepancy sequences by 50Dim.HDS. Figure 2 (top) shows the result for the Kolmogorov superposition integral described in Example 1, where $a = 100\exp(0.5/365) = 100.137..$, $b = 30/\sqrt{365} = 1.57\dots$, $d = 100$, and $K = 100$. In finance words, stock price $S_0 = 100$, risk free rate $r = 0.05$, and volatility $\sigma = 0.3$. The Black-Scholes formula gives the option price to be 6.9193. Figure 2 (bottom) gives the result for the following integral:

$$\int_{[0,1]^d} \frac{dx_1 \cdots dx_d}{1 + x_1 + \cdots + x_d}. \tag{2}$$

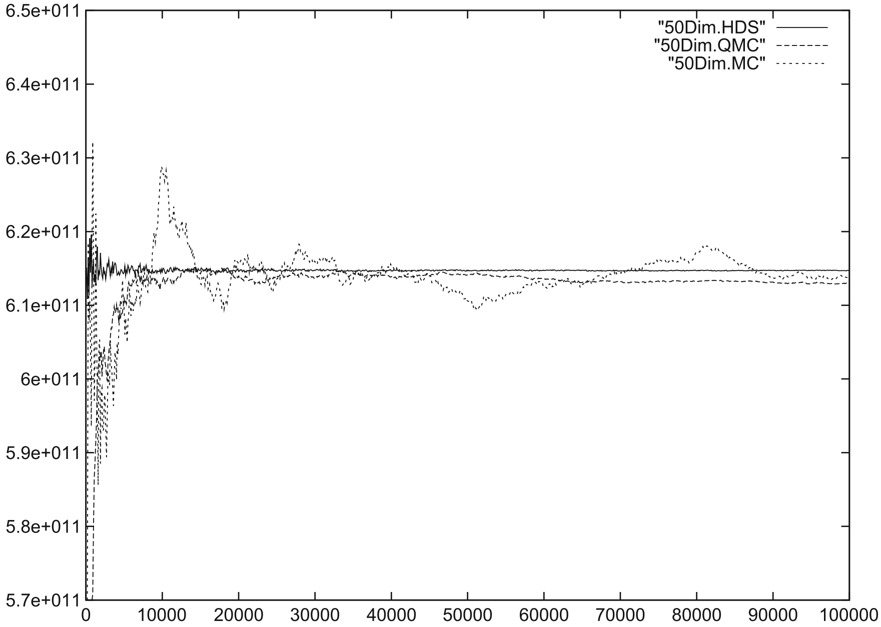


Fig. 1 Comparison of three methods for the isotropic integral with $h(x) = \cos(x)$.

This is a Kolmogorov superposition integral with $g(x) = 1/(1 + d \cdot x)$, $\Psi(x) = x$, and $a_1 = \cdots = a_d = 1/d$. In Fig. 2 we denote Monte Carlo methods by 100Dim.MC, quasi-Monte Carlo methods by 100Dim.QMC, and high-discrepancy sequences by 100Dim.HDS. In Figs. 1 and 2 (bottom) the results show that QMC and HDS are much faster to converge than MC. If we look closely at the figure for a small number of points, say, $N \leq 1,000$, then HDS looks slightly faster to become stable than QMC. Figure 2 (top) shows that QMC is as slow as MC, and much slower than HDS.

5 Conclusion

If d-dimensional low-discrepancy sequences are applied to numerical integration for the two classes of integrals discussed in this paper, then the integration error is $O(\log N)^d/N)$ according to the Koksma-Hlawka bound. On the other hand, high-discrepancy sequences give the integration error $O(\sqrt{\log N}/N)$ for the isotropic case and $O((\log N)/N)$ for the Kolmogorov superposition case. We should note that the Koksma-Hlawka bound is useless for the integration using high-discrepancy sequences because their d-dimensional discrepancy never goes to zero as the number of points approaches to the infinity.

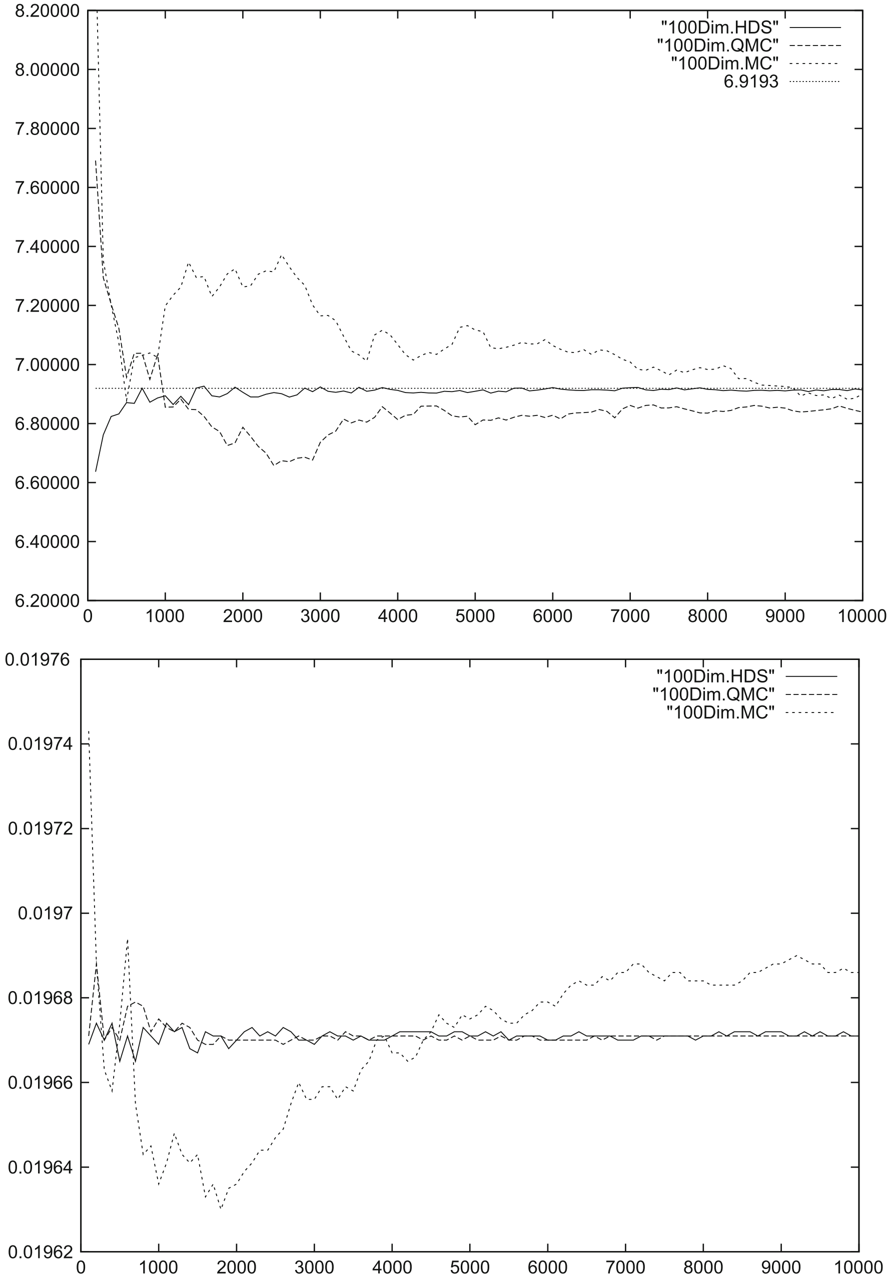


Fig. 2 Comparison of three methods for two Kolmogorov superposition integrals: the integral from Example 1 (*top*) and the integral given in (2) (*bottom*).

Acknowledgements The author thanks the anonymous referees for their valuable comments. This research was supported by KAKENHI(22540141).

References

1. J. Braun amd M. Griebel, On a Constructive Proof of Kolmogorov's Superposition Theorem, *Constructive Approximation*, **30** (2009), 653–675.
2. B. D. Keister, Multidimensional Quadrature Algorithms, *Computers in Physics*, **10**(20) (1996), 119–122.
3. C. Lemieux, *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer, 2009.
4. H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF Regional Conference Series in Applied Mathematics, No. 63, SIAM, 1992.
5. E. Novak and H. Woźniakowski, *Tractability of Multivariate Problems, Volume II: Standard Information for Functionals*, European Mathematical Society, 2010.
6. O. Strauch and Š. Porubský, *Distribution of Sequences: A Sampler*, Peter Lang, 2005.
7. A. Papageorgiou, Fast Convergence of Quasi-Monte Carlo for a Class of Isotropic Integrals, *Mathematics of Computation* **70** (2001), 297–306.
8. A. Papageorgiou, Sufficient Conditions for Fast Quasi-Monte Carlo Convergence, *Journal of Complexity*, **19** (2003), 332–351.
9. A. Papageorgiou and J. F. Traub, Faster Evaluation of Multidimensional Integrals, *Computers in Phyics*, **11**(6) (1997), 574–578.
10. S. Tezuka, *Uniform Random Numbers: Theory and Practice*, Kluwer Academic Publishers, 1995.
11. S. Tezuka, High-Discrepancy Sequences, *Kyushu Journal of Mathematics*, **61** (2007), 431–441.
12. S. Tezuka, Scrambling Non-Uniform Nets, *Mathematica Slovaca*, **59** (2009), 379–386.

Multilevel Path Simulation for Jump-Diffusion SDEs

Yuan Xia and Michael B. Giles

Abstract We investigate the extension of the multilevel Monte Carlo path simulation method to jump-diffusion SDEs. We consider models with finite rate activity using a jump-adapted discretisation in which the jump times are computed and added to the standard uniform discretisation times. The key component in multilevel analysis is the calculation of an expected payoff difference between a coarse path simulation and a fine path simulation with twice as many timesteps. If the Poisson jump rate is constant, the jump times are the same on both paths and the multilevel extension is relatively straightforward, but the implementation is more complex in the case of state-dependent jump rates for which the jump times naturally differ.

1 Introduction

In the Black-Scholes Model, the price of an option is given by the expected value of a payoff depending upon an asset price modelled by a stochastic differential equation driven by Brownian motion,

$$\mathrm{d}S(t) = a(S(t),t)\,\mathrm{d}t + b(S(t),t)\,\mathrm{d}W(t), \quad 0 \le t \le T, \tag{1}$$

with given initial data S_0. Although this model is widely used, the fact that asset returns are not log-normal has motivated people to suggest models which better capture the characteristics of the asset price dynamics. Merton [9] instead proposed a jump-diffusion process, in which the asset price follows a jump-diffusion SDE:

Y. Xia · M.B. Giles
Oxford-Man Institute of Quantitative Finance, Walton Well Road, Oxford, OX2 6ED, UK
e-mail: yuan.xia@maths.ox.ac.uk; mike.giles@maths.ox.ac.uk

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_41, © Springer-Verlag Berlin Heidelberg 2012

$$dS(t) = a(S(t-), t)\, dt + b(S(t-), t)\, dW(t) + c(S(t-), t)\, dJ(t), \quad 0 \le t \le T, \tag{2}$$

where the jump term $J(t)$ is a compound Poisson process $\sum_{i=1}^{N(t)}(Y_i - 1)$, the jump magnitude Y_i has a prescribed distribution, and $N(t)$ is a Poisson process with intensity λ, independent of the Brownian motion. Due to the existence of jumps, the process is a càdlàg process, i.e., having right continuity with left limits. We note that $S(t-)$ denotes the left limit of the process while $S(t) = \lim_{s \to t+} S(t)$. In [9], Merton also assumed that $\log Y_i$ has a normal distribution.

There are several ways in which to generalize the Merton model. Here we consider one case investigated by Glasserman and Merener [7], in which the jump rate depends on the asset price, namely $\lambda = \lambda(S(t-), t)$.

For European options, we are interested in the expected value of a function of the terminal state, $f(S(T))$, but in the case of exotic options the valuation depends on the entire path $S(t), 0 \le t \le T$. The expected value can be estimated by a simple Monte Carlo method with a suitable approximation to the SDE solution. However, if the discretisation has first order weak convergence then to achieve an $O(\epsilon)$ root mean square (RMS) error requires $O(\epsilon^{-2})$ paths, each with $O(\epsilon^{-1})$ timesteps, leading to a computational complexity of $O(\epsilon^{-3})$.

Giles [4,5] introduced a multilevel Monte Carlo path simulation method, demonstrating that the computational cost can be reduced to $O(\epsilon^{-2})$ for SDEs driven by Brownian motion. This has been extended by Dereich and Heidenreich [2, 3] to approximation methods for both finite and infinite activity Lévy-driven SDEs with globally Lipschitz payoffs. The work in this paper differs in considering simpler finite activity jump-diffusion models, but also one example of a more challenging non-Lipschitz payoff, and also uses a more accurate Milstein discretisation to achieve an improved order of convergence for the multilevel correction variance which will be defined later.

We first present the jump-adapted discretisation of jump-diffusion processes, and review the multilevel Monte Carlo method and some modifications for jump-diffusion processes. We then present the numerical algorithm in detail for the constant rate jump-diffusion model, and show numerical results for various options. The next section presents the thinning algorithm used for state-dependent intensities, and the final section draws conclusions and indicates directions for future research.

2 A Jump-Adapted Milstein Discretisation

To simulate finite activity jump-diffusion processes, we choose to use the jump-adapted approximation proposed by Platen [10]. For each path simulation, the set of jump times $\mathbb{J} = \{\tau_1, \tau_2, \ldots, \tau_m\}$ within the time interval $[0, T]$ is added to a set of uniformly spaced times $t'_i = i\,T/N,\ i = 0,\ldots,N$, to form a combined set of discretisation times $\mathbb{T} = \{0 = t_0 < t_1 < t_2 < \ldots < t_M = T\}$. As a result, the length of each timestep $h_n = t_{n+1} - t_n$ will be no greater than $h = T/N$.

Within each timestep the first order Milstein discretisation is used to approximate the SDE, and then the jump is simulated when the simulation time is equal to one of the jump times. This gives the following numerical method:

$$\widehat{S}^-_{n+1} = \widehat{S}_n + a_n\,h_n + b_n\,\Delta W_n + \tfrac{1}{2}\,b'_n\,b_n\,(\Delta W_n^2 - h_n),$$

$$\widehat{S}_{n+1} = \begin{cases} \widehat{S}^-_{n+1} + c(\widehat{S}^-_{n+1}, t_{n+1})(Y_i - 1), & \text{when } t_{n+1} = \tau_i; \\ \widehat{S}^-_{n+1}, & \text{otherwise,} \end{cases} \tag{3}$$

where the subscript n is used to denotes the timestep index, $\widehat{S}^-_n = \widehat{S}(t_n-)$ is the left limit of the approximated path, ΔW_n is the Brownian increment during the timestep, a_n, b_n, b'_n are the values of a, b, b' based on $(\widehat{S}_n, t_n)$, and Y_i is the jump magnitude at τ_i.

3 Multilevel Monte Carlo Method

For Brownian diffusion SDEs, suppose we perform Monte Carlo path simulations on different levels of resolution ℓ, with 2^ℓ uniform timesteps on level ℓ. For a given Brownian path $W(t)$, let P denote the payoff, and let $\widehat{P}_\ell$ denote its approximation by a numerical scheme with timestep h_ℓ. As a result of the linearity of the expectation operator, we have the following identity:

$$\mathbb{E}[\widehat{P}_L] = \mathbb{E}[\widehat{P}_0] + \sum_{\ell=1}^{L} \mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]. \tag{4}$$

Let $\widehat{Y}_0$ denote the standard Monte Carlo estimate for $\mathbb{E}[\widehat{P}_0]$ using N_0 paths, and for $\ell > 0$, we use N_ℓ independent paths to estimate $\mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]$ using

$$\widehat{Y}_\ell = N_\ell^{-1} \sum_{i=1}^{N_\ell} \left(\widehat{P}^{(i)}_\ell - \widehat{P}^{(i)}_{\ell-1} \right). \tag{5}$$

The multilevel method exploits the fact that $V_\ell := \mathbb{V}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]$ decreases with ℓ, and adaptively chooses N_ℓ to minimise the computational cost to achieve a desired root-mean-square error. This is summarized in the following theorem:

Theorem 1. *Let P denote a functional of the solution of stochastic differential equation (1) for a given Brownian path $W(t)$, and let $\widehat{P}_\ell$ denote the corresponding approximation using a numerical discretisation with timestep $h_\ell = 2^{-\ell}\, T$.*

If there exist independent estimators $\widehat{Y}_\ell$ based on N_ℓ Monte Carlo samples, and positive constants $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ such that

(i) $\left|\mathbb{E}[\widehat{P}_\ell - P]\right| \leq c_1\, h_\ell^\alpha$

(ii) $\mathbb{E}[\widehat{Y}_\ell] = \begin{cases} \mathbb{E}[\widehat{P}_0], & l = 0 \\ \mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}], & l > 0 \end{cases}$

(iii) $\mathbb{V}[\widehat{Y}_\ell] \leq c_2\, N_\ell^{-1} h_\ell^\beta$

(iv) *C_ℓ, the computational complexity of $\widehat{Y}_\ell$, is bounded by*

$$C_\ell \leq c_3\, N_\ell\, h_\ell^{-1},$$

then there exists a positive constant c_4 such that for any $\epsilon < e^{-1}$ there are values L and N_ℓ for which the multilevel estimator

$$\widehat{Y} = \sum_{\ell=0}^{L} \widehat{Y}_\ell,$$

has a mean-square-error with bound

$$MSE \equiv \mathbb{E}\left[\left(\widehat{Y} - \mathbb{E}[P]\right)^2\right] < \epsilon^2$$

with a computational complexity C with bound

$$C \leq \begin{cases} c_4\, \epsilon^{-2}, & \beta > 1, \\ c_4\, \epsilon^{-2} (\log \epsilon)^2, & \beta = 1, \\ c_4\, \epsilon^{-2-(1-\beta)/\alpha}, & 0 < \beta < 1. \end{cases}$$

Proof. See [5].

In the case of the jump-adapted discretisation, h_ℓ should be taken to be the uniform timestep at level ℓ, to which the jump times are added to form the set of discretisation times. We have to define the computational complexity as the expected computational cost since different paths may have different numbers of jumps. However, the expected number of jumps is finite and therefore the cost bound in assumption (iv) will still remain valid for an appropriate choice of the constant c_3.

4 Multilevel Monte Carlo for Constant Jump Rate

The Multilevel Monte Carlo approach for a constant jump rate is straightforward. The jump times τ_j, which are the same for the coarse and fine paths, are simulated by setting $\tau_j - \tau_{j-1} \sim \exp(\lambda)$. The Brownian increments ΔW_n are generated for the fine path, and then summed appropriately to generate the increments for the coarse path. In the following we show numerical results for European call, lookback and barrier options. Asian and digital options have also been simulated; numerical results for these are available in [12] along with more details of the construction of the multilevel estimators for the path-dependent payoffs.

All of the options are priced for the Merton model in which the jump-diffusion SDE under the risk-neutral measure is

$$\frac{\mathrm{d}S(t)}{S(t-)} = (r - \lambda m)\,\mathrm{d}t + \sigma\,\mathrm{d}W(t) + \mathrm{d}J(t), \quad 0 \le t \le T,$$

where λ is the jump intensity, r is the risk-free interest rate, σ is the volatility, the jump magnitude satisfies $\log Y_i \sim N(a,b)$, and $m = \mathbb{E}[Y_i] - 1$ is the compensator to ensure the discounted asset price is a martingale. All of the simulations in this section use the parameter values $S_0 = 100$, $K = 100$, $T = 1$, $r = 0.05$, $\sigma = 0.2$, $a = 0.1$, $b = 0.2$, $\lambda = 1$.

4.1 European Call Option

Figure 1 shows the numerical results for the European call option with payoff $\exp(-rT)\,(S(T)-K)^+$, with $(x)^+ \equiv \max(x,0)$ and strike $K = 100$.

The top left plot shows the behaviour of the variance of both $\widehat{P}_\ell$ and the multilevel correction $\widehat{P}_\ell - \widehat{P}_{\ell-1}$, estimated using 10^5 samples so that the Monte Carlo sampling error is negligible. The slope of the MLMC line indicates that $V_\ell \equiv \mathbb{V}[\widehat{P}_\ell - \widehat{P}_{\ell-1}] = O(h_\ell^2)$, corresponding to $\beta = 2$ in condition (iii) of Theorem 1. The top right plot shows that $\mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}]$ is approximately $O(h_\ell)$, corresponding to $\alpha = 1$ in condition (i). Noting that the payoff is Lipschitz, both of these are consistent with the first order strong convergence proved in [11].

The bottom two plots correspond to five different multilevel calculations with different user-specified accuracies to be achieved. These use the numerical algorithm given in [5] to determine the number of grid levels, and the optimal number of samples on each level, which are required to achieve the desired accuracy. The left plot shows that in each case many more samples are used on level 0 than on any other level, with very few samples used on the finest level of resolution. The right plot shows that the the multilevel cost is approximately proportional to ϵ^{-2}, which agrees with the computational complexity bound in Theorem 1 for the $\beta > 1$ case.

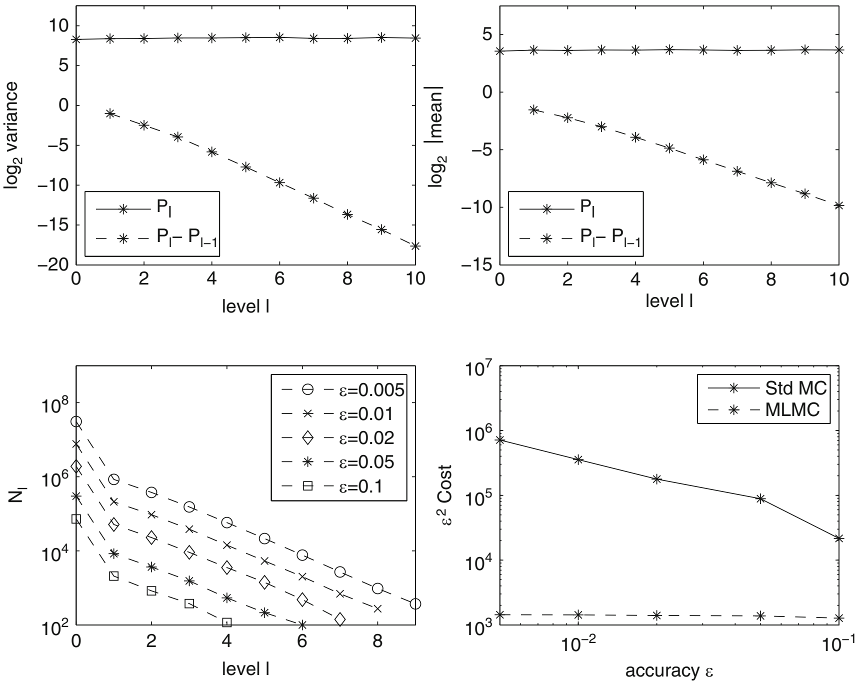


Fig. 1 European call option with constant Poisson rate.

4.2 Lookback Option

The payoff of the lookback option we consider is

$$P = \exp(-rT) \left(S(T) - \min_{0 \le t \le T} S(t) \right).$$

Previous work [4] achieved a second order convergence rate for the multilevel correction variance using the Milstein discretisation and an estimator constructed by approximating the behaviour within a timestep as an Itô process with constant drift and volatility, conditional on the endpoint values $\widehat{S}_n$ and $\widehat{S}_{n+1}$. Brownian Bridge results (see Sect. 6.4 in [6]) give the minimum value within the timestep $[t_n, t_{n+1}]$, conditional on the end values, as

$$\widehat{S}_{n,min} = \tfrac{1}{2} \left(\widehat{S}_n + \widehat{S}_{n+1} - \sqrt{\left(\widehat{S}_{n+1} - \widehat{S}_n\right)^2 - 2\, b_n^2\, h \log U_n} \right), \qquad (6)$$

where b_n is the constant volatility and U_n is a uniform random variable on $[0, 1]$. The same treatment can be used for the jump-adapted discretisation in this paper, except that $\widehat{S}_{n+1}^{-}$ must be used in place of $\widehat{S}_{n+1}$ in (6).

Equation 6 is used for the fine path approximation, but a different treatment is used for the coarse path, as in [4]. This involves a change to the original telescoping sum in (4) which now becomes

$$\mathbb{E}[\widehat{P}_L^f] = \mathbb{E}[\widehat{P}_0^f] + \sum_{\ell=1}^{L} \mathbb{E}[\widehat{P}_\ell^f - \widehat{P}_{\ell-1}^c], \tag{7}$$

where $\widehat{P}_\ell^f$ is the approximation on level ℓ when it is the finer of the two levels being considered, and $\widehat{P}_\ell^c$ is the approximation when it is the coarser of the two. This modified telescoping sum remains valid provided $\mathbb{E}[\widehat{P}_\ell^f] = \mathbb{E}[\widehat{P}_\ell^c]$.

Considering a particular timestep in the coarse path construction, we have two possible situations. If it does not contain one of the fine path discretisation times, and therefore corresponds exactly to one of the fine path timesteps, then it is treated in the same way as the fine path, using the same uniform random number U_n. This leads naturally to a very small difference in the respective minima for the two paths.

The more complicated case is the one in which the coarse timestep contains one of the fine path discretisation times t', and so corresponds to the union of two fine path timesteps. In this case, the value at time t' is given by the conditional Brownian interpolant

$$\widehat{S}(t') = \widehat{S}_n + \mu\,(\widehat{S}_{n+1}^{-} - \widehat{S}_n) + b_n\left(W(t') - W_n - \mu\,(W_{n+1} - W_n)\right), \tag{8}$$

where $\mu = (t' - t_n)/(t_{n+1} - t_n)$ and the value of $W(t')$ comes from the fine path simulation. Given this value for $\widehat{S}(t')$, the minimum values for $S(t)$ within the two intervals $[t_n, t']$ and $[t', t_{n+1}]$ can be simulated in the same way as before, using the same uniform random numbers as the two fine timesteps.

The equality $\mathbb{E}[\widehat{P}_\ell^f] = \mathbb{E}[\widehat{P}_\ell^c]$ is respected in this treatment because $W(t')$ comes from the correct distribution, conditional on W_{n+1}, W_n, and therefore, conditional on the values of the Brownian path at the set of coarse discretisation points, the computed value for the coarse path minimum has exactly the same distribution as it would have if the fine path algorithm were applied.

Further discussion and analysis of this is given in [13], including a proof that the strong error between the analytic path and the conditional interpolation approximation is at worst $O(h\ \log h)$.

Figure 2 presents the numerical results. The results are very similar to those obtained by Giles for geometric Brownian motion [4]. The top two plots indicate second order variance convergence rate and first order weak convergence, both of which are consistent with the $O(h\ \log h)$ strong convergence. The computational cost of the multilevel method is therefore proportional to ϵ^{-2}, as shown in the bottom right plot.

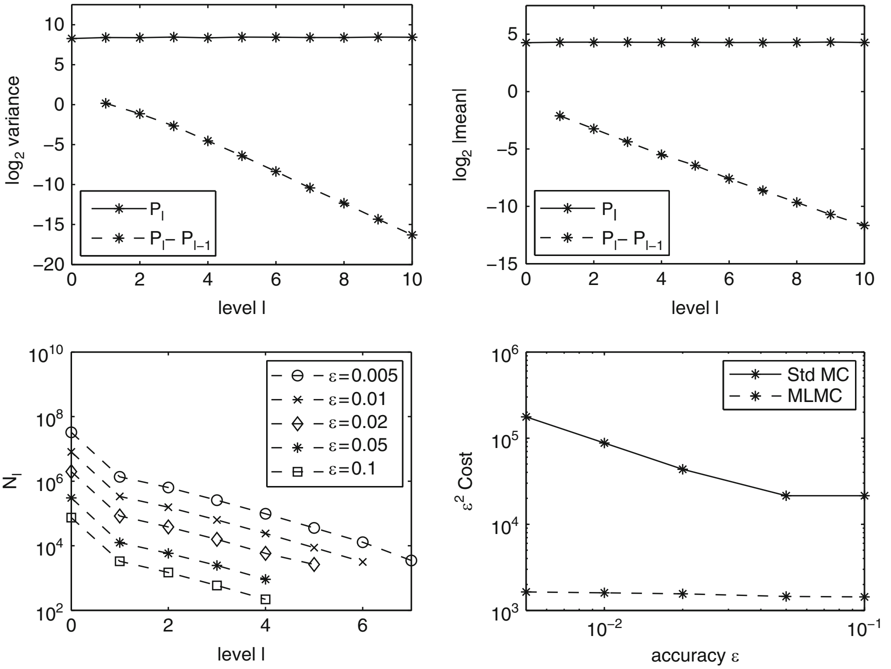


Fig. 2 Lookback option with constant Poisson rate.

4.3 Barrier Option

We consider a down-and-out call barrier option for which the discounted payoff is

$$P = \exp(-rT)\,(S(T) - K)^{+} \quad {}_{\{M_T > B\}},$$

where $M_T = \min_{0 \le t \le T} S(t)$. The jump-adapted Milstein discretisation with the Brownian interpolation gives the approximation

$$\widehat{P} = \exp(-rT)\,(\widehat{S}(T) - K)^{+} \quad {}_{\left\{\widehat{M}_T > B\right\}}$$

where $\widehat{M}_T = \min_{0 \le t \le T} \widehat{S}(t)$. This could be simulated in exactly the same way as the lookback option, but in this case the payoff is a discontinuous function of the minimum M_T and an $O(h)$ error in approximating M_T would lead to an $O(h)$ variance for the multilevel correction.

Instead, following the approach of Cont and Tankov (see p. 177 in [1]), it is better to use the expected value conditional on the values of the discrete Brownian increments and the jump times and magnitudes, all of which may be represented collectively as $\mathscr{F}$. This yields

$$\begin{aligned}
&\mathbb{E}\left[\exp(-rT)\,(\widehat{S}(T)-K)^{+}\ {}_{\{\widehat{M}_T>B\}}\right]\\
&\qquad=\mathbb{E}\left[\exp(-rT)\,(\widehat{S}(T)-K)^{+}\mathbb{E}\left[\ {}_{\{\widehat{M}_T>B\}}\mid\mathscr{F}\right]\right]\\
&\qquad=\mathbb{E}\left[\exp(-rT)\,(\widehat{S}(T)-K)^{+}\prod_{n=0}^{n_T-1}\widehat{p}_n\right]
\end{aligned}$$

where n_T is the number of timesteps, and $\widehat{p}_n$ denotes the conditional probability that the path does not cross the barrier B during the n^{th} timestep:

$$\widehat{p}_n = 1-\exp\left(\frac{-2\,(\widehat{S}_n-B)^{+}(\widehat{S}^{-}_{n+1}-B)^{+}}{b_n^2\,(t_{n+1}-t_n)}\right). \tag{9}$$

This barrier crossing probability is computed through conditional expectation and can be used to deduce (6).

For the coarse path calculation, we again deal separately with two cases. When the coarse timestep does not include a fine path time, then we again use (9). In the other case, when it includes a fine path time t' we evaluate the Brownian interpolant at t' and then use the conditional expectation to obtain

$$\begin{aligned}
\widehat{p}_n = &\left\{1-\exp\left(\frac{-2\,(\widehat{S}_n-B)^{+}(\widehat{S}(t')-B)^{+}}{b_n^2\,(t'-t_n)}\right)\right\}\\
&\times\left\{1-\exp\left(\frac{-2\,(\widehat{S}(t')-B)^{+}(\widehat{S}^{-}_{n+1}-B)^{+}}{b_n^2\,(t_{n+1}-t')}\right)\right\}.
\end{aligned}\tag{10}$$

Figure 3 shows the numerical results for $K=100$, $B=85$. The top left plot shows that the multilevel variance is $O(h_\ell^{\beta})$ for $\beta\approx 3/2$. This is similar to the behavior for a diffusion process [4]. The bottom right plot shows that the computational cost of the multilevel method is again almost perfectly proportional to ϵ^{-2}.

5 Path-Dependent Rates

In the case of a path-dependent jump rate $\lambda(S_t,t)$, the implementation of the multilevel method becomes more difficult because the coarse and fine path approximations may jump at different times. These differences could lead to a large difference between the coarse and fine path payoffs, and hence greatly increase the variance of the multilevel correction. To avoid this, we modify the simulation approach of Glasserman and Merener [7] which uses "thinning" to treat the case in which λ is bounded.

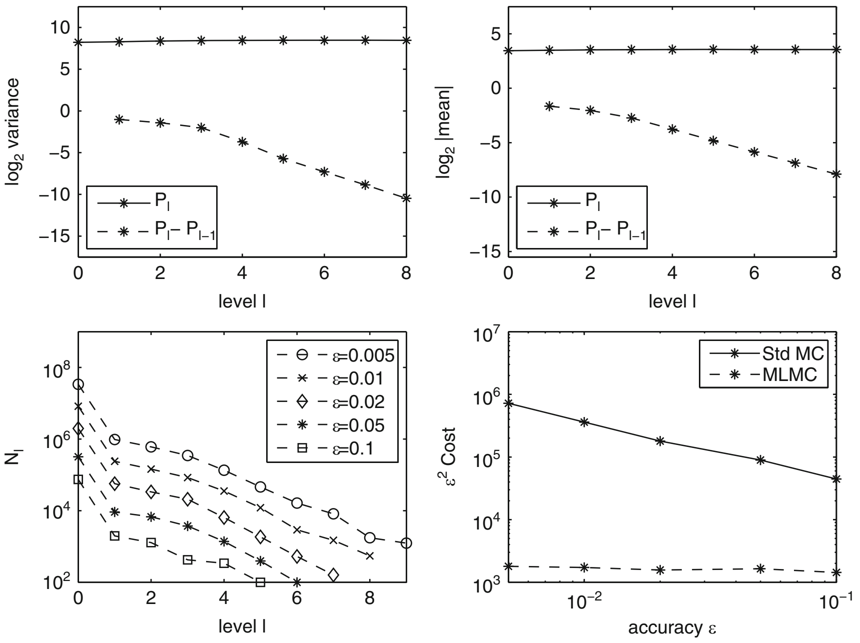


Fig. 3 Barrier option with constant Poisson rate.

The idea of the thinning method is to construct a Poisson process with a constant rate λ_{sup} which is an upper bound of the state-dependent rate. This gives a set of candidate jump times, and these are then selected as true jump times with probability $\lambda(S_t, t)/\lambda_{\text{sup}}$. Hence we have the following jump-adapted thinning Milstein scheme:

1. Generate the jump-adapted time grid for a Poisson process with constant rate λ_{sup};
2. Simulate each timestep using the Milstein discretisation;
3. When the endpoint t_{n+1} is a candidate jump time, generate a uniform random number $U \sim [0, 1]$, and if $U < p_{t_{n+1}} = \dfrac{\lambda(S(t_{n+1}-), t_{n+1})}{\lambda_{\text{sup}}}$, then accept t_{n+1} as a real jump time and simulate the jump.

5.1 Multilevel Treatment

In the multilevel implementation, if we use the above algorithm with different acceptance probabilities for fine and coarse level, there may be some samples in which a jump candidate is accepted for the fine path, but not for the coarse path, or vice versa. Because of first order strong convergence, the difference in acceptance probabilities will be $O(h)$, and hence there is an $O(h)$ probability of coarse and fine paths differing in accepting candidate jumps. Such differences will give an $O(1)$

difference in the payoff value, and hence the multilevel variance will be $O(h)$. A more detailed analysis of this is given in [13].

To improve the variance convergence rate, we use a change of measure so that the acceptance probability is the same for both fine and coarse paths. This is achieved by taking the expectation with respect to a new measure Q:

$$\mathbb{E}[\widehat{P}_\ell - \widehat{P}_{\ell-1}] = \mathbb{E}_Q[\widehat{P}_\ell \prod_\tau R^f_\tau - \widehat{P}_{\ell-1} \prod_\tau R^c_\tau]$$

where τ are the jump times. The acceptance probability for a candidate jump under the measure Q is defined to be $\frac{1}{2}$ for both coarse and fine paths, instead of $p_\tau = \lambda(S(\tau-), \tau) \,/\, \lambda_{\sup}$. The corresponding Radon-Nikodym derivatives are

$$R^f_\tau = \begin{cases} 2p^f_\tau, & \text{if } U < \dfrac{1}{2}\,; \\ 2(1-p^f_\tau), & \text{if } U \geq \dfrac{1}{2}\,, \end{cases} \qquad R^c_\tau = \begin{cases} 2p^c_\tau, & \text{if } U < \dfrac{1}{2}\,; \\ 2(1-p^c_\tau), & \text{if } U \geq \dfrac{1}{2}\,, \end{cases}$$

Since $R^f_\tau - R^c_\tau = O(h)$ and $\widehat{P}_\ell - \widehat{P}_{\ell-1} = O(h)$, this results in the multilevel correction variance $\mathbb{V}_Q[\widehat{P}_\ell \prod_\tau R^f_\tau - \widehat{P}_{\ell-1} \prod_\tau R^c_\tau]$ being $O(h^2)$.

If the analytic formulation is expressed using the same thinning and change of measure, the weak error can be decomposed into two terms as follows:

$$\mathbb{E}_Q\left[\widehat{P}_\ell \prod_\tau R^f_\tau - P \prod_\tau R_\tau\right] = \mathbb{E}_Q\left[(\widehat{P}_\ell - P) \prod_\tau R^f_\tau\right] + \mathbb{E}_Q\left[P\,(\prod_\tau R^f_\tau - \prod_\tau R_\tau)\right].$$

Using Hölder's inequality, the bound $\max(R_\tau, R^f_\tau) \leq 2$ and standard results for a Poisson process, the first term can be bounded using weak convergence results for the constant rate process, and the second term can be bounded using the corresponding strong convergence results [13]. This guarantees that the multilevel procedure does converge to the correct value.

5.1.1 Numerical Results

We show numerical results for a European call option using

$$\lambda = \frac{1}{1+(S(t-)/S_0)^2}, \quad \lambda_{\sup} = 1,$$

and with all other parameters as used previously for the constant rate cases.

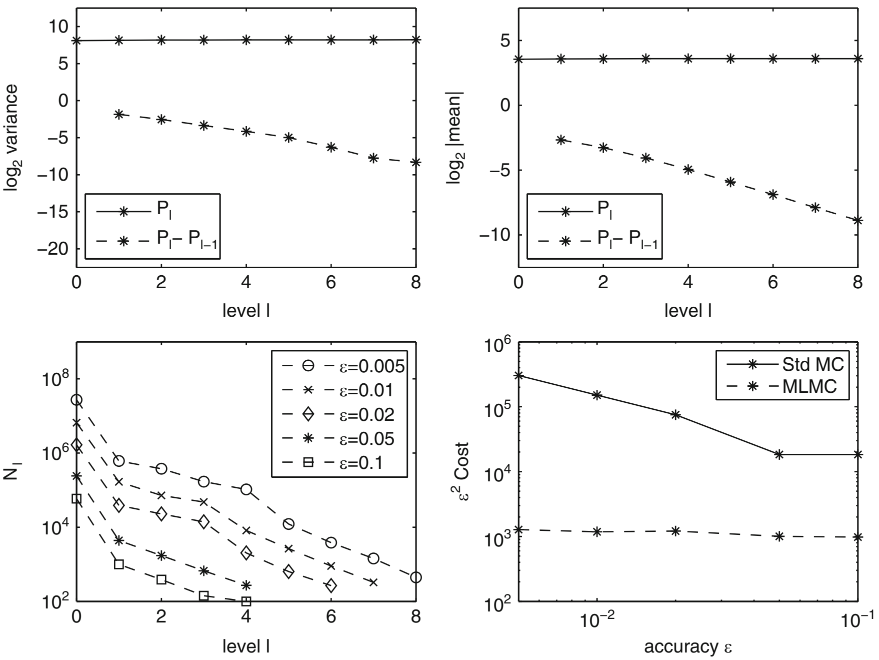


Fig. 4 European call option with path-dependent Poisson rate using thinning without a change of measure.

Comparing Figs. 4 and 5 we see that the variance convergence rate is significantly improved by the change of measure, but there is little change in the computational cost. This is due to the main computational effort being on the coarsest level, which suggests using quasi-Monte Carlo on that level [8].

The bottom left plot in Fig. 4 shows a slightly erratic behaviour. This is because the $O(h_\ell)$ variance is due to a small fraction of the paths having an $O(1)$ value for $\widehat{P}_\ell - \widehat{P}_{\ell-1}$. In the numerical procedure, the variance is estimated using an initial sample of 100 paths. When the variance is dominated by a few outliers, this sample size is not sufficient to provide an accurate estimate, leading to this variability.

6 Conclusions and Future Work

In this work we have extended the multilevel Monte Carlo method to scalar jump-diffusion SDEs using a jump-adapted discretisation. Second order variance convergence is maintained in the constant rate case for European options with Lipschitz payoffs, and also for lookback options by constructing estimators using a previous Brownian interpolation technique. Variance convergence of order 1.5 is obtained for barrier and digital options, which again matches the convergence which

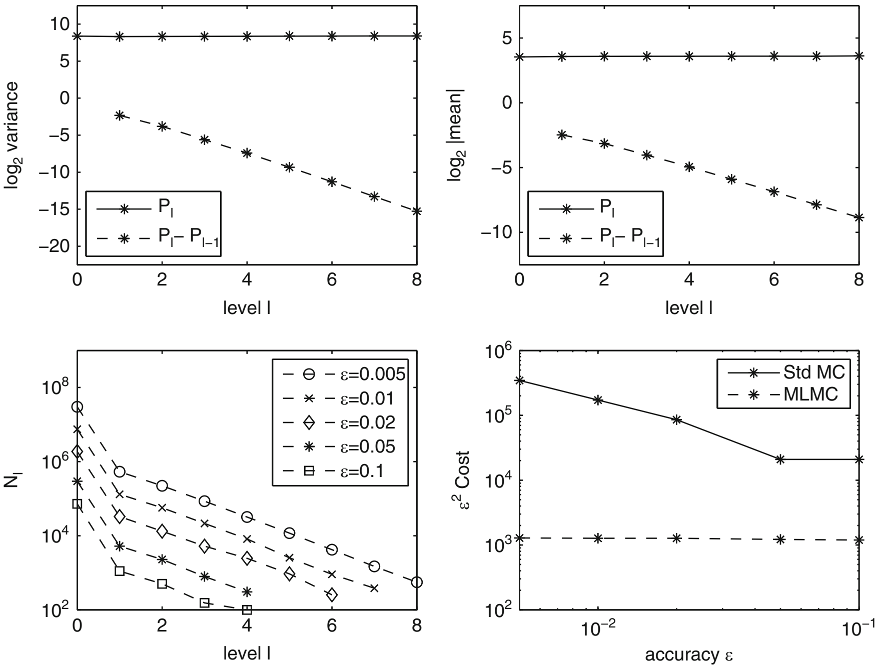


Fig. 5 European call option with path-dependent Poisson rate using thinning with a change of measure.

has been achieved previously for scalar SDEs without jumps. In the state-dependent rate case, we use thinning with a change of measure to avoid asynchronous jumps in the fine and coarse levels. In separate work [12] we have also investigated an alternative approach using a time-change Poisson process to handle cases in which there is no upper bound on the jump rate.

The first natural direction for future work is numerical analysis to determine the order of convergence of multilevel correction variance [13]. A second is to investigate other Lévy processes, such as VG (Variance-Gamma), and NIG (Normal Inverse Gaussian). We also plan to investigate whether the multilevel quasi-Monte Carlo method [8] will further reduce the cost.

Acknowledgements Xia is grateful to the China Scholarship Council for financial support, and the research has also been supported by the Oxford-Man Institute of Quantitative Finance. The authors are indebted to two anonymous reviewers for their invaluable comments.

References

1. R. Cont and P. Tankov. *Financial modelling with jump processes*. Chapman & Hall, 2004.
2. S. Dereich. Multilevel Monte Carlo algorithms for Lévy-driven SDEs with Gaussian correction. *The Annals of Applied Probability*, 21(1):283–311, 2011.

3. S. Dereich and F. Heidenreich. A multilevel Monte Carlo algorithm for Lévy driven stochastic differential equations. *Stochastic Processes and their Applications*, 121(7):1565–1587, 2011.
4. M.B. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 343–358. Springer-Verlag, 2007.
5. M.B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
6. P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.
7. P. Glasserman and N. Merener. Convergence of a discretization scheme for jump-diffusion processes with state-dependent intensities. *Proc. Royal Soc. London A*, 460:111–127, 2004.
8. M.B. Giles and B.J. Waterhouse. Multilevel quasi-Monte Carlo path simulation. *Radon Series Comp. Appl. Math.*, **8**, 1–18, 2009.
9. R.C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2):125–144, 1976.
10. E. Platen. A generalized Taylor formula for solutions of stochastic equations. *Sankhyā: The Indian Journal of Statistics, Series A*, 44(2):163–172, 1982.
11. N. Platen and E. Bruti-Liberati. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*, volume 64 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, 1st edition, 2010.
12. Y. Xia. Multilevel Monte Carlo method for jump-diffusion SDEs. Technical report. http://arxiv.org/abs/1106.4730
13. Y. Xia and M.B. Giles. Numerical analysis of multilevel Monte Carlo for scalar jump-diffusion SDEs. Working paper in preparation, 2011.

Randomized Algorithms for Hamiltonian Simulation

Chi Zhang

Abstract We consider *randomized* algorithms for simulating the evolution of a Hamiltonian $H = \sum_{j=1}^{m} H_j$ for time t. The evolution is simulated by a product of exponentials of H_j in a random sequence, and random evolution times. Hence the final state of the system is approximated by a mixed quantum state. First we provide a scheme to bound the error of the final quantum state in a randomized algorithm. Then we obtain randomized algorithms which have the same efficiency as certain deterministic algorithms but which are simpler to implement.

1 Introduction

Simulation of quantum systems is one of the main applications of quantum computing. While the computational cost of simulating many particle quantum systems using classical computers grows exponentially with the number of particles, quantum computers have the potential to carry out the simulation efficiently. This property, pointed out by Feynman, is one of the founding ideas of the field of quantum computing [1]. In addition to predicting and simulating the behavior of physical and chemical systems [1–5], the simulation problem also has other applications such as unstructured search, adiabatic optimization, quantum walks, and the NAND tree evaluation algorithms [6–13].

In a Hamiltonian simulation problem, the goal is to simulate the unitary operator e^{-iHt}, for some given time-independent Hamiltonian H and evolution time t. The accuracy ε of the simulation is measured by the trace distance [14] between the simulated final state and the desired final state.

C. Zhang
Department of Computer Science, Columbia University, New York, USA, 10027
e-mail: czhang@cs.columbia.edu

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4_42, © Springer-Verlag Berlin Heidelberg 2012

Of particular interest are *splitting methods* that approximate the unitary evolution $U = e^{-iHt}$ by a product of exponentials of H_{j_s} for some sequence of j_s and intervals t_s, i.e., $\hat{U} = \prod_{s=1}^{N} e^{-iH_{j_s}t_s}$, where $H = \sum_{j=1}^{m} H_j$ and assuming H_j can be implemented efficiently. Without loss of generality, throughout this paper, the norm of H, $||H||$, is assumed to be constant, since the evolution of any Hamiltonian H for time t can be reduced to the evolution of $H/||H||$ for time $||H||t$. The cost of a quantum algorithm is measured by the number of exponentials of $H_1, \cdots, H_m$.

Various deterministic algorithms for this problem have been proposed in the literature, see, e.g., [4, 14–17], including algorithms based on the Trotter formula [14], the Strang splitting formula [15] and Suzuki's decompositions[16, 17]. Consider for an example, the algorithm based on the Trotter formula. Let ε be the error bound. First, the evolution time t is divided into $K = O(t^2/\varepsilon)$ small intervals of size $\Delta t = t/K$. Then, each $e^{-iH\Delta t}$ is simulated by $\prod_{j=1}^{m} e^{-iH_j\Delta t}$. From the Trotter formula, the increase of the trace distance is at most $O(\Delta t^2)$ in each interval, hence the error in the final state is guaranteed not to exceed ε. Moreover, high order splitting methods [4, 16, 17] can be used to derive asymptotically tight bounds for the number of required exponentials. However, as far as we know only deterministic algorithms have been considered.

In this paper, we consider *randomized* algorithms for Hamiltonian simulation. By randomized we mean algorithms simulating $U = e^{-iHt}$ by $\hat{U}_\omega = \prod_{s=1}^{N_\omega} e^{-iH_{j_{s,\omega}}t_{s,\omega}}$ for random sequences of $j_{s,\omega}$ and intervals $t_{s,\omega}$, occurring with probability p_ω. Consequently, the final state is approximated by a mixed quantum state. We show that:

1. Consider a randomized algorithm, where the unitary operator U is simulated by $\hat{U}_\omega$ with probability p_ω. Assume the initial and final states for U are ρ_{init} and ρ_{final}, and the initial and final state of the simulation process are $\tilde{\rho}_{\text{init}}$ and $\tilde{\rho}_{\text{final}}$, respectively. Then

$$D(\rho_{\text{final}}, \tilde{\rho}_{\text{final}}) \le D(\rho_{\text{init}}, \tilde{\rho}_{\text{init}}) + 2||E(\hat{U}_\omega) - U|| + E(||\hat{U}_\omega - U||^2)$$

   where $D(\cdot)$ be the trace distance and $E(\cdot)$ denotes the expectation.
2. There are randomized algorithms which are easier to implement than deterministic algorithms having the same accuracy and efficiency.

2 Random Models for Hamiltonian Simulation

Let us now state the problem in more detail and then discuss the algorithms and their performance. A quantum system evolves according to the Schrödinger equation

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle,$$

where H is the system Hamiltonian. For a time-independent H, the solution of the Schrödinger is $|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle$, where $|\psi_0\rangle$ is the initial state at $t = 0$. Here we assume that H is the sum of local Hamiltonians, i.e.,

$$H = \sum_{j=1}^{m} H_j, \tag{1}$$

and all the H_j are such that $e^{-iH_j\tau}$ can be implemented efficiently for any τ. We will use a product of exponentials of H_j for some sequence of j_s and intervals t_s, i.e., $\hat{U} = \prod_{s=1}^{N} e^{-iH_{j_s}t_s}$, to simulate $U = e^{-iHt}$. However, since the H_j do not commute in general, this introduces an error in the simulation. We measure this error using the trace distance, as in [4]. The trace distance between quantum states ρ and σ is

$$D(\rho, \sigma) \equiv \frac{1}{2}\mathrm{Tr}|\rho - \sigma|,$$

where $|A| \equiv \sqrt{A^\dagger A}$ is the positive square root of $A^\dagger A$ [14]. Our goal is to obtain tight bounds on N for algorithms achieving accuracy ε in the simulation.

In the randomized model, the sequence of unitary operators is selected randomly according to a certain probability distribution. The distribution can be realized either by "coin-flips" or by "control qubits". As a result, the algorithm is a product of a random sequence of unitary operators $\hat{U}_\omega = \prod_{s=1}^{N_\omega} e^{-iH_{j_{s,\omega}}t_{s,\omega}}$ selected with probability p_ω. Hence, for initial state $\rho_{\text{init}} = |\psi_0\rangle\langle\psi_0|$, the final state of the quantum algorithm is a mixed state $\sum_\omega p_\omega \hat{U}_\omega \rho_{\text{init}} \hat{U}_\omega^\dagger$. For more general cases, where the initial state of the simulation is not exactly ρ_{init}, but is a different mixed state $\tilde{\rho}_{\text{init}}$, the final state becomes

$$\tilde{\rho}_{\text{final}} = \sum_\omega p_\omega \hat{U}_\omega \tilde{\rho}_{\text{init}} \hat{U}_\omega^\dagger.$$

We now obtain an upper bound for the trace distance between the desired state and the one computed by a randomized algorithm.

Theorem 1. *Let U be the unitary operator being simulated by a set of random unitary operators $\{\hat{U}_\omega\}$ with probability distribution $\{p_\omega\}$. Assume the initial state for U is ρ_{init}, and the final state is $\rho_{\text{final}} = U\rho_{\text{init}}U^\dagger$. While the initial state of the simulation process is $\tilde{\rho}_{\text{init}}$, and the final state is $\tilde{\rho}_{\text{final}}$. Then, the trace distance between ρ_{final} and $\tilde{\rho}_{\text{final}}$ is bounded from above by*

$$D(\rho_{\text{final}}, \tilde{\rho}_{\text{final}}) \leq D(\rho_{\text{init}}, \tilde{\rho}_{\text{init}}) + 2||E(\hat{U}_\omega) - U|| + E(||\hat{U}_\omega - U||^2), \tag{2}$$

where $D(\cdot)$ denotes the trace distance, $E(\cdot)$ denotes the expectation, and $||\cdot||$ is the 2-norm.

Proof. First, we calculate the difference between ρ_{final} and $\tilde{\rho}_{\text{final}}$, which is

$$\tilde{\rho}_{\text{final}} - \rho_{\text{final}} = \sum_{\omega} p_{\omega} \hat{U}_{\omega} \tilde{\rho}_{\text{init}} \hat{U}_{\omega}^{\dagger} - U \rho_{\text{init}} U^{\dagger}$$

$$= \sum_{\omega} p_{\omega} (U + \hat{U}_{\omega} - U) \tilde{\rho}_{\text{init}} (U + \hat{U}_{\omega} - U)^{\dagger} - U \rho_{\text{init}} U^{\dagger}$$

$$= \sum_{\omega} p_{\omega} (\hat{U}_{\omega} - U) \tilde{\rho}_{\text{init}} U^{\dagger} + \sum_{\omega} p_{\omega} U \tilde{\rho}_{\text{init}} (\hat{U}_{\omega} - U)^{\dagger}$$

$$+ \sum_{\omega} p_{\omega} (\hat{U}_{\omega} - U) \tilde{\rho}_{\text{init}} (\hat{U}_{\omega} - U)^{\dagger} + \sum_{\omega} p_{\omega} U \tilde{\rho}_{\text{init}} U^{\dagger} - U \rho_{\text{init}} U^{\dagger}$$

$$= \left(\sum_{\omega} p_{\omega} \hat{U}_{\omega} - U \right) \tilde{\rho}_{\text{init}} U^{\dagger} + U \tilde{\rho}_{\text{init}} \left(\sum_{\omega} p_{\omega} \hat{U}_{\omega} - U \right)^{\dagger}$$

$$+ \sum_{\omega} p_{\omega} (\hat{U}_{\omega} - U) \tilde{\rho}_{\text{init}} (\hat{U}_{\omega} - U)^{\dagger} + U (\tilde{\rho}_{\text{init}} - \rho_{\text{init}}) U^{\dagger}.$$

Hence,

$$D(\tilde{\rho}_{\text{final}}, \rho_{\text{final}}) = \text{Tr} |\tilde{\rho}_{\text{final}} - \rho_{\text{final}}|$$

$$\leq \text{Tr} | \left(\sum_{\omega} p_{\omega} \hat{U}_{\omega} - U \right) \tilde{\rho}_{\text{init}} U^{\dagger} | + \text{Tr} | U \tilde{\rho}_{\text{init}} \left(\sum_{\omega} p_{\omega} \hat{U}_{\omega} - U \right)^{\dagger} |$$

$$+ \sum_{\omega} p_{\omega} \text{Tr} | (\hat{U}_{\omega} - U) \tilde{\rho}_{\text{init}} (\hat{U}_{\omega} - U)^{\dagger} | + \text{Tr} | \tilde{\rho}_{\text{init}} - \rho_{\text{init}} |$$

$$\leq 2 || \sum_{\omega} p_{\omega} \hat{U}_{\omega} - U || + \sum_{\omega} p_{\omega} || \hat{U}_{\omega} - U ||^2 + D(\tilde{\rho}_{\text{init}}, \rho_{\text{init}})$$

$$= D(\tilde{\rho}_{\text{init}}, \rho_{\text{init}}) + 2 || E(\hat{U}_{\omega}) - U || + E(|| \hat{U}_{\omega} - U ||^2).$$

□

Similarly to deterministic splitting algorithms, in a randomized algorithm the evolution time t is divided into K small intervals of size $\Delta t = t/K$, where K is decided later. The algorithm is comprised of K stages, and in each stage it approximates $e^{-iH\Delta t}$ by a product of unitary operators selected randomly according to a ceratin probability distribution. More precisely, in the k-th stage, the initial state is $\tilde{\rho}_{k-1}$, and the algorithm selects a product of exponentials randomly according to a certain probability distribution $\{p_{\omega}\}$ from $\{\hat{U}_{\omega} = \prod_{s=1}^{n_{\omega}} e^{-iH_{j_{s,\omega}} \Delta t_{s,\omega}}\}$. Then, the final state of the k-th stage is

$$\tilde{\rho}_k = \sum_{\omega} p_{\omega} \hat{U}_{\omega} \tilde{\rho}_{k-1} \hat{U}_{\omega}^{\dagger}.$$

Assume that $\tilde{\rho}_{\text{init}} = \rho_{\text{init}} = |\psi(0)\rangle\langle\psi(0)|$. The final state of the algorithm, $\tilde{\rho}_K$, is used to approximate $\rho_{\text{final}} = |\psi(t)\rangle\langle\psi(t)|$. Then, by choosing different unitary operator sets $\{\hat{U}_\omega\}$ and the corresponding distributions $\{p_\omega\}$, we can provide several randomized algorithms with different efficiencies.

From Theorem 1, in each stage the increase of the trace distance is bounded from above by $||E(\hat{U}_\omega) - U||$ and $E(||\hat{U}_\omega - U||^2)$, modulo a constant. If both of these two terms are bounded from above by $O(\Delta t^{r+1})$, for some integer r, then the randomized algorithm yields a total error scaling as $O(t^{r+1}/K^r)$. Hence, the value of K required to achieve a given error bound ε scales as $O(t^{1+1/r}/\varepsilon^{1/r})$. When the number of exponentials in each stage can be considered constant, the number of exponentials equals

$$N = O(K) = O\left(t^{1+1/r}/\varepsilon^{1/r}\right).$$

We have the following corollary.

Corollary 1. *Consider a randomized algorithm for e^{-iHt}, where the evolution time t is divided into K small intervals of length Δt, and the evolution in each interval is simulated by the randomly selected $\hat{U}_\omega$ with probability p_ω. If*

$$\max\{||E(\hat{U}_\omega) - e^{-iH\Delta t}||, E(||\hat{U}_\omega - e^{-iH\Delta t}||^2)\} = O(\Delta t^{r+1}),$$

for some integer r, then the total number of exponentials of the algorithm approximating e^{-iHt} is

$$O(t^{1+1/r}/\varepsilon^{1/r}).$$

The number of exponentials in each stage represents the difficulty to implement the algorithm, i.e., the more exponentials needed in each stage, the more parameters needed to store and the more factors needed to control in the implementation of the algorithm. On the other hand, the efficiency of an algorithm is the total number of exponentials needed, i.e., the number of exponentials in each stage multiplies the number of total stages in the algorithm. Since in this paper, the number of exponentials in each stage is considered as constant, it can be determined by the total number of stages asymptotically. Particularly, an algorithm has fewer exponentials in each stage does not mean that it has higher efficiency.

3 Examples of Randomized Algorithms

In this section, we give several examples of randomized algorithms and use the lemma above to analyze their costs. The goal is to simulate the evolution of $H = \sum_{j=1}^m H_j$ for an evolution time t.

In each stage of the algorithm, the operator $U_0 = e^{-iH\Delta t}$ is simulated by the product of random operators $\hat{U}_{\omega_l}$, for $l = 1, \cdots, m$ in m consecutive substages. Let $\tilde{U}_\omega = \prod_{l=1}^m \hat{U}_{\omega_l}$, then due to Theorem 1, the error of the algorithm in each stage is decided by two elements, $||E(\tilde{U}_\omega) - U_0||$ and $E(||\tilde{U}_\omega - U_0||^2)$. Since the selection of each operator is independent and uniform,

Algorithm 1

```
1: Divide the total evolution time $t$ into $K$ equal small segments of size $\Delta t$, where $K$ will be defined later. The algorithm is comprised of $K$ stages, and in the $k$-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.
2: Let $\tilde{\rho}_0 = \rho_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.
3: In the $k$-th stage of the algorithm, there are $m$ substages. In the $l$-th substage, the initial state is $\tilde{\rho}_{k,l}$, and of course $\tilde{\rho}_{k,0} = \tilde{\rho}_{k-1}$.
    1. In each substage, the algorithm chooses uniformly and independently at random an operator from $\{e^{-iH_1\Delta t}, \ldots, e^{-iH_m\Delta t}\}$, i.e., in the $l$-th substage, the operator would be $\hat{U}_{\omega_l} = e^{-iH_{\omega_l}\Delta t}$ with probability $p_{\omega_l} = \frac{1}{m}$ for $\omega_l = 1, \cdots .m$. Taking into account all the alternatives, the final state of $l$-th substage in the $k$-th stage is
       $$\tilde{\rho}_{k,l} = \sum_{j=1}^{m} \frac{1}{m} e^{-iH_j\Delta t} \tilde{\rho}_{k,l-1} e^{iH_j\Delta t}.$$
    2. The final state of the $k$-th stage is $\tilde{\rho}_k = \tilde{\rho}_{k,m}$.
4: The final result of the algorithm is $\tilde{\rho}_K$ and is used to approximate the final quantum state.
```

$$E(\tilde{U}_\omega) = \left(\frac{1}{m}\sum_{j=1}^{m} e^{-iH_j\Delta t}\right)^m = I - i\sum_{j=1}^{m} H_j\Delta t + O(\Delta t^2).$$

Hence,

$$||E(\tilde{U}_\omega) - U_0|| = O(\Delta t^2).$$

Furthermore, for any ω, $\tilde{U}_\omega = I + O(\Delta t)$, then

$$E(||\tilde{U}_\omega - U_0||^2) = O(\Delta t^2).$$

Thus the total error is $\varepsilon = O(K\Delta t^2)$ and the total number of exponentials used in the algorithm is

$$N = mK = O(t^2/\varepsilon).$$

We remark that, modulo a constant, there is a deterministic algorithm with the same performance. The algorithm is based on a direct application of the Trotter formula $\prod_{j=1}^{m} e^{-iH_j\Delta t}$. However, **Algorithm 1** has a certain advantage over this deterministic algorithm. In each stage, the deterministic algorithm has a product of m exponentials in a precise sequence, hence it has to store the current index j of $e^{-iH_j\Delta t}$, for $j = 1, \cdots, m$. However, in **Algorithm 1**, the exponentials are random and independent of each other, hence the algorithm can be considered to be “memoryless”.

In each stage, the algorithm simulates $U_0 = e^{-iH\Delta t}$ by $\hat{U}_1$ or $\hat{U}_2$ with equal probability $1/2$. Since

Algorithm 2

```
1: Divide the total evolution time $t$ into $K$ equal small segments of size $\Delta t$. The algorithm is comprised of $K$ stages, and in the $k$-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.
2: Let $\tilde{\rho}_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.
3: In the $k$-th stage of the algorithm where the initial state is $\tilde{\rho}_{k-1}$, $k = 1, \ldots, K$. The algorithm selects a unitary operator uniformly and independently at random from $\{\hat{U}_1 = \prod_{j=1}^{m} e^{-iH_j \Delta t}, \hat{U}_2 = \prod_{j=m}^{1} e^{-iH_j \Delta t}\}$, i.e., in the $k$-th stage the operator would be $\hat{U}_\omega$ with probability $p_\omega = 1/2$, for $\omega = 1, 2$. Taking into account all the alternatives, the final state of the $k$-th stage is
       $$\tilde{\rho}_k = \frac{1}{2}\left(\hat{U}_1 \tilde{\rho}_{k-1} \hat{U}_1^\dagger + \hat{U}_2 \tilde{\rho}_{k-1} \hat{U}_2^\dagger\right).$$
4: The final result of the algorithm is $\tilde{\rho}_K$ and is used to approximate the final quantum state.
```

$$\begin{aligned}\hat{U}_1 &= \prod_{j=1}^{m}(I - iH_j\Delta t - \frac{1}{2}H_j^2\Delta t^2 + O(\Delta t^3))\\ &= I - i\sum_{j=1}^{m} H_j \Delta t - \frac{1}{2}\sum_{j=1}^{m} H_j^2 \Delta t^2 - \sum_{j_1<j_2} H_{j_1} H_{j_2} \Delta t^2 + O(\Delta t^3),\end{aligned}$$

$$\begin{aligned}\hat{U}_2 &= \prod_{j=m}^{1}(I - iH_j\Delta t - \frac{1}{2}H_j^2\Delta t^2 + O(\Delta t^3))\\ &= I - i\sum_{j=1}^{m} H_j \Delta t - \frac{1}{2}\sum_{j=1}^{m} H_j^2 \Delta t^2 - \sum_{j_1<j_2} H_{j_2} H_{j_1} \Delta t^2 + O(\Delta t^3),\end{aligned}$$

$||\hat{U}_\omega - U_0|| = O(\Delta t^2)$, for $\omega = 1, 2$, hence

$$E(||\hat{U}_\omega - U_0||^2) = O(\Delta t^4).$$

Moreover, $E(\hat{U}_\omega) = I - iH\Delta t - \frac{1}{2}H^2\Delta t^2 + O(\Delta t^3)$, hence

$$||E(\hat{U}_\omega) - U_0|| = O(\Delta t^3).$$

Due to Theorem 1, the error of this algorithm simulating $e^{-iH\Delta t}$ in each stage is $O(\Delta t^3)$. Hence, for a given accuracy ε, the total number of exponentials in **Algorithm 2** is

$$N = mK = O(t^{3/2}/\varepsilon^{1/2}).$$

We remark that, modulo a constant, there is a deterministic algorithm with the same performance. The difference is that the deterministic algorithm is more complicated, since it is based on the Strang splitting formula

$$\hat{U} = \prod_{j=1}^{m} e^{-i\frac{1}{2}H_j \Delta t} \prod_{j=m}^{1} e^{-i\frac{1}{2}H_j \Delta t}.$$

As a result, in each stage, the deterministic algorithm has $2m - 1$ exponentials, but **Algorithm 2** only has m exponentials.

Next, we focus on the case that $m = 2$. In Ref.[16], the author shows the deterministic algorithm

$$\hat{U} = e^{-i\frac{1}{2}sH_1\Delta t} e^{-isH_2\Delta t} e^{-i\frac{1}{2}(1-s)H_1\Delta t} e^{-i(1-2s)H_2\Delta t} e^{-i\frac{1}{2}(1-s)H_1\Delta t} e^{-isH_2\Delta t} e^{-i\frac{1}{2}sH_1\Delta t}, \tag{3}$$

for simulating $e^{-iH\Delta t}$, where $s = \frac{1}{2-\sqrt[3]{2}}$. The algorithm yields an error $O(\Delta t^4)$ in each stage, hence having $O(t^{4/3}/\varepsilon^{1/3})$ exponentials. However, it requires irrational evolution times, which cannot be fully accurately represented. The inaccuracy caused by these will affect the efficiency of the algorithm, and make the algorithm more difficult to implement. We present a randomized algorithm with the same performance, but using fewer and simpler exponentials in each stage.

It can be proved (see, the Appendix) that

$$E(||\hat{U}_\omega - U||^2) = O(\Delta t^4),$$

and

$$||E(\hat{U}_\omega) - U_0|| = O(\Delta t^4).$$

Hence, from Theorem 1, the error in each stage is bounded by $O(\Delta t^4)$, and the total number of exponentials used is $N = 4K = O(t^{4/3}/\varepsilon^{1/3})$.

Compared to the deterministic algorithm which uses seven exponentials in each stage, **Algorithm 3** uses only four exponentials. Moreover, the exponentials in the deterministic algorithm have irrational factors in their evolution time, which certainly bring difficulties in implementation. However, all of the exponentials used in **Algorithm 3** have very simple factors in the evolution time. From [16], it is known that it requires at least six exponentials in each stage to simulate $e^{-iH\Delta t}$ within error bound $O(\Delta t^4)$ for deterministic algorithms. For the same efficiency, we have presented a randomized algorithm, i.e., **Algorithm 3**, which uses only four exponentials in each stage.

Note that, **Algorithm 3** is not the only randomized algorithm which has $O(t^{4/3}/\varepsilon^{1/3})$ exponentials. In fact, if in each stage an algorithm selects

$$\hat{U}_\omega = e^{-ix_\omega H_1\Delta t} e^{-i(1-x_\omega)H_2\Delta t} e^{-i(1-x_\omega)H_1\Delta t} e^{-ix_\omega H_2\Delta t}$$

Algorithm 3

```
1: Divide the total evolution time $t$ into $K$ equal small segments of size $\Delta t$. The algorithm is comprised of $K$ stages, and in the $k$-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.
2: Let $\tilde{\rho}_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.
3: In the $k$-th stage of the algorithm where the initial state is $\tilde{\rho}_{k-1}$, $k = 1, \ldots, K$. The algorithm selects
       $\hat{U}_1 = e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t}$   with probability $p_1 = \frac{5}{12}$,
       $\hat{U}_2 = e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t}$   with probability $p_2 = \frac{5}{12}$,
       $\hat{U}_3 = e^{-i\frac{3}{2}H_1\Delta t}e^{i\frac{1}{2}H_2\Delta t}e^{i\frac{1}{2}H_1\Delta t}e^{-i\frac{3}{2}H_2\Delta t}$   with probability $p_3 = \frac{1}{12}$
       $\hat{U}_4 = e^{-i\frac{3}{2}H_2\Delta t}e^{i\frac{1}{2}H_1\Delta t}e^{i\frac{1}{2}H_2\Delta t}e^{-i\frac{3}{2}H_1\Delta t}$   with probability $p_4 = \frac{1}{12}$.
   i.e., the operator in the $k$-th stage is $\hat{U}_\omega$ with $p_\omega$ for $\omega = 1, 2, 3, 4$. Taking into account all the alternatives, the final state of stage $k$ is
       $\tilde{\rho}_k = \sum_{\omega=1}^{4} p_\omega \hat{U}_\omega \tilde{\rho}_{k-1} \hat{U}_\omega^\dagger.$
4: The final result of the algorithm is $\tilde{\rho}_K$ and is used to approximate the final quantum state.
```

with probability $\frac{1}{2}p_\omega$ and

$$\hat{V}_\omega = e^{-ix_\omega H_2\Delta t}e^{-i(1-x_\omega)H_1\Delta t}e^{-i(1-x_\omega)H_2\Delta t}e^{-ix_\omega H_1\Delta t}$$

with the same probability $\frac{1}{2}p_\omega$, as long as $E(x_\omega(1-x_\omega)^2) = \sum_\omega p_\omega x_\omega(1-x_\omega)^2 = \frac{1}{6}$, then the algorithm also has $O(t^{4/3}/\varepsilon^{1/3})$ exponentials.

The above examples show that there are randomized algorithms which have the same efficiencies with some deterministic algorithms, but are easier to implement. However, it is not clear whether there are randomized algorithms which have higher efficiencies than all deterministic algorithms known. It would be an interesting problem to work on.

Acknowledgements We are grateful to Anargyros Papageorgiou, Joseph F. Traub, Henryk Wozniakowski, Columbia University and Zhengfeng Ji, Perimeter Institute for Theoretical Physics, for their very helpful discussions and comments.

Appendix

Here, we provide the details of the analysis for the efficiency for **Algorithm 3**.

In each stage, the algorithm simulate $U_0 = e^{-iH\Delta t}$ with U_ω with p_ω, for $\omega = 1, 2, 3, 4$. It is easy to check, for any x,

$$\begin{aligned}
&e^{-ixH_1\Delta t}e^{-i(1-x)H_2\Delta t}e^{-i(1-x)H_1\Delta t}e^{-ixH_2\Delta t}\\
&= I - i(H_1 + H_2)\Delta t - \left[\frac{1}{2}H_1^2 + \frac{1}{2}H_2^2 + x(2-x)H_1H_2 + (1-x)^2H_2H_1\right]\Delta t^2\\
&\quad + \Bigg\{\frac{1}{6}H_1^3 + \frac{1}{6}H_2^3 + \left[x^2\left(\frac{3}{2} - x\right) + \frac{1}{2}x(1-x)^2\right]H_1^2H_2 + \frac{1}{2}(1-x)^3H_2^2H_1\\
&\quad + \left[\frac{1}{2}x(1-x)^2 + x^2(\frac{3}{2} - x)\right]H_1H_2^2 + \frac{1}{2}(1-x)^3H_2H_1^2\\
&\quad + x(1-x)^2H_1H_2H_1 + x(1-x)^2H_2H_1H_2\Bigg\}\Delta t^3 + O(\Delta t^4).
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
&e^{-ixH_2\Delta t}e^{-i(1-x)H_1\Delta t}e^{-i(1-x)H_2\Delta t}e^{-ixH_1\Delta t}\\
&= I - i(H_1 + H_2)\Delta t - \left[\frac{1}{2}H_1^2 + \frac{1}{2}H_2^2 + x(2-x)H_2H_1 + (1-x)^2H_1H_2\right]\Delta t^2\\
&\quad + \Bigg\{\frac{1}{6}H_1^3 + \frac{1}{6}H_2^3 + \left[x^2\left(\frac{3}{2} - x\right) + \frac{1}{2}x(1-x)^2\right]H_2^2H_1 + \frac{1}{2}(1-x)^3H_1^2H_2\\
&\quad + \left[\frac{1}{2}x(1-x)^2 + x^2\left(\frac{3}{2} - x\right)\right]H_2H_1^2 + \frac{1}{2}(1-x)^3H_1H_2^2\\
&\quad + x(1-x)^2H_2H_1H_2 + x(1-x)^2H_1H_2H_1\Bigg\}\Delta t^3 + O(\Delta t^4).
\end{aligned} \tag{5}$$

Therefore, $||\hat{U}_\omega - U_0|| = O(\Delta t^2)$, for $\omega = 1, \cdots, 4$, and

$$E(||\hat{U}_\omega - U||^2) = O(\Delta t^4).$$

Furthermore, from Eqs. 4 and 5,

$$\begin{aligned}
E(\hat{U}_\omega) =& I - i(H_1 + H_2)\Delta t - \frac{1}{2}(H_1 + H_2)^2\Delta t^2\\
&+ i\Bigg[\frac{1}{2}(\frac{1}{2} - E(x(1-x)^2))(H_1H_2^2 + H_2H_1^2 + H_1^2H_2 + H_2^2H_1)\\
&+ E(x(1-x)^2)(H_1H_2H_1 + H_2H_1H_2)\Bigg]\Delta t^3 + O(\Delta t^4),
\end{aligned}$$

where

$$E(x(1-x)^2) = 2 \times \frac{5}{12} \times \frac{1}{2}\left(1 - \frac{1}{2}\right)^2 + 2 \times \frac{1}{12} \times \frac{3}{2}\left(1 - \frac{3}{2}\right)^2 = \frac{1}{6}.$$

Hence,

$$||E(\hat{U}_\omega) - U_0|| = O(\Delta t^4).$$

From Theorem 1, the error in each stage is bounded by $O(\Delta t^4)$, and the total number of exponentials used is $N = 4K = O(t^{4/3}/\varepsilon^{1/3})$.

References

1. R. P. Feynman, Simulating Physics with computers, Int. J. Theoret. Phys. 21, 467–488 (1982)
2. S. Lloyd, Universal quantum simulators, Science 273, 1073–1078 (1996)
3. C. Zalka, Simulating Quantum Systems on a Quantum Computer, Proc. R. Soc. Lond. A, 454, 313–323 (1998)
4. D. W. Berry, G. Ahokas, R. Cleve, B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Communications in Mathematical Physics 270, 359 (2007)
5. I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, A. Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, Proc. Natl. Acad. Sci. 105, 18681(2008)
6. D. Aharonov, A. Ta-Shma, Adiabatic quantum state generation and statistical zero knowledge, Proc. 35th Annual ACM Symp. on Theoty of Computing, 20–29 (2003)
7. E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, quant-ph/0001106 (2000)
8. E. Farhi, S. Gutmann, Analog analogue of a digital quantum computation. Phys. Rev. A 57(4), 24032406 (1998)
9. A. M. Childs, E. Farhi, S. Gutmann, An example of the difference between quantum and classical random walks, J. Quant. Inf. Proc. 1, 35–43 (2002)
10. E. Farhi, J. Goldstone, S. Gutmann, A Quantum Algorithm for the Hamiltonian NAND Tree, quant-ph/0702144 (2007)
11. A. M. Childs, Universal computation by quantum walk, Phys. Rev. Lett, 102, 180501 (2009)
12. A. M. Childs, On the relationship between continuous- and discrete-time quantum walk, quant-ph/0810.0312 (2008)
13. D. W. Berry, A. M. Childs, The quantum query complexity of implementing black-box unitary transformations, quant-ph/0910. 4157 (2009)
14. M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press (2000)
15. G. Strang, On the construction and comparison of difference schemes, SIAM J. Num. Analysis, 506–517 (1968)
16. M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Phys. Lett. A 146, 319–323 (1990)
17. M. Suzuki, General theory of fractal path integrals with application to many-body theories and statistical physics, J. Math. Phys, 32, 400–407 (1991)

Conference Participants

Hamza Alkhatib Leibniz University Hannover, Ninenburger Str. 1, Hannover, 30167, Germany, alkhatib@gih.uni-hannover.de

Anton Antonov Saint-Petersburg State University, Basseynaya str., 67-37, Saint-Petersburg, 196211, Russia, tonytonov@mail.ru

Søren Asmussen Aarhus University, Department of Mathematical Sciences, Aarhus, DK-8000, Denmark, asmus@imf.au.dk

Rainer Avikainen Åbo Akademi University, Matematik, Fänriksgatan 3B, Åbo, 20500, Finland, rainer.avikainen@abo.fi

Jan Baldeaux University of Technology, Sydney, School of Finance and Economics, Sydney, 2007, Australia, JanBaldeaux@gmail.com

Lev Barash Landau Institute for Theoretical Physics, 142432 Chernogolovka, Akademika Semenova av., 1-A, Chernogolovka, Moscow Region, 142432, Russia, barash@itp.ac.ru

Serge Barbeau Visitor, 3, rue du Helder, Paris, 75009, France, serge_barbeau@hotmail.com

Mylene Bedard Universite de Montreal, DMS, UdeM, CP 6128, succ Centre-ville, Montreal, H3C3J7, Canada, bedard@dms.umontreal.ca

Dmitriy Bilyk University of South Carolina, 1523 Greene St, Dept of Math, USC, Columbia, SC, 29208, United States of America, bilyk@math.sc.edu

Bruno Bouchard CEREMADE Univ. Paris Dauphine - CREST Ensae, Place du Marechal, Paris Cedex 16, 75775, France, bouchard@ceremade.dauphine.fr

Sylvestre Burgos University of Oxford, Lady Margaret Hall, Norham Gardens, Oxford, OX26QA, United Kingdom, sylvestre.burgos@maths.ox.ac.uk

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4, © Springer-Verlag Berlin Heidelberg 2012

Aleksandr Burmistrov Institute of Computational Mathematics and Mathematical Geophysics SB RAS, prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia, burm@osmf.sscc.ru

William Chen Macquarie University Sydney, Department of Mathematics, Macquarie University NSW Sydney, 2109, Australia, william.chen@mq.edu.au

Nan Chen The Chinese University of Hong Kong, 709A, William Mong Engineering Building, CUHK, Shatin, NA, Hong Kong-S.A.R of China, nchen@se.cuhk.edu.hk

Su Chen Stanford University, 39 Angell Court, Apt 106, Stanford, CA, 94305, United States of America, suchenpku@gmail.com

Janusz Chwastowski Institute of Teleinformatics, Cracow University of Technology & INP Cracow, Warszawska 24, Kraków, 31-155, Poland, jchwastowski@pk.edu.pl

Ronald Cools Katholieke Universiteit Leuven, Celestijnenlaan 200A - bus 2402, Heverlee, B-3001, Belgium, ronald.cools@cs.kuleuven.be

Fred Daum Raytheon, 318 Acton Street, Carlisle MA, 01741, United States of America, frederick_e_daum@raytheon.com

Thomas Daun University of Kaiserslautern, Kurt-Schumacher-Straße 32, Kaiserslautern, 67663, Germany, daun@cs.uni-kl.de

Steffen Dereich Philipps-Universität Marburg, Frankfurter Str. 17, Marburg, 35037, Germany, dereich@mathematik.uni-marburg.de

Josef Dick The University of New South Wales, School of Mathematics and Statistics, UNSW, Sydney, 2052, Australia, josef.dick@unsw.edu.au

Jacques du Toit Smith Institute, Surrey Technology Centre, Surrey Research Park, Guidford, Surrey, GU2 7Y, United Kingdom, office@smithinst.co.uk

R. Gabriel Esteves University of Waterloo, School of Computer Science, Waterloo, N2L 3G, Canada, rgesteve@uwaterloo.ca

Henri Faure Institut de Mathématiques de Luminy, CNRS UMR 6206, IML, 163 Av.de Luminy, case 907, Marseille Cedex 09, 13288, France, faure@iml.univ-mrs.fr

James Flegal University of California, Riverside, 2626 STAT/COMP, Riverside, CA, 92521, United States of America, jflegal@ucr.edu

Gersende Fort CNRS / LTCI, 46 rue Barrault 75634 Paris cedex 13, paris, 75634, France, gersende.fort@telecom-paristech.fr

Stefan Geiss University of Innsbruck, Technikerstrasse 13, Innsbruck, 6020, Austria, stefan.geiss@uibk.ac.at

Christel Geiss University of Innsbruck, Technikerstraße 13/7, Innsbruck, A-6020, Austria, chgeiss@maths.jyu.fi

Alan Genz Washington State University, PO Box 643113, Pullman, WA, 99164, United States of America, alangenz@wsu.edu

Mike Giles University of Oxford, Oxford-Man Institute, Eagle House, Walton Well Roa, Oxford, OX2 6E, United Kingdom, mike.giles@maths.ox.ac.uk

Michael Gnewuch Columbia University, Department of Computer Science, 1214 Amsterdam Ave, New York, 10027, United States of America, mig@informatik.uni-kiel.de

Maciej Goćwin AGH Kraków, ul. Zwierzyniecka 17/2, Kraków, 31-103, Poland, gocwin@agh.edu.pl

Emmanuel Gobet Grenoble Institute of Technology, LJK - BP 53, Grenoble cedex 09, 38041, France, emmanuel.gobet@imag.fr

Anatoly Gormin Saint-Petersburg State University, Peterhof, Universitetsky prospekt, 28, Saint-Pete, Saint-Petersburg, 198504, Russia, anatoly_ag@yahoo.com

C.-H. Sean Han National Tsing-Hua University, Department of Quantitative Finance NTHU, 101, Sec, Hsinchu, 300, Taiwan, chhan@mx.nthu.edu.tw

Hiroshi Haramoto Kure National College of Technology, 2-2-11 Agaminami, Kure, Hiroshima, 737850, Japan, haramoto@hiroshima-u.ac.jp

Shin Harase The University of Tokyo, 3-8-1, Komaba, Meguro-ku, Tokyo, 153-89, Japan, harase@ms.u-tokyo.ac.jp

Josef Höök AlRoyal Institute of Technology (KTH), Teknikringen 31, Stockholm, 10044, Sweden, joh@kth.se

Carole Hayakawa University of California, Irvine, 916 Engineering Tower, Irvine, 92697, United States of America, hayakawa@uci.edu

Felix Heidenreich TU Kaiserslautern, Erwin-Schroedinger-Strasse, Kaiserslautern, 67663, Germany, heidenreich@mathematik.uni-kl.de

Stefan Heinrich University of Kaiserslautern, Fachbereich Informatik, Kaiserslautern, 67653, Germany, heinrich@informatik.uni-kl.de

Peter Hellekalek University of Salzburg, Hellbrunner Strasse 34, Salzburg, 5020, Austria, peter.hellekalek@sbg.ac.at

Daniel Henkel TU Darmstadt, Schlossgartenstrasse 7, Darmstadt, 64289, Germany, henkel@mathematik.tu-darmstadt.de

Fred J. Hickernell Illinois Institute of Technology, Applied Math, Room E1-208, 10 W. 32nd Street, Chicago, IL, 60616, United States of America, hickernell@iit.edu

Aicke Hinrichs FSU Jena, Institute of Mathematics, FSU Jena, Ernst-Abbe-Platz 2, Jena, 07743, Germany, a.hinrichs@uni-jena.de

Mark Huber Claremont McKenna College, 800 Columbia Avenue, Claremont, CA, 91711, United States of America, mhuber@cmc.edu

Andrzej Jarynowski UNESCO Chair(KUSI). Wroclaw University, Slowackiego 24/12, Kwidzyn, 82-500, Poland, gulakov@dsv.su.se

Arnulf Jentzen Bielefeld University, Universitätsstraße 25, Bielefeld, 33501, Germany, jentzen@math.uni-bielefeld.de

Stephen Joe University of Waikato, Dept of Mathematics, University of Waikato, Hamilton, 3240, New Zealand, stephenj@math.waikato.ac.nz

Galin Jones University of Minnesota, 224 Church Street S.E., Minneapolis, MN, 55455, United States of America, galin@stat.umn.edu

Bolesław Kacewicz AGH-UST Kraków, al. Mickiewicza 30, paw. A3/A4, p.301, Kraków, 30-059, Poland, kacewicz@agh.edu.pl

Roman Kapuscinski University of Michigan, 701 Tappan St, Ann Arbor, 48109, United States of America, kapuscin@umich.edu

Aneta Karaivanova IPP-BAS, Acad. G. Bonchev St., bl. 25A, Sofia, 1113, Bulgaria, anet@parallel.bas.bg

Reiichiro Kawai University of Leicester, Department of Mathematics, University Road, Leicester, LE17RH, United Kingdom, reiichiro.kawai@gmail.com

Thomas Kühn MUniversität Leipzig, Johannisgasse 26, Leipzig, 04103, Germany, kuehn@math.uni-leipzig.de

Lutz Kämmerer Chemnitz University of Technology, Fakultät für Mathematik, Chemnitz, 09107, Germany, kaemmerer@mathematik.tu-chemnitz.de

Alexander Keller NVIDIA Research, Berlin, Germany, keller.alexander@gmail.com

Mariya Korotchenko Institute of Computational Mathematics and Mathematical Geophysics SB RAS, prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia, kmaria@osmf.sscc.ru

Marcin Krzysztofik NAG, Wilkinson House, Jordan Hill Road, Oxford, OX2 8D, United Kingdom, marcin.krzysztofik@nag.co.uk

Piotr Krzyżanowski University of Warsaw, Banacha 2, Warszawa, 00-097, Poland, przykry@mimuw.edu.pl

Sergei Kucherenko Imperial College London, via Foresteria, 248, app 20, Ispra, 21027, Italy, s.kucherenko@ic.ac.uk

Frances Kuo University of New South Wales, School of Mathematics and Statistics, University o, Sydney NSW, 2052, Australia, f.kuo@unsw.edu.au

Marek Kwas Warsaw School of Economics, Al. Niepodleglosci 164, Warsaw, 05-825, Poland, kwasem@gmail.com

Pierre L'Ecuyer University of Montreal, DIRO, 2920 Chemin de la Tour, local 2194, Montreal (Que), H3T 1J, Canada, lecuyer@iro.umontreal.ca

Krzysztof Latuszynski Warwick University, Dept of Statistics, Coventry, CV4 7A, United Kingdom, latuch@gmail.com

Christian Lecot LAMA UMR 5127 CNRS & Universite de Savoie, Campus scientifique, Le Bourget-du-Lac, 73376, France, Christian.Lecot@univ-savoie.fr

Paul Leopardi Australian National University, MSI Building 27, ANU ACT, 0200, Australia, paul.leopardi@anu.edu.au

Yanchu Liu The Chinese University of Hong Kong, Rm 609, William M. W. Mong Engineering Building, T, Shatin, Hong Kong, 000000, Hong Kong-S.A.R of China, ycliu@se.cuhk.edu.hk

Sylvain Maire Université de Toulon (France), 7 avenue Guy Teisseire, Cuers, 83390, France, maire@univ-tln.fr

Earl Maize Jet Propulsion Laboratory, 4800 Oak Grove Drive MS 198-106, Pasadena, CA, 91109, United States of America, earl.h.maize@jpl.nasa.gov

Roman Makarov Wilfrid Laurier University, 75 University Avenue West, Waterloo, Ontario, N2L3C5, Canada, rmakarov@wlu.ca

Peter Mathe Weierstrass Institute, Mohrenstrasse 39, Berlin, 10117, Germany, mathe@wias-berlin.de

Makoto Matsumoto University of Tokyo, Dept. Math, 3-8-1 Komaba Meguro-ku, Tokyo, 153891, Japan, matumoto@ms.u-tokyo.ac.jp

Ilya Medvedev ICMMG of SB RAS, pr. Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia, min@osmf.sscc.ru

Błażej Miasojedow University of Warsaw, Department of Mathematics, Banacha 2, Warszawa, 00-097, Poland, bmia@mimuw.edu.pl

Ladislav Mišík University of Ostrava, 30. dubna 22, Ostrava, 70102, Czech Republic, ladislav.misik@osu.cz

Hozumi Morohosi National Graduate Institute for Policy Studies, 7-22-1 Roppongi, Minato-ku, Tokyo, 106867, Japan, morohosi@grips.ac.jp

Thomas Mueller-Gronbach University of Passau, FIM, Innstrasse 33, Passau, 94032, Germany, thomas.mueller-gronbach@uni-passau.de

Andreas Neuenkirch TU Dortmund, Vogelpothsweg 87, Dortmund, 44227, Germany, andreas.neuenkirch@math.tu-dortmund.de

James Nichols University of New South Wales, 9/253 Palmer St, Darlinghurst, 2010, Australia, james.ashton.nichols@gmail.com

Harald Niederreiter Austrian Academy of Sciences, Josefiaustr. 16, Salzburg, 5020, Austria, ghnied@gmail.com

Wojciech Niemiro Nicolaus Copernicus University, Toruń and University of Warsaw, Banacha 2, Warszawa, 02-097, Poland, wniemiro@gmail.com

Erich Novak Jena University, Ernst Abbe Platz 2, Jena, 07743, Germany, erich.novak@uni-jena.de

Dirk Nuyens Katholieke Universiteit Leuven, Celestijnenlaan 200A - bus 2402, Heverlee, B-3001, Belgium, dirk.nuyens@cs.kuleuven.be

Maciej Obremski University of Warsaw, ul. Banacha 2, Warsaw, 02-097, Poland, obremski@mimuw.edu.pl

Giray Okten Florida State University, FSU Mathematics, 208 Love Building, 1017 Academic, Tallahassee, FL, 32306-, United States of America, okten@math.fsu.edu

Art Owen Stanford University, Department of Statistics, Sequoia Hall, Stanford, 94305, United States of America, owen@stanford.edu

Andrzej Palczewski University of Warsaw, Banacha 2, Warsaw, 02-097, Poland, apalczew@mimuw.edu.pl

Anargyros Papageorgiou Columbia University, Department of Computer Science, 1214 Amsterdam Avenue, MC0401, New York, NY, 10027, United States of America, ap@cs.columbia.edu

Peter Parczewski Saarland university, Department of Mathematics, Saarbrücken, 66041, Germany, parcz@math.uni-sb.de

Florian Pausinger Universität Salzburg, Hellbrunnerstraße 34, Salzburg, 5020, Austria, florianfranz.pausinger@sbg.ac.at

François Perron University of Montreal, Dept of Math, CP 6128 centre-ville, Montreal, H3T1J4, Canada, perronf@dms.umontreal.ca

Friedrich Pillichshammer University of Linz, Altenbergerstrasse 69, Linz, 4040, Austria, friedrich.pillichshammer@jku.at

Leszek Plaskota University of Warsaw, ul. Banacha 2, Warsaw, 02-097, Poland, leszekp@mimuw.edu.pl

Yvo Pokern University College London, Department of Statistical Science, UCL, London, WC1E 6, United Kingdom, yvo@stats.ucl.ac.uk

Nick Polydorides Cyprus Institute, P.O. Box 27456, Nicosia, 1645, Cyprus, nickpld@mit.edu

Koen Poppe Katholieke Universiteit Leuven, Celestijnenlaan 200A - bus 2402, Heverlee, B-3001, Belgium, koen.poppe@cs.kuleuven.be

Paweł Przybyowicz AGH Krak”ow, Al. Mickiewicza 30, Cracow, 30-059, Poland, przybyl83@gmail.com

Klaus Ritter TU Kaiserslautern, Department of Mathematics, Kaiserslautern, 67653, Germany, ritter@mathematik.uni-kl.de

Gareth Roberts University of Warwick, Department of Statistics, Coventry, CV4 7AL, United Kingdom, Gareth.O.Roberts@warwick.ac.uk

Daniel Rudolf University of Jena, Ernst-Abbe-Platz 2, Jena, 07743, Germany, daniel.rudolf@uni-jena.de

Anna Rukavishnikova S-Petersburg State University, Gavanskaya 10, 38, S-Petersburg, 199106, Russia, anyaruk@mail.ru

Eero Saksman University of Helsinki, P.O. Box 68, University of Helsinki, 00014, Finland, eero.saksman@helsinki.fi

Wolfgang Ch. Schmid University of Salzburg, Hellbrunnerstr. 34, Salzburg, A-5020, Austria, wolfgang.schmid@sbg.ac.at

Heikki Seppälä Univeristy of Jyväskylä, , P.O. Box 35, University of Jyväskylä, FIN-40, Finland, heikki.seppala@jyu.fi

John Sepikas Pasadena City College, 6149 Fulton Ave, Van Nuys, 91401, United States of America, jsepikas@hotmail.com

Fatin Sezgin Bilkent University, Dept of BIM SATM, Ankara, 06533, Turkey, fatin@bilkent.edu.tr

Vasile Sinescu Katholieke Universiteit Leuven, Dept. of Computer Science, Celestijnenlaan 200A -, Heverlee, 3001, Belgium, vasile.sinescu@cs.kuleuven.be

Mehdi Slassi TU Darmstadt, Schlossgartenstraße 7, Darmstadt, 64289, Germany, slassi@mathematik.tu-darmstadt.de

Ian Sloan University of New South Wales, School of Mathematics and Statsistics, Sydney, NSW, 2052, Australia, i.sloan@unsw.edu.au

Sergey Smirnov Saint-Petersburg State University, Peterhof,Botanicheskaya 70/2,flat 204, Saint-Petersburg, 198504, Russia, q4ality@gmail.com

Jerome Spanier University of California, Irvine, 1002 Health Sciences Road, E., Irvine, California, 92612, United States of America, jspanier@uci.edu

Łukasz Stettner Polish Academy Sciences, Sniadeckich 8, Warsaw, 00-956, Poland, stettner@impan.gov.pl

Oto Strauch Slovak Academy of Sciences, Štefánikova 49, Bratislava, SK-814, Slovakia, strauch@mat.savba.sk

Ralph Tandetzky University of Jena, Mathematisches Institut Ernst-Abbe-Platz 2, Jena, 07743, Germany, ralph.tandetzky@googlemail.com

Peter Tankov Laboratoire de Probabilités et Modeles Aléatoires, Université Paris Diderot – Paris 7, Paris, France, tankov@math.univ-paris-diderot.fr

Shu Tezuka Kyushu University, Motooka 744, Fukuoka, 819-03, Japan, tezuka@math.kyushu-u.ac.jp

Tomáš Tichý DepTechnical University Ostrava, Sokolska 33, Ostrava, 70121, Czech Republic, tomas.tichy@vsb.cz

Anni Toivola University of Jyväskylä, Survontie 46 B 72, Jyväskylä, 40520, Finland, anni.toivola@jyu.fi

Mario Ullrich Friedrich Schiller Universtity Jena, FSU Jena - Math. Institut, Ernst-Abbe-Platz 2, Jena, 07743, Germany, mario.ullrich@uni-jena.de

Matti Vihola University of Jyväskylä, Department of Mathematics and Statistics, P.O.B.35, University of Jyväskylä, 40014, Finland, matti.vihola@jyu.fi

Jochen Voss University of Leeds, School of Mathematics,, Leeds, LS2 9J, United Kingdom, J.Voss@leeds.ac.uk

Magnus Wahlström Max Planck Institute for Informatics, Stuhlsatzenhausweg 85, Saarbrücken, 66111, Germany, wahl@mpi-inf.mpg.de

Ruihong Wang IAPCM, NO.2 Fenghaodong Road,Haidian District, Beijing, 100094, China, wang_ruihong@iapcm.ac.cn

Xiaoqun Wang Tsinghua University, Department of Mathmatical Sciences, Beijing, 100084, China, xwang@math.tsinghua.edu.cn

Grzegorz Wasilkowski University of Kentucky, Computer Science Dept., 773 Anderson Hall, Lexington, 40515, United States of America, greg@cs.uky.edu

Markus Weimar University of Jena, FSU Jena, Mathematical Institute, Jena, 07740, Germany, markus.weimar@uni-jena.de

Art Werschulz Fordham University, Columbia University, Columbia University, Dept. Computer Science, New York, 10027, United States of America, agw@cs.columbia.edu

David White Warwick University, Maths Dept, Coventry, CV47AL, United Kingdom, david.white@warwick.ac.uk

Marek Wielgosz Polish Financial Supervision Authority, Plac Powstancow Warszawy 1, Warszawa, 00-950, Poland, marek.wielgosz@knf.gov.pl

Dawn Woodard Cornell University, 206 Rhodes Hall, ORIE, Ithaca, NY, 14853, United States of America, woodard@cornell.edu

Henryk Woźniakowski Columbia University & University of Warsaw, Dept. Computer Science, Columbia University, New York, 10027, USA, Poland, henryk@cs.columbia.edu

Yuan Xia University of Oxford, St Hugh's College, Oxford, OX2 6L, United Kingdom, yuan.xia@oxford-man.ox.ac.uk

Andrey Yakovenko Saint-Petersburg State University, Chernyshevsky sq. 10-112, Saint-Petersburg, 196070, Russia, yakovenko.a.b@gmail.com

Larisa Yaroslavtseva University of Passau, FIM, Innstrasse 33, Passau, 94032, Germany, larisa.yaroslavtseva@uni-passau.de

Marta Zalewska Medical University of Warsaw, Zaklad Profilaktyki Zagrozen Srodowiskowych i Alergologii, Warszawa, 02-097, Poland, zalewska.marta@gmail.com

Pawel Zareba Kempen & Co, Amsterdam, Beethovenstraat 300, Amsterdam, 1077 WZ, Netherlands, pawel.zareba@kempen.nl

Chi Zhang Columbia University, Dept. of Computer Science, 110 Morningside Drive 32#C, New York, 10027, United States of America, czhang@cs.columbia.edu

Index

L. Plaskota and H. Woźniakowski (eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2010*, Springer Proceedings in Mathematics and Statistics 23,
DOI 10.1007/978-3-642-27440-4, © Springer-Verlag Berlin Heidelberg 2012