Toeplitz Monte Carlo

Josef Dick · Takashi Goda · Hiroya Murata

Received: date / Accepted: date

Abstract Motivated mainly by applications to partial differential equations with random coefficients, we introduce a new class of Monte Carlo estimators, called Toeplitz Monte Carlo (TMC) estimator for approximating the integral of a multivariate function with respect to the direct product of an identical univariate probability measure. The TMC estimator generates a sequence $x_1, x_2, \ldots$ of i.i.d. samples for one random variable, and then uses $(x_{n+s-1}, x_{n+s-2}, \ldots, x_n)$ with $n = 1, 2, \ldots$ as quadrature points, where $s$ denotes the dimension. Although consecutive points have some dependency, the concatenation of all quadrature nodes is represented by a Toeplitz matrix, which allows for a fast matrix-vector multiplication. In this paper we study the variance of the TMC estimator and its dependence on the dimension $s$. Numerical experiments confirm the considerable efficiency improvement over the standard Monte Carlo estimator for applications to partial differential equations with random coefficients, particularly when the dimension $s$ is large.

Keywords Monte Carlo · Toeplitz matrix · fast matrix-vector multiplication · high-dimensional integration · PDEs with random coefficients

Mathematics Subject Classification (2010) MSC 65C05

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

T. Goda · H. Murata
School of Engineering, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: goda@frcer.t.u-tokyo.ac.jp
E-mail: mh22151210@gmail.com

1 Introduction

The motivation of this research mainly comes from applications to uncertainty quantification for ordinary or partial differential equations with random coefficients. The problem we are interested in is to estimate an expectation (integral)

$$I_{\rho_s}(f) := \int_{\Omega^s} f(x)\rho_s(x) \, dx \quad \text{with} \quad \rho_s(x) = \prod_{j=1}^{s} \rho(x_j),$$

for large $s$ with $\rho$ being the univariate probability density function defined over $\Omega \subseteq \mathbb{R}$. In some applications, the integrand is of the form

$$f(x) = g(xA),$$

for a matrix $A \in \mathbb{R}^{s \times t}$ and a function $g: \mathbb{R}^t \rightarrow \mathbb{R}$, see Dick et al. (2015). Here we note that $x$ is defined as a row vector. Typically, $\rho$ is given by the uniform distribution on the unit interval $\Omega = [0, 1]$, or by the standard normal distribution on the real line $\Omega = \mathbb{R}$.

The standard Monte Carlo method approximates $I_{\rho_s}(f)$ as follows: we first generate a sequence of i.i.d. samples of the random variables $x \sim \rho_s$:

$$x_1 = (x_{1,1}, \ldots, x_{s,1}), x_2 = (x_{1,2}, \ldots, x_{s,2}), \ldots,$$

and then approximate $I_{\rho_s}(f)$ by

$$I_{\rho_s}^{\text{MC}}(f; N) = \frac{1}{N} \sum_{n=1}^{N} f(x_n) = \frac{1}{N} \sum_{n=1}^{N} g(x_nA). \quad (1)$$

It is well known that

$$\mathbb{E}[I_{\rho_s}^{\text{MC}}(f; N)] = I_{\rho_s}(f)$$

and

$$\mathbb{V}[I_{\rho_s}^{\text{MC}}(f; N)] = \frac{I_{\rho_s}(f^2) - (I_{\rho_s}(f))^2}{N},$$
which ensures the canonical “one over square root of \( N \)” convergence.

Now let us consider a situation where computing \( x_n A \) for \( n = 1, \ldots, N \) takes a significant amount of time in the computation of \( I^{\text{MC}}_{\rho_s}(f; N) \). In general, if the matrix \( A \) does not have any special structure such as circulant, Hankel, Toeplitz, or Vandermonde, then fast matrix-vector multiplication is not available and the computation of \( I^{\text{MC}}_{\rho_s}(f; N) \) requires \( O(Nst) \) arithmetic operations. Some examples where a fast matrix-vector multiplication has been established are the following: In Feischl et al. (2018) the authors use H-matrices to obtain an approximation of a covariance matrix which also permits a fast matrix vector multiplication; In Giles et al. (2008) the authors show how a (partially) fast matrix vector product can be implemented for multi-asset pricing in finance; Brownian bridge and principle component analysis factorizations of the covariance matrix in finance also permit a fast matrix vector multiplication (Giles et al. 2008). Here we consider the case where either a fast matrix-vector product is not available, or one wants to avoid H-matrices and particular covariance factorizations, since we do not impose any restrictions on \( A \).

In order to reduce this computational cost, we propose an alternative, novel Monte Carlo estimator in this paper. Instead of generating a sequence of i.i.d. samples of the vector \( x \), we generate a sequence of i.i.d. samples of a single random variable, denoted by \( x_1, x_2, \ldots, \), and then approximate \( I^{\text{TMC}}_{\rho_s}(f) \) by

\[
I^{\text{TMC}}_{\rho_s}(f; N) = \frac{1}{N} \sum_{n=1}^{N} f(\tilde{x}_n)
\]

with

\[
\tilde{x}_n = (x_{n+s-1}, \ldots, x_n).
\]

The computation of \( I^{\text{TMC}}_{\rho_s}(f; N) \) can be done as follows:

**Algorithm 1** For \( N \in \mathbb{Z}_{\geq 0} \), let \( x_1, x_2, \ldots, x_{N+s-1} \) be \( N + s - 1 \) i.i.d. samples of a random variable following \( \rho \).

1. Define \( X \in \mathbb{R}^{N \times s} \) by

\[
X = \begin{pmatrix}
X_s & x_{s-1} & x_{s-2} & \cdots & x_1 \\
x_{s+1} & x_s & x_{s-1} & \cdots & x_2 \\
x_{s+2} & x_{s+1} & x_s & \cdots & x_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_{N+s-2} & x_{N+s-3} & x_{N+s-4} & \cdots & x_{N-1} \\
x_{N+s-1} & x_{N+s-2} & x_{N+s-3} & \cdots & x_N
\end{pmatrix}
\]

Note that \( X \) is a Toeplitz matrix.

2. Compute

\[
XA = Y = \begin{pmatrix}
y_1 \\
y_2 \\
y_N
\end{pmatrix} \in \mathbb{R}^{N \times t}.
\]

3. Then \( I^{\text{TMC}}_{\rho_s}(f; N) \) is given by

\[
I^{\text{TMC}}_{\rho_s}(f; N) = \frac{1}{N} \sum_{n=1}^{N} g(y_n).
\]

The idea behind introducing this algorithm comes from a recent paper by Dick et al. (2015) who consider replacing the point set used in the standard Monte Carlo estimator (MC) with a special class of quasi-Monte Carlo point sets which permit a fast matrix-vector multiplication \( x_n A \) for \( n = 1, \ldots, N \). This paper considers a sampling scheme different from Dick et al. (2015) while still allowing for a fast matrix-vector multiplication.

When \( s \) is quite large, say thousands or million, \( N \) has to be set significantly smaller than \( 2^s \). Throughout this paper we consider the case where \( N \approx s^\kappa \) for some \( \kappa > 0 \). Since the matrix-vector multiplication between a Toeplitz matrix \( X \) and each column vector of \( A \) can be done with \( O(N \log s) \) arithmetic operations by using the fast Fourier transform (Frigo and Johnson 2005), the matrix-matrix multiplication \( XA \) appearing in the second item of Algorithm 1 can be done with \( O(tN \log s) \) arithmetic operations. This way the necessary computational cost can be reduced from \( O(Nst) \) to \( O(tN \log s) \), which is the major advantage of using \( I^{\text{TMC}}_{\rho_s}(f; N) \).

In this paper we call \( I^{\text{TMC}}_{\rho_s}(f; N) \) a Toeplitz Monte Carlo (TMC) estimator of \( I_{\rho_s}(f) \) as we rely on the Toeplitz structure of \( X \) to achieve a faster computation. In the remainder of this paper, we study the variance of the TMC estimator and its dependence on the dimension \( s \), and also see practical efficiency of the TMC estimator by carrying out numerical experiments for applications from ordinary/partial differential equations with random coefficients.

\footnote{If the sample nodes are given by \( \tilde{x}_n = (x_1, \ldots, x_{n+s-1}) \) instead of \( \tilde{x}_n = (x_{n+s-1}, \ldots, x_1) \), the matrix \( X \) becomes a Hankel matrix, which also allows for a fast matrix-vector multiplication. Therefore we can call our proposal a Hankel Monte Carlo (HMC) estimator instead. However, in the context of Monte Carlo methods, HMC often refers to the Hamiltonian Monte Carlo algorithm, and we would like to avoid duplication of the abbreviations by coining the name Toeplitz Monte Carlo.}
2 Theoretical results

2.1 Variance analysis

In order to study the variance of $I_{\rho}^{\text{TMC}}(f; N)$, we introduce the concept of the analysis-of-variance (ANOVA) decomposition of multivariate functions (Hoeffding 1948, Kuo et al. 2010, Sobol’ 1993). In what follows, for simplicity of notation, we write $[1 : s] = \{1, \ldots, s\}$. For a subset of $u \subseteq [1 : s]$, we write $-u := [1 : s] \setminus u$ and denote the cardinality of $u$ by $|u|$. Let $f$ be a square-integrable function, i.e., $I_{\rho}(f^2) < \infty$. Then $f$ can be decomposed into

$$f(x) = \sum_{u \subseteq [1 : s]} f_u(x_u),$$

where we write $x_u = (x_j)_{j \in u}$ and each summand is defined recursively by $f_{\emptyset} = I_{\rho}(f)$ and

$$f_u(x_u) = \int_{D_{-u}} f(x)\rho_{s|-u}(x_{-u})\,dx_u - \sum_{v \subseteq u} f_v(x_v)$$

for $\emptyset \neq u \subseteq [1 : s]$. Regarding this decomposition of multivariate functions, the following properties hold. We refer to Lemmas A.1 & A.3 of Owen (2019) for the proof of the case where $\rho$ is the uniform distribution over the unit interval $\Omega = [0, 1]$.

**Lemma 1** With the notation above, we have:

1. For any non-empty $u \subseteq [1 : s]$ and $j \in u$,

$$\int_{\Omega} f_u(x_u)\rho(x_j)\,dx_j = 0.$$

2. For any $u, v \subseteq [1 : s]$,

$$I_{\rho}(f_u f_v) = \int_{\Omega} f_u(x_u) f_v(x_v)\rho_{|u}(x_u)\,dx_u - \sum_{x_u \in u} I_{\rho}(f_u^2)$$

$$= \begin{cases} I_{\rho}(f_u^2) & \text{if } u = v, \\ 0 & \text{otherwise}. \end{cases}$$

It follows from the second assertion of Lemma [1] that

$$I_{\rho}(f^2) = I_{\rho} \left( \sum_{u \subseteq [1 : s]} f_u f_u \right) = \sum_{u \subseteq [1 : s]} I_{\rho}(f_u f_u) = \sum_{u \subseteq [1 : s]} I_{\rho}(f_u^2).$$

This equality means that the variance of $f$ can be expressed as a sum of the variances of the lower-dimensional functions:

$$I_{\rho}(f^2) - (I_{\rho}(f))^2 = \sum_{\emptyset \neq u \subseteq [1 : s]} I_{\rho}(f_u^2). \quad (2)$$

Using these facts, the variance of the TMC estimator $I_{\rho}^{\text{TMC}}(f; N)$ can be analyzed as follows:

**Theorem 1** We have

$$E[I_{\rho}^{\text{TMC}}(f; N)] = I_{\rho}(f)$$

and

$$\mathbb{V}[I_{\rho}^{\text{TMC}}(f; N)] = \mathbb{V}[I_{\rho}(f; N)] + \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} (N - \ell) \sum_{\emptyset \neq u \subseteq [1:s-\ell]} I_{\rho}(f_u f_{u+\ell}),$$

where we write $u + \ell = \{j + \ell : j \in u\}$.

Note that, in the theorem, we write $I_{\rho}(f_u f_{u+\ell}) = \int_{\Omega} f_u(x_u) f_{u+\ell}(x_{u+\ell})\rho_{u}(x_u)\,dx_u$.

The readers should not be confused with $I_{\rho}(f_u f_{u+\ell}) = \int_{\Omega} f_u(x_u) f_{u+\ell}(x_{u+\ell})\rho_{u}(x_u)\,dx_u = 0$.

**Proof** The first assertion follows immediately from the linearity of expectation and the trivial equality $E[f(\tilde{x}_n)] = I_{\rho}(f)$. For the second assertion, by using the ANOVA decomposition of $f$, we have

$$\mathbb{V}[I_{\rho}^{\text{TMC}}(f; N)] = \mathbb{E}\left[[I_{\rho}^{\text{TMC}}(f; N)]^2\right] - \left(\mathbb{E}[I_{\rho}(f; N)]\right)^2$$

$$= \mathbb{E}\left[[I_{\rho}^{\text{TMC}}(f; N)]^2\right] - (I_{\rho}(f))^2$$

$$= \frac{1}{N^2} \sum_{m,n=1}^{N} \mathbb{E}[f(\tilde{x}_m)f(\tilde{x}_n)] - (I_{\rho}(f))^2$$

$$= \frac{1}{N^2} \sum_{n=1}^{N} \mathbb{E}[(f(\tilde{x}_n))^2] + \frac{2}{N^2} \sum_{m,n=1}^{N} \mathbb{E}[f(\tilde{x}_m)f(\tilde{x}_n)]$$

$$- (I_{\rho}(f))^2$$

$$= \frac{I_{\rho}(f^2)}{N} + \frac{2}{N^2} \sum_{m,n=1}^{N} \sum_{m>n} \mathbb{E}[f_u(\tilde{x}_{m,u})f_v(\tilde{x}_{n,v})]$$

$$- (I_{\rho}(f))^2.$$
where we reordered the sum over \( m \) and \( n \) with respect to the difference \( m - n \) in the last equality.

If \( m - n \geq s \), there is no overlapping of the components between \( \tilde{x}_{m,j} \) and \( \tilde{x}_{n,j} \). Because of the independence of samples, it follows from the first assertion of Lemma [1] that the inner sum over \( u \) and \( v \) above is given by

\[
\sum_{\emptyset \neq u, v \subseteq [1:s]} \mathbb{E}[f_u(\tilde{x}_{m,u})f_v(\tilde{x}_{n,v})] = \sum_{\emptyset \neq u, v \subseteq [1:s]} \mathbb{E}[f_u(\tilde{x}_{m,u})] \mathbb{E}[f_v(\tilde{x}_{n,v})] = 0.
\]

If \( \ell = m - n < s \), on the other hand, we have \( \tilde{x}_{n,j} = \tilde{x}_{m,j+\ell} \) for any \( j = 1, \ldots, s - \ell \). With this equality and the first assertion of Lemma [1] the inner sum over \( u \) and \( v \) becomes

\[
\sum_{\emptyset \neq u, v \subseteq [1:s]} \mathbb{E}[f_u(\tilde{x}_{m,u})f_v(\tilde{x}_{n,v})] = \sum_{\emptyset \neq u \subseteq [1:s-\ell]} \sum_{m=1}^{N} \mathbb{E}[f_u(\tilde{x}_{m,u})f_u(\tilde{x}_{m,u})] = \sum_{\emptyset \neq u \subseteq [1:s-\ell]} I_{\rho(u)}(f_u^2).
\]

Altogether we obtain

\[
\mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] = \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} \sum_{\emptyset \neq u \subseteq [1:s-\ell]} I_{\rho(u)}(f_u^2) + \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} \sum_{\emptyset \neq u \subseteq [1:s-\ell]} \mathbb{E}[f_u(\tilde{x}_{m,u})f_u(\tilde{x}_{n,v})] = \mathbb{V}[I_{\rho_s}^{\text{MC}}(f;N)] + \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} \sum_{\emptyset \neq u \subseteq [1:s-\ell]} I_{\rho(u)}(f_u^2) + \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} \sum_{\emptyset \neq u \subseteq [1:s-\ell]} \mathbb{E}[f_u(\tilde{x}_{m,u})f_u(\tilde{x}_{n,v})] = \mathbb{V}[I_{\rho_s}^{\text{MC}}(f;N)] + \frac{2}{N^2} \sum_{\ell=1}^{\min(s,N)-1} \sum_{\emptyset \neq u \subseteq [1:s-\ell]} I_{\rho(u)}(f_u^2).
\]

Thus we are done.

As is clear from Theorem [1] the TMC estimator is unbiased and maintains the canonical “one over square root of \( N \)” convergence. Moreover, the TMC estimator can be regarded as a variance reduction technique since the second term on the variance \( \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] \) can be negative, depending on the function.

Example 1 To illustrate the last comment, let us consider a simple test function \( f: \mathbb{R}^3 \to \mathbb{R} \) given by

\[
f(x, y, z) = x - y - z + xy - xz - yz,
\]

and let \( x, y, z \) be normally distributed independent random variables with mean 0 and variance 1. It is easy to see that

\[
f_1(x) = x, \quad f_2(y) = -y, \quad f_3(z) = -z, \\
f_{1,2}(x, y) = xy, \quad f_{1,3}(x, z) = -xz, \\
f_{2,3}(y, z) = -yz, \quad f_{1,2,3}(x, y, z) = 0.
\]

Then it follows that

\[
\mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] = \frac{1}{N} \sum_{\emptyset \neq u \subseteq [1:s]} I_{\rho(u)}(f_u^2) = \frac{6}{N},
\]

whereas, for \( N \geq 3 \), we have

\[
\mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] = \frac{6}{N} + \frac{2(N-2)}{N^2} I_{\rho_s}(f_1 f_3) + \frac{2(N-1)}{N^2} \times [I_{\rho_s}(f_1 f_2) + I_{\rho_s}(f_2 f_3) + I_{\rho_s}(f_1 f_2 f_3)]
\]

\[
= \frac{6}{N} - \frac{2(N-2)}{N^2} \frac{2(N-1)}{N^2} = \frac{2}{N} + \frac{6}{N^2}.
\]

Therefore the variance of the TMC estimator is almost one-third of the variance of the standard Monte Carlo estimator.

It is also possible that the variance of the TMC estimator increases compared to standard Monte Carlo, however, we show below that this increase is bounded.

2.2 Weighted \( L_2 \) space and tractability

Here we study the dependence of the variance \( \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] \) on the dimension \( s \). For this purpose, we first give a bound on \( \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f;N)] \). For \( 1 \leq \ell \leq s \), let

\[
\alpha_\ell(f) := \left( \sum_{\emptyset \neq u \subseteq [1:s]} I_{\rho(u)}(f_u^2) \right)^{1/2}.
\]
Then it follows from the decomposition (2) that
\[ I_{\rho_s}(f^2) - (I_{\rho_s}(f))^2 = \sum_{\ell=1}^{s} \sum_{\min_{j \leq \ell} I_{\rho_s}(f^2) = \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2, \]
resulting in an equality
\[ \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N)] = \frac{1}{N} \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2. \]
Using Theorem 1, the variance \( \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N)] \) is bounded above as follows.

**Corollary 1** We have
\[ \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N)] \leq \frac{1}{N} \left( \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2 \right)^2. \]

**Proof** For any \( \ell = 1, \ldots, s - 1 \), the Cauchy-Schwarz inequality leads to
\[
\sum_{\ell \neq u \subseteq [1 : s - \ell]} I_{\rho_{\ell u}}(f_u f_{u+\ell}) \\
\leq \sum_{\ell \neq u \subseteq [1 : s - \ell]} (I_{\rho_{\ell u}}(f_u^2))^{1/2} (I_{\rho_{\ell u}}(f_{u+\ell}^2))^{1/2} \\
= \sum_{\ell=1}^{s-\ell} \left( \sum_{\min_{j \leq \ell} I_{\rho_{\ell u}}(f_u^2)} \right)^{1/2} \left( \sum_{\min_{j \leq \ell} I_{\rho_{\ell u}}(f_{u+\ell}^2)} \right)^{1/2} \\
\leq \frac{1}{N} \sum_{\ell=1}^{s-1} \left( \sum_{\min_{j \leq \ell} I_{\rho_{\ell u}}(f_u^2)} \right)^{1/2} \left( \sum_{\min_{j \leq \ell} I_{\rho_{\ell u}}(f_{u+\ell}^2)} \right)^{1/2} \\
= \frac{1}{N} \left( \sum_{\ell=1}^{s-1} (\alpha_{\ell}(f))^2 \right)^{1/2} \left( \sum_{\ell=1}^{s-1} (\alpha_{\ell}(f))^2 \right)^{1/2} \\
\leq \frac{1}{N} \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2 \alpha_{\ell+1}(f).
\]
Applying this bound to the second assertion of Theorem 3, we obtain
\[ \mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N)] \leq \frac{1}{N} \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2 + \frac{2}{N^2} \sum_{\ell=1}^{s-1} (N-\ell) \sum_{v=1}^{s-\ell} \alpha_{\ell}(f) \alpha_{\ell+v}(f) \\
\leq \frac{1}{N} \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2 + \frac{2}{N} \sum_{\ell=1}^{s-1} \alpha_{\ell}(f) \sum_{\ell=1}^{s} \alpha_{\ell}(f) \\
= \frac{1}{N} \left( \sum_{\ell=1}^{s} (\alpha_{\ell}(f))^2 \right). \]

Using this result, we have
\[ \frac{\mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N)]}{\mathbb{V}[I_{\rho_s}^{\text{MC}}(f; N)]} \leq \frac{(\alpha_1(f) + \cdots + \alpha_s(f))^2}{\alpha_1^2(f) + \cdots + \alpha_s^2(f)} \leq s, \]
wherein, for the second inequality, the equality is attained if and only if \( \alpha_1(f) = \cdots = \alpha_s(f) \). Therefore, when we fix the number of samples, the variance of the TMC estimator can at most be \( s \) times larger than the variance of the standard Monte Carlo estimator.

Now let us consider the case \( s = t \) and assume, as discussed in the first section, that the computational time for the standard Monte Carlo estimator is proportional to \( Ns^2 \), whereas the computational time for the TMC estimator is proportional to \( Ns \log s \) (assuming that the main cost in evaluating \( f(x) = g(x \Lambda) \) lies in the computation of \( x \Lambda \)). When we fix the cost instead of the number of samples, we have
\[ N \log s \approx N_{\text{TMC}} \log s, \]
where \( \approx \) indicates that the terms should be of the same order, and so
\[ \frac{\mathbb{V}[I_{\rho_s}^{\text{TMC}}(f; N_{\text{TMC}})]}{\mathbb{V}[I_{\rho_s}^{\text{MC}}(f; N_{\text{MC}})]} \leq \frac{N_{\text{MC}}}{N_{\text{TMC}}} \cdot \frac{(\alpha_1(f) + \cdots + \alpha_s(f))^2}{\alpha_1^2(f) + \cdots + \alpha_s^2(f)} \propto \log s. \]
Thus, the variance of the TMC estimator for a given cost is at most \( s \log s \) times as large as the standard Monte Carlo estimator (up to some constant factor). On the other hand, if there is some decay of the importance of the ANOVA terms as the index of the variable increases, for instance, if the first few terms in \( \alpha_1(f) + \cdots + \alpha_s(f) \) dominate the sum, then the ratio \( (\alpha_1(f) + \cdots + \alpha_s(f))^2 / \alpha_1^2(f) + \cdots + \alpha_s^2(f) \) can be bounded independently of \( s \), leading to a gain in the efficiency of the TMC estimator. We observe such a behaviour in our numerical experiments below.

Following the idea from Sloan and Woźniakowski [1998], we now introduce the notion of a weighted \( L_2 \) space. Let \( (\gamma_n)_{n \in \mathbb{N}} \) be a sequence of the non-negative real numbers called weights. Then the weighted \( L_2 \) space is defined by
\[ F_{s, \gamma} = \{ f : \Omega^s \to \mathbb{R} \mid \| f \|_{\gamma, \gamma} \leq \infty \}, \]
where 
\[ ||f||_{s,\gamma} := \left( \sum_{u \subseteq [1:s]} \gamma_u^{-1} I_{\rho_u}(f_u^2) \right)^{1/2}. \]

For any subset \( u \) with \( \gamma_u = 0 \), we assume that the corresponding ANOVA term \( f_u \) is \( 0 \) and we formally set \( 0 / 0 = 0 \).

For a randomized algorithm using \( N \) function evaluations of \( f \) to estimate \( I_{\rho_u}(f) \), which we denote by \( \text{Alg}(f;N) \), let us consider the minimal cost to estimate \( I_{\rho_u}(f) \) with mean square error \( \epsilon^2 \) for any \( f \) in the unit ball of \( F_{s,\gamma} \):

\[ N(\epsilon, \text{Alg}) := \min \{ N \in \mathbb{Z}_{>0} \mid \epsilon^2(\text{Alg}, F_{s,\gamma}) \leq \epsilon^2 \}, \]

where

\[ \epsilon^2(\text{Alg}, F_{s,\gamma}) := \sup_{f \in F_{s,\gamma}} \frac{1}{||f||_{s,\gamma} \leq 1} \mathbb{E}\left[ (\text{Alg}(f;N) - I_{\rho_u}(f))^2 \right]. \]

We say that the algorithm \( \text{Alg} \) is

– a weakly tractable algorithm if

\[ \lim_{\epsilon^{-1} + s \to \infty} \frac{\ln N(\epsilon, \text{Alg})}{\epsilon^{-1} + s} = 0, \]

– a polynomially tractable algorithm if there exist non-negative constants \( C, p, q \), such that

\[ N(\epsilon, \text{Alg}) \leq C\epsilon^{-p}s^q \]

holds for all \( s = 1, 2, \ldots \), where \( p \) and \( q \) are called the \( \epsilon^{-1} \)-exponent and the \( s \)-exponent, respectively,

– a strongly polynomially tractable algorithm if \( \text{Alg} \) is a polynomially tractable algorithm with the \( s \)-exponent \( 0 \).

We refer to [Novak and Woźniakowski (2010)] for more information on the notion of tractability.

For instance, the standard Monte Carlo estimator \( I_{\rho_u}^{MC}(f;N) \) is a strongly polynomially tractable algorithm with \( \epsilon^{-1} \)-exponent \( 2 \) if

\[ \sup_{u \subseteq \mathbb{N}} \gamma_u < \infty \tag{3} \]

holds. This claim can be proven as follows:

\[ \mathbb{E}\left[ (I_{\rho_u}^{MC}(f;N) - I_{\rho_u}(f))^2 \right] \]

\[ = \mathbb{V}[I_{\rho_u}^{MC}(f;N)] = \frac{1}{N} \sum_{\emptyset \neq u \subseteq [1:s]} I_{\rho_u}(f_u^2) \]

\[ \leq \frac{1}{N} \left( \max_{u \subseteq [1:s]} \gamma_u \right) \left( \sum_{u \subseteq [1:s]} I_{\rho_u}(f_u^2) \right) \]

\[ = \frac{||f||_{s,\gamma}^2}{N} \max_{u \subseteq [1:s]} \gamma_u. \]

It follows that, in order to have

\[ \mathbb{E}\left[ (I_{\rho_u}^{MC}(f;N) - I_{\rho_u}(f))^2 \right] \leq \epsilon^2 \]

for any \( f \in F_{s,\gamma} \) with \( ||f||_{s,\gamma} \leq 1 \), we need \( N \geq \epsilon^{-2} \max_{u \subseteq [1:s]} \gamma_u \). Thus the minimal cost is bounded above by

\[ N(\epsilon, I_{\rho_u}^{MC}) \leq \epsilon^{-2} \max_{u \subseteq [1:s]} \gamma_u. \]

Given the condition (3), we see that \( N(\epsilon, I_{\rho_u}^{MC}) \) is bounded independently of the dimension \( s \) and the algorithm \( I_{\rho_u}^{MC} \) is strongly polynomially tractable with the \( \epsilon^{-1} \)-exponent \( 2 \).

The following theorem gives the necessary conditions on the weights \( (\gamma_u)_{u \subseteq \mathbb{N}} \) for the TMC estimator to be a weakly tractable algorithm, a polynomially tractable algorithm, or a strongly polynomially tractable algorithm.

**Theorem 2** The TMC estimator is

– a weakly tractable algorithm if

\[ \lim_{s \to \infty} \frac{1}{s} \ln \left( \sum_{\ell=1}^{s} \max_{u \subseteq [1:s]} \gamma_u \right) = 0, \]

– a polynomially tractable algorithm with the \( \epsilon^{-1} \)-exponent \( 2 \) if there exists \( q > 0 \) such that

\[ \sup_{s=1,2,\ldots} \frac{1}{s} \sum_{\ell=1}^{s} \max_{u \subseteq [1:s]} \gamma_u < \infty, \]

– a strongly polynomially tractable algorithm with the \( \epsilon^{-1} \)-exponent \( 2 \) if

\[ \sum_{\ell=1}^{\infty} \sup_{u \subseteq \mathbb{N}} \gamma_u < \infty. \]

**Proof** It follows from Corollary 4 and Hölder’s inequality for sums that

\[ \mathbb{E}\left[ (f_{\text{TMC}}^{MC}(f;N) - I_{\rho_u}(f))^2 \right] \]

\[ = \mathbb{V}[f_{\text{TMC}}^{MC}(f;N)] \leq \frac{1}{N} \left( \sum_{\ell=1}^{s} \max_{u \subseteq [1:s]} \gamma_u \right)^2 \]

\[ \leq \frac{1}{N} \left( \sum_{\ell=1}^{s} \max_{u \subseteq [1:s]} \gamma_u \right)^{1/2} \]

\[ \leq \frac{1}{N} \left( \sum_{\ell=1}^{s} \max_{u \subseteq [1:s]} \gamma_u \right)^{1/2} \]
Toeliptz Monte Carlo

Thus, the minimal cost to have

\[ \sum_{\ell=1}^{∞} \gamma_u^{-1} I_{ρ_u}(f_u^2) \] 1/2 2

\leq \frac{1}{N} \left( \sum_{\ell=1}^{∞} \max_{\emptyset \neq u \subseteq [1:s]} \gamma_u \right) \left( \sum_{\ell=1}^{∞} \sum_{\min_j \neq j = \ell}^{s} \gamma_u^{-1} I_{ρ_u}(f_u^2) \right)

\leq \frac{\|f\|^2}{N} \frac{1}{s} \sum_{\ell=1}^{s} \max_{\emptyset \neq u \subseteq [1:s]} \gamma_u.

Thus, the minimal cost to have

\[ \mathbb{E} \left[ (I^\text{TMC}_{ρ_u}(f; N) - I_{ρ_u}(f))^2 \right] \leq ε^2 \]

for any \( f \in F_{s,γ} \) with \( \|f\|_{s,γ} \leq 1 \) is bounded above by

\[ N(ε, I^\text{TMC}_{ρ_u}) \leq ε^{-2} \sum_{\ell=1}^{s} \max_{\emptyset \neq u \subseteq [1:s]} \gamma_u. \]

Let us consider the first assertion of the theorem. If the weights satisfy

**Lemma:**

\[ \lim_{s \to ∞} \frac{1}{s} \ln \left( \sum_{u \subseteq [1:s]} \max_{\min_j \neq j = \ell} \gamma_u \right) = 0, \]

then we have

\[ \lim_{ε^{-1} + s \to ∞} \ln N(ε, I^\text{TMC}_{ρ_u}) \]

\[ \leq \lim_{ε^{-1} + s \to ∞} \left( \ln ε^{-2} + s \ln \left( \sum_{\ell=1}^{s} \max_{\emptyset \neq u \subseteq [1:s]} \gamma_u \right) \right) = 0, \]

meaning that \( I^\text{TMC}_{ρ_u} \) is a weakly tractable algorithm. Since the second and third assertions can be shown similarly, we omit the proof.

For instance, if the weights satisfy \( γ_u \geq γ_v \) whenever \( u \subset v \), we always have

\[ \sup_{u \subseteq [1:s]} \gamma_u = γ(ℓ)\]

for any \( ℓ \in \mathbb{Z}_0^∞ \). Therefore, the necessary condition for the TMC estimator to be strongly polynomially tractable reduces to a simple summability:

\[ \sum_{ℓ=1}^{∞} γ(ℓ) < ∞. \]

It is obvious to see that the necessary condition for the TMC estimator to be weakly tractable is stronger than that for the standard Monte Carlo estimator to be strongly tractable. Whether we can weaken the necessary conditions for the TMC estimator given in Theorem 2 or not is an open question.

3 Numerical experiments

In order to see the practical performance of the TMC estimator, we conduct four kinds of numerical experiments. The first test case follows Section 4.1 of Dick et al. (2015), which considers generating quadrature points from a multivariate normal distribution with a general covariance matrix. The second test case, taken from Section 4.2 of Dick et al. (2015), deals with approximating linear functionals of solutions of one-dimensional PDE with “uniform” random coefficients. The third test case is an extension of the second test case to a one-dimensional PDE with “log-normal” random coefficients. Finally, in the fourth test case we consider another possible extension of the second test case, namely a two-dimensional PDE with uniform random coefficients. All computations are performed on a laptop with 1.6 GHz Intel Core i5 CPU and 8 GB memory.

For every test case, we carry out numerical experiments with various values of \( N \) and \( s \) using both the standard Monte Carlo estimator and the TMC estimator. For each pair of \( N \) and \( s \), we repeat computations \( R = 25 \) times independently and calculate the average computational time. For the latter three test cases, the variances of these estimators are measured by

\[ \frac{1}{R(R-1)} \sum_{r=1}^{R} \left( I^\bullet_{ρ_u}(f; N) - I^\text{TMC}_{ρ_u}(f; N) \right)^2 \]

with

\[ I^\bullet_{ρ_u}(f; N) = \frac{1}{R} \sum_{r=1}^{R} I^\bullet_{ρ_u}(f; N), \]

for \( \bullet \in \{ \text{MC, TMC} \} \), where \( I^\bullet_{ρ_u}(f; N) \) denotes the \( r \)-th realization of the estimator \( I^\bullet_{ρ_u}(f; N) \).

3.1 Generating points from multivariate Gaussian

Generating quadrature points from the multivariate normal distribution \( N(μ, Σ) \) with mean vector \( μ \in \mathbb{R}^s \) and covariance matrix \( Σ \in \mathbb{R}^{s \times s} \) is ubiquitous in scientific computation. The standard procedure is as follows (Devroye 1986, Chapter XI.2): Let \( A \in \mathbb{R}^{s \times s} \) be a matrix which satisfies \( A^T A = Σ \). For instance, the Cholesky decomposition gives such \( A \) in an upper triangular form for any symmetric positive-definite matrix \( Σ \). Using this decomposition, we can generate a point \( y \sim N(μ, Σ) \) by first generating \( x = (x_1, \ldots, x_s) \) with \( x_j \sim N(0, 1) \) and then transforming \( x \) by

\[ y = μ + x A. \]
### Table 1: Average times (in seconds) to generate normally distributed points with a random covariance matrix for various values of \( N \) and \( s \) using the standard Monte Carlo method and the TMC method.

| \( N \) | \( s = 128 \) | \( s = 256 \) | \( s = 512 \) | \( s = 1024 \) | \( s = 2048 \) |
|-------|-------------|-------------|-------------|-------------|-------------|
| stdMC | 1024        | 0.041       | 0.192       | 1.369       | 9.613       | –           |
| TMC   | 0.165       | 0.239       | 0.403       | 0.872       | –           | –           |
| saving| 0.250       | 0.800       | 3.399       | 11.021      | –           | –           |
| stdMC | 2048        | 0.083       | 0.367       | 2.077       | 20.154      | 217.162     |
| TMC   | 0.322       | 0.465       | 0.923       | 1.794       | 3.599       | –           |
| saving| 0.259       | 0.790       | 2.251       | 11.234      | 60.342      | –           |
| stdMC | 4096        | 0.154       | 0.740       | 5.551       | 45.287      | 446.974     |
| TMC   | 0.586       | 0.981       | 1.909       | 3.407       | 7.112       | –           |
| saving| 0.286       | 0.754       | 2.908       | 13.291      | 62.851      | –           |
| stdMC | 8192        | 0.288       | 1.669       | 10.332      | 87.988      | 830.648     |
| TMC   | 1.169       | 1.868       | 3.110       | 6.204       | 14.561      | –           |
| saving| 0.247       | 0.893       | 4.922       | 14.183      | 57.034      | –           |
| stdMC | 16384       | 0.600       | 3.255       | 19.764      | 199.792     | 1920.420    |
| TMC   | 2.434       | 3.750       | 7.088       | 14.073      | 28.082      | –           |
| saving| 0.271       | 0.868       | 2.788       | 11.355      | 64.826      | –           |
| stdMC | 32768       | 1.078       | 5.627       | 33.011      | 331.747     | 3048.755    |
| TMC   | 5.486       | 9.098       | 15.164      | 25.950      | 57.171      | –           |
| saving| 0.197       | 0.618       | 2.177       | 12.784      | 63.822      | –           |

Even if the matrix \( A \) does not have any further structure, a set of quadrature points can be generated in a fast way by following Algorithm 1.

For our experiments, we fix \( \mu = (0, \ldots, 0) \) and choose \( A \) randomly such that \( A \) is a random upper triangular matrix with positive diagonal entries. Table 1 shows the average computational times for various values of \( N \) and \( s \). As the theory predicts, the computational time for the standard Monte Carlo (stdMC) estimator scales as \( Ns^2 \), whereas that for the TMC estimator it approximately scales as \( Ns \log s \). For low-dimensional cases up to \( s = 256 \), the stdMC estimator is faster to compute than the TMC estimator. However, as the dimension \( s \) increases, the relative speed of the TMC estimator also increases. For the case \( s = 2048 \), the TMC is approximately 60 times faster than the stdMC. This result indicates that the TMC estimator is useful in high-dimensional settings for generating normally distributed quadrature points for a general covariance matrix.

#### 3.2 1D differential equation with uniform random coefficients

Let us consider the ODE

\[
-\frac{d}{dx} \left( a(x,y) \frac{d}{dx} u(x,y) \right) = g(x) \equiv 1
\]

for \( x \in (0,1) \) and \( y \in \left[ \frac{1}{2}, \frac{1}{2} \right]^N \)

\[ u(x,y) = 0 \quad \text{for} \quad x = 0,1 \]

\[ a(x,y) = 2 + \sum_{j=1}^{\infty} y_j \sin(2\pi j x) \]

In order to solve this ODE approximately with the finite element method, we consider a system of hat functions

\[ \phi_m(x) = \begin{cases} 
(x-x_{m-1})M & \text{if} \ x_{m-1} \leq x \leq x_m, \\
(x_{m+1}-x)M & \text{if} \ x_m \leq x \leq x_{m+1}, \\
0 & \text{otherwise,}
\end{cases} \]

for \( m = 1, \ldots, M-1 \) over \( M+1 \) equi-distributed nodes \( x_m = m/M \) for \( m = 0,1, \ldots, M \), and truncate the infinite sum appearing in the random field \( a \) by the first \( s \) terms. Therefore, we have three different parameters \( N, M \) and \( s \).

Given the boundary condition on \( u \), the approximate solution \( u_M \) of the ODE for \( y_1,\ldots,y_s \sim U[-1/2,1/2] \) is given by

\[ u_M = \sum_{m=1}^{M-1} \hat{u}_m \phi_m(x), \]

with

\[ (\hat{u}_1, \ldots, \hat{u}_{M-1}) \cdot B(y_1, \ldots, y_s) = (\hat{g}_1, \ldots, \hat{g}_{M-1}), \quad (4) \]

for the symmetric stiffness matrix \( B \) depending on \( y_1, \ldots, y_s \) and the forcing vector with entries \( \hat{g}_m = \int_0^1 g(x) \phi_m(x) \, dx = 1/M. \) The entries of the matrix \( B = (b_{k,\ell})_{k,\ell} \in \mathbb{R}^{(M-1) \times (M-1)} \) are given by

\[ b_{k,\ell} = \int_0^1 a(x, (y_1, \ldots, y_s, 0, 0, \ldots)) \phi_k'(x) \phi_\ell'(x) \, dx \]
\[
= 2 \int_0^1 \phi_k(x) \phi_\ell'(x) \, dx \\
+ \sum_{j=1}^s y_j \int_0^1 \frac{\sin(2\pi j x)}{j^2} \phi_k(x) \phi_\ell'(x) \, dx \\
=: a_{k,\ell}^{(0)} + \sum_{j=1}^s y_j a_{k,\ell}^{(j)}.
\]

Hence, by letting \( A^{(j)} = (a_{k,\ell}^{(j)})_{k,\ell} \in \mathbb{R}^{(M-1) \times (M-1)} \) for \( j = 0, 1, \ldots, s \), we have
\[
B = A^{(0)} + \sum_{j=1}^s y_j A^{(j)}.
\]

Here we note that every entry of \( A^{(j)} \) can be explicitly calculated as
\[
a_{k,\ell}^{(0)} = \begin{cases} 
4M & \text{if } k = \ell, \\
-2M & \text{if } |k - \ell| = 1, \\
0 & \text{otherwise},
\end{cases}
\]
and
\[
a_{k,\ell}^{(j)} = \begin{cases} 
\frac{M^2}{\pi j^2} \sin \left( \frac{2\pi j}{M} \right) \sin \left( \frac{2\pi j}{M} \right) & \text{if } k = \ell, \\
\frac{M^2}{\pi j^2} \sin \left( \frac{\pi j}{M} \right) \sin \left( \frac{\pi j}{M} \right) & \text{if } |k - \ell| = 1, \\
0 & \text{otherwise},
\end{cases}
\]
for \( 1 \leq j \leq s \). Since each matrix \( A^{(j)} \) is tridiagonal, the LU decomposition requires only \( O(M) \) computational time to solve the system of linear equations \( \mathbf{B} \mathbf{X} = \mathbf{Y} \). This way, it is clear that computing the matrix \( B \) for \( N \) Monte Carlo samples on \((y_1, \ldots, y_s)\) is computationally dominant for the whole process, and as shown in Section 3.2 of Dick et al. (2015), the standard Monte Carlo method requires \( O(NM) \) arithmetic operations, whereas Algorithm 1 can reduce them to \( O(NM \log s) \). It is expected that the TMC estimator brings substantial computational cost savings, particularly for large \( s \). For our experiments, we estimate the expectation of \( u(1/2, \cdot) \).

Table 2 shows the results for various values of \( N, M \) and \( s \). Since computations are repeated 25 times independently for each pair, here we report the average of the estimation values and the variance of the estimator for two methods. By comparing the mean values computed by two methods, we can confirm that the TMC estimator is also unbiased (just as the stdMC estimator is). The variances for both of the estimators decay with the rate \( O(1/N) \), whereas the magnitude for the TMC estimator is approximately 2–5 larger than the stdMC estimator. On the other hand, the computational time for the stdMC estimator increases with \( N \) (equivalently, with \( s \)) significantly faster than the TMC estimator.

This increment behavior of the computation times indicates that computation of the stiffness matrix is the most computationally dominant part in this computation, and so the TMC estimator is quite effective in reducing the computation time.

As is standard (see for instance Chapter 8 of Owen (2013)), we measure the relative efficiency of the TMC estimator compared to the stdMC estimator by the ratio
\[
\frac{T_{\text{TMC}} \sigma_{\text{MC}}^2}{T_{\text{MC}} \sigma_{\text{TMC}}^2},
\]
where \( T \) and \( \sigma^2 \) denote the computational time spent and the variance, respectively, for the estimators \( \bullet \in \{\text{MC, TMC}\} \). As shown in the rightmost column of Table 2, the relative efficiency is smaller than 1 for low-dimensional cases, which means that we do not gain any benefit from using the TMC estimator. However, because of the substantial computational time savings, the efficiency increases significantly for large \( s \) where it goes well beyond 1.

### 3.3 1D differential equation in the log-normal case

Let us move on to an ODE with the log-normal random coefficients:
\[
- \frac{d}{dx} \left( a(x, y) \frac{d}{dx} u(x, y) \right) = g(x) \equiv 1
\]
for \( x \in (0, 1) \) and \( y_j \sim \text{i.i.d. } N(0, 1) \)
\[
u(x, y) = 0 \quad \text{for } x = 0, 1
\]
\[
a(x, y) = \exp \left( \sum_{j=1}^{\infty} y_j \sin(2\pi j x) / j^2 \right),
\]
Similarly to the uniform case, we truncate the infinite sum appearing in the random field \( a \) by the first \( s \) terms. A similar test case was also used in Section 4.3 of Dick et al. (2013).

Now, as the entries of the stiffness matrix \( B = (b_{k,\ell})_{k,\ell} \in \mathbb{R}^{(M-1) \times (M-1)} \) cannot be expressed simply as a linear sum of \( y_1, y_2, \ldots \), we need to approximate the integral by using some quadrature formulas, except for the case \( |k - \ell| \geq 2 \) where we just have \( b_{k,\ell} = 0 \).

\[2\] We point out that in Section 4.3 of Dick et al. (2013) the authors replaced the normal distribution by a uniform distribution. However, a normal distribution could have been used in Dick et al. (2012) Section 4.3 as well, for instance by shifting the QMC points in \([0, 1]^s\) in the quadrature rule by \((1/2N), \ldots, (1/2N)\) and applying the inverse CDF to the shifted points. Doing so avoids the point \((0, \ldots, 0) \in [0, 1]^s\) which would get transformed to \((-\infty, \ldots, -\infty) \). Note that this shift does not affect the fast QMC matrix vector product, which can still be applied in the usual way.
Table 2 Estimating the expectation of $u(1/2, \cdot)$ with various values of $N, M$ and $s$ using the standard Monte Carlo method and the TMC method for the uniform case. The average estimate, the variance of the estimator and the average computational time (in seconds) are shown for each method. The efficiency is defined by the ratio of the product of the variance and the computational time between two methods.

|       | stdMC                | TMC                     |
|-------|----------------------|-------------------------|
|       | mean | variance | time   | mean | variance | time   | efficiency |
| $N$   | $N = M = s$           |                        |
| 64    | 0.066 | 2.00·10^{-8} | 0.002  | 0.066 | 4.39·10^{-8} | 0.023  | 0.046 |
| 128   | 0.065 | 4.20·10^{-9} | 0.020  | 0.065 | 1.47·10^{-8} | 0.071  | 0.081 |
| 256   | 0.064 | 2.00·10^{-9} | 0.189  | 0.064 | 7.11·10^{-9} | 0.221  | 0.240 |
| 512   | 0.064 | 1.01·10^{-9} | 1.455  | 0.064 | 2.27·10^{-9} | 0.823  | 0.781 |
| 1024  | 0.064 | 5.75·10^{-10}| 19.720 | 0.064 | 8.45·10^{-10}| 3.202  | 4.190 |
| 2048  | 0.064 | 1.96·10^{-10}| 217.303| 0.064 | 5.51·10^{-10}| 12.069 | 6.405 |
| 4096  | 0.064 | 9.50·10^{-11}| 2367.836| 0.064 | 2.36·10^{-10}| 57.402 | 16.612 |
| $N$   | $N = M^2 = s$         |                        |
| 256   | 0.076 | 3.34·10^{-8} | 0.005  | 0.076 | 1.17·10^{-7} | 0.013  | 0.125 |
| 512   | 0.076 | 7.88·10^{-9} | 0.042  | 0.072 | 2.44·10^{-8} | 0.031  | 0.441 |
| 1024  | 0.070 | 2.77·10^{-9} | 0.298  | 0.070 | 4.98·10^{-9} | 0.087  | 1.892 |
| 2048  | 0.068 | 4.78·10^{-10}| 1.871  | 0.068 | 2.07·10^{-9} | 0.235  | 1.838 |
| 4096  | 0.066 | 3.66·10^{-10}| 10.965 | 0.066 | 6.48·10^{-10}| 0.766  | 8.088 |
| 8192  | 0.066 | 6.70·10^{-11}| 75.863 | 0.066 | 2.28·10^{-10}| 2.390  | 9.329 |
| 16384 | 0.065 | 2.20·10^{-11}| 963.569| 0.065 | 9.70·10^{-11}| 7.033  | 31.073 |
| $N$   | $N = 2M^2 = 2s$      |                        |
| 64    | 0.069 | 3.06·10^{-8} | 0.001  | 0.070 | 9.98·10^{-8} | 0.017  | 0.013 |
| 128   | 0.066 | 6.16·10^{-9} | 0.004  | 0.066 | 2.36·10^{-8} | 0.044  | 0.026 |
| 256   | 0.065 | 2.15·10^{-9} | 0.041  | 0.065 | 9.24·10^{-9} | 0.135  | 0.070 |
| 512   | 0.064 | 7.61·10^{-10}| 0.326  | 0.064 | 2.58·10^{-9} | 0.449  | 0.214 |
| 1024  | 0.064 | 3.86·10^{-10}| 2.681  | 0.064 | 8.95·10^{-10}| 1.413  | 0.818 |
| 2048  | 0.064 | 1.72·10^{-10}| 35.731 | 0.064 | 5.67·10^{-10}| 5.648  | 1.919 |
| 4096  | 0.064 | 7.84·10^{-11}| 444.156| 0.064 | 2.41·10^{-10}| 23.480 | 6.144 |
| $N$   | $N = 4M^2 = 4s$      |                        |
| 512   | 0.076 | 1.39·10^{-8} | 0.011  | 0.076 | 4.37·10^{-8} | 0.026  | 0.137 |
| 1024  | 0.072 | 3.25·10^{-9} | 0.064  | 0.072 | 9.13·10^{-9} | 0.062  | 0.364 |
| 2048  | 0.070 | 7.20·10^{-10}| 0.534  | 0.070 | 3.26·10^{-9} | 0.183  | 0.645 |
| 4096  | 0.068 | 2.23·10^{-10}| 3.204  | 0.068 | 9.55·10^{-10}| 0.524  | 1.427 |
| 8192  | 0.066 | 1.41·10^{-10}| 20.888 | 0.066 | 3.01·10^{-10}| 1.596  | 6.133 |
| 16384 | 0.066 | 3.60·10^{-11}| 159.880| 0.066 | 1.22·10^{-10}| 4.618  | 10.216 |
| 32768 | 0.065 | 1.02·10^{-11}| 2023.179| 0.065 | 5.80·10^{-11}| 13.586 | 26.144 |

Denoting the quadrature nodes and the corresponding weights by $x_{1,k,\ell}, \ldots, x_{I,k,\ell}$ and $\omega_{1,k,\ell}, \ldots, \omega_{I,k,\ell}$, the entry $b_{k,\ell}$ is approximated by

$$b_{k,\ell} \approx \tilde{b}_{k,\ell} = \frac{1}{I} \sum_{i=1}^{I} \omega_{i,k,\ell} \exp(\theta_{i,k,\ell}) \phi_{\ell}^i(x_{i,k,\ell}) \phi_{\ell}(x_{i,k,\ell}),$$

where

$$\theta_{i,k,\ell} = \sum_{j=1}^{s} \frac{y_j \sin(2\pi j x_{i,k,\ell})}{j^2},$$

for the case $|k - \ell| \leq 1$. As stated in Section 3.2 of Dick et al. (2015), computing $\theta_{i,k,\ell}$ for all $1 \leq i \leq I$ and all $N$ Monte Carlo samples on $(y_1, \ldots, y_s)$ can be done in a fast way by using the TMC estimator, requiring $O(INM \log s)$ arithmetic operations. On the other hand, we need $O(INMs)$ arithmetic operations when using the standard Monte Carlo estimator. In this paper we apply the 3-point closed Newton-Cotes formula with nodes at $x_{k-1}, x_k, x_{k+1}$ if $k = \ell$ and the 2-point closed one with nodes at $x_{(k+\ell-1)/2}, x_{(k+\ell+1)/2}$ if $|k - \ell| = 1$. Again we estimate the expectation of $u(1/2, \cdot)$.

Table 3 shows the results for various values of $N, M$ and $s$. Similarly to the uniform case, we see that the TMC estimator is unbiased as the mean values agree well with the results for the stdMC estimator. In this case, however, the variances for both of the estimators do not necessarily decay with the rate $O(1/N)$. This is possible because we increase $M$ and $s$ simultaneously with $N$, which may lead to an increment of the variance of $u_{\Lambda}(1/2, \cdot)$ in a non-asymptotic range of $N$. As $N$ increases further, it is expected that the variance of $u_{\Lambda}(1/2, \cdot)$ stays almost the same and that the variances for both of the estimators tend to decay with the rate $O(1/N)$. Moreover, it can be seen from the table that the magnitude of the variance for the TMC estimator is comparable to that of the stdMC estimator for many choices of $N, M, s$. As expected, the computational time
for the stdMC estimator increases with $s$ significantly faster than the TMC estimator, and it is clear that computation of the stiffness matrix takes most of the computational time, even for the log-normal case. The relative efficiency of the TMC estimator over the stdMC estimator gets larger as $N$ (or, equivalently $s$) increases.

3.4 2D differential equation with random coefficients

Following Section 4 of [Dick et al., 2016], our last example considers the following two-dimensional ODE with the uniform random coefficients:

$- \nabla \cdot (a(x, y) \nabla u(x, y)) = g(x) \equiv 100x_1$

for $x \in [0, 1]^2$ and $y \in \left[\frac{1}{N}, \frac{1}{2}\right]^N$.

$u(x, y) = 0$ for $x \in \partial ([0, 1]^2)$

$a(x, y) = 1 + \sum_{j=1}^{\infty} y_j \sin(\pi k_{1,j} x_1) \sin(\pi k_{2,j} x_2) / (k_{1,j}^2 + k_{2,j}^2)^2$

Here the elements $(k_{1,j}, k_{2,j})_j$ are ordered in such a way that $\{(k_{1,j}, k_{2,j}) : j \in \mathbb{N}\} = \mathbb{N} \times \mathbb{N}$ and

$1 / (k_{1,j}^2 + k_{2,j}^2)^2 \geq 1 / (k_{1,j+1}^2 + k_{2,j+1}^2)^2$ for all $j \in \mathbb{N}$.

In cases where equality holds, the ordering is arbitrary.

We solve this ODE by a finite element discretization.

For the boundary condition on $u$, we exclude the basis functions along the boundary and use a set of local piecewise linear hat functions $\{\phi_{p,q} : 0 < p, q < M\}$ as the system of basis functions. The basis function $\phi_{p,q}$ has center at $(p/M, q/M)$ and the support is given by the following hexagon:

The approximate solution $u_M$ of the ODE in this setting is given by

$u_M = \sum_{p,q=1}^{M-1} \hat{u}_{p,q} \phi_{p,q}(x_1, x_2)$,
Table 3 Estimating the expectation of $u(1/2, \cdot)$ with various values of $N, M$ and $s$ using the standard Monte Carlo method and the TMC method for the log-normal case.

| $N$ | stdMC mean | stdMC variance | stdMC time | TMC mean | TMC variance | TMC time | TMC efficiency |
|-----|------------|----------------|------------|----------|--------------|----------|---------------|
| 64  | 0.018      | 8.34·10^{-8}  | 0.073      | 0.018    | 2.97·10^{-7} | 0.079    | 0.260         |
| 128 | 0.018      | 1.17·10^{-8}  | 0.580      | 0.018    | 1.76·10^{-7} | 0.171    | 0.227         |
| 256 | 0.018      | 1.55·10^{-8}  | 4.650      | 0.018    | 7.74·10^{-8} | 0.549    | 1.694         |
| 512 | 0.018      | 2.11·10^{-8}  | 36.640     | 0.018    | 3.35·10^{-8} | 2.260    | 10.230        |
| 1024| 0.018      | 2.65·10^{-8}  | 330.415    | 0.018    | 6.63·10^{-8} | 10.385   | 11.956        |
| 2048| 0.018      | 1.23·10^{-8}  | 2391.495   | 0.018    | 3.06·10^{-8} | 41.504   | 23.262        |
| 4096| 0.018      | 2.33·10^{-8}  | 20739.823  | 0.018    | 2.03·10^{-8} | 182.282  | 131.012       |

| $N$ |
|-----|
| 256 |
| 512 |
| 1024|
| 2048|
| 4096|

| $N = M = s$ |
|-------------|
| 64  | 0.017 | 8.08·10^{-9} | 0.301 | 0.017 | 1.24·10^{-8} | 0.032 | 6.069 |
| 128 | 0.018 | 1.44·10^{-8} | 1.680 | 0.018 | 9.75·10^{-9} | 0.093 | 26.813 |
| 256 | 0.018 | 2.68·10^{-8} | 9.762 | 0.018 | 1.24·10^{-8} | 0.277 | 76.387 |
| 512 | 0.018 | 8.39·10^{-9} | 56.480 | 0.018 | 5.37·10^{-8} | 0.615 | 14.695 |
| 1024| 0.018 | 1.89·10^{-8} | 317.294 | 0.018 | 2.98·10^{-8} | 2.212 | 91.011 |
| 2048| 0.018 | 6.15·10^{-9} | 1865.110 | 0.018 | 3.27·10^{-8} | 7.094 | 494.909 |
| 4096| 0.018 | 1.91·10^{-8} | 9964.289 | 0.018 | 2.35·10^{-8} | 16.548 | 473.083 |

| $N = 2M = 2s$ |
|----------------|
| 64  | 0.018 | 1.16·10^{-7} | 0.021 | 0.018 | 1.83·10^{-7} | 0.046 | 0.288 |
| 128 | 0.018 | 9.58·10^{-8} | 0.163 | 0.018 | 1.46·10^{-7} | 0.126 | 0.844 |
| 256 | 0.018 | 5.09·10^{-8} | 1.279 | 0.018 | 7.30·10^{-8} | 0.394 | 2.264 |
| 512 | 0.018 | 2.40·10^{-8} | 9.997 | 0.018 | 3.33·10^{-8} | 1.089 | 6.616 |
| 1024| 0.018 | 9.53·10^{-8} | 82.822 | 0.018 | 6.00·10^{-8} | 4.914 | 26.758 |
| 2048| 0.018 | 1.18·10^{-8} | 644.228 | 0.018 | 2.97·10^{-8} | 19.265 | 13.245 |
| 4096| 0.018 | 2.45·10^{-9} | 5127.910 | 0.018 | 1.72·10^{-8} | 73.837 | 58.506 |

| $N = 2M^2 = 2s$ |
|-------------------|
| 512               |
| 1024              |
| 2048              |
| 4096              |

the one-dimensional log-normal case, the magnitude for the TMC estimator is comparable to that of the stdMC estimator. The computational time for the stdMC estimator increases with $s$ much faster than the TMC estimator, resulting in a substantial relative efficiency of the TMC estimator over the stdMC estimator for larger $s$.

4 Conclusion

Motivated by applications to partial differential equations with random coefficients, we introduced the Toeplitz Monte Carlo estimator in this paper. The theoretical analysis of the TMC estimator shows that it is unbiased and the variance converges with the canonical $1/N$ rate. From the viewpoint of tractability in the weighted $L_2$ space, the TMC estimator requires a stronger condition on the weights than the standard Monte Carlo estimator to achieve strong polynomial tractability. Through a series of numerical experiments for PDEs with random coefficients, we observed that the TMC estimator is quite effective in reducing necessary computational times and the relative efficiency over the standard Monte Carlo estimator is substantial, particularly for high-dimensional settings.

We leave the following topics open for future research:

- Combination with variance reduction techniques. In our numerical experiments for the one-dimensional uniform case, the variance of the TMC estimator tends to be much larger than the standard Monte Carlo estimator. To address this issue, it would be reasonable to consider applying some variance reduction techniques to the TMC estimator such that the resulting algorithm still allows for a fast matrix-vector multiplication. In particular, it would be interesting to design a variance reduction technique...
Table 4 Estimating the expectation of \(\phi(z,\omega)\) with various values of \(N, M\) and \(s\) using the standard Monte Carlo method and the TMC method for the two-dimensional uniform case.

| \(N\) | stdMC \(N = M^2 = s\) | TMC \(N = M^2 = s\) | mean | variance | time | mean | variance | time | efficiency |
|-------|-------------------------|----------------------|-----|----------|------|-----|----------|------|---------|
| 16    | 3.516 1.09-10^{-2}     | 3.514 7.17-10^{-4}  | 0.0002 | 3.514 | 7.17-10^{-4} | 0.0020 | 0.119 |
| 64    | 3.647 2.16-10^{-4}     | 3.643 2.43-10^{-4}  | 0.009  | 3.643 | 2.43-10^{-4} | 0.027 | 0.302 |
| 256   | 3.676 8.03-10^{-5}     | 3.678 8.84-10^{-5}  | 1.248  | 3.678 | 8.84-10^{-5} | 0.406 | 2.792 |
| 1024  | 3.685 1.89-10^{-5}     | 3.686 1.25-10^{-5}  | 127.366 | 3.686 | 1.25-10^{-5} | 8.369 | 22.925 |
| 4096  | 3.688 3.62-10^{-6}     | 3.688 3.74-10^{-6}  | 14773.594 | 3.688 | 3.74-10^{-6} | 228.490 | 62.562 |

which reduces the term

\[
\sum_{\ell=1}^{\min(s,N)-1} (N - \ell) \sum_{\emptyset \neq u \subseteq \{1:s-\ell\}} \mathbb{E}_{\rho(u)}(f_u f_{u+\ell}).
\]

- Multilevel Toeplitz Monte Carlo (MLTMC). Recently a number of papers on applying multilevel Monte Carlo (MLMC) methods from [Giles (2008)] to PDEs with random coefficients, such as [Cliffe et al. (2011)] and [Teckentrup et al. (2013)], have appeared. By combining the TMC estimator with MLMC, the dependence of the total computational complexity not only on the truncation dimension \(s\) but also on the discretization parameter \(M\) can be possibly weakened.

- Applications to different areas. Although this work has been originally motivated by PDEs with random coefficients, the TMC estimator itself is more general and can be applied in different contexts as well. Since generating points from multivariate normal distribution is quite common, for instance, in financial engineering, operations research and machine learning, one may apply the TMC estimator also to those areas.

References

Cliffe, K. A., Giles, M. B., Scheichl, R., Teckentrup, A. L.: Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. Comput. Vis. Sci. 14, 3–15 (2011)

Devroye, L.: Non-Uniform Random Variate Generation. Springer, New York (1986)

Dick, J., Kuo, F. Y., Le Gia, Q. T., Schwab, C.: Fast QMC matrix-vector multiplication. SIAM J. Comput. 37, A1436–A1450 (2015)

Dick, J., Kuo, F. Y., Le Gia, Q. T., Schwab, C.: Multilevel higher order QMC Petrov-Galerkin discretization for affine parametric operator equations. SIAM J. Numer. Anal. 54, 2541–2568 (2016)

Feischl, M., Kuo, F. Y., Sloan, I. H.: Fast random field generation with \(H\)-matrices. Numer. Math. 140, 639–676 (2018)

Frigo, M., Johnson, S. G.: The design and implementation of FFTW3. Proc. IEEE 93, 216–231 (2005)

Giles, M. B.: Multilevel Monte Carlo path simulation. Oper. Res. 56, 607–617 (2008)

Giles, M. B., Kuo, F. Y., Sloan, I. H., Waterhouse, B. J.: Quasi-Monte Carlo for finance applications. ANZIAM J. 50, C308–C323 (2008)

Hoeffding, W.: A class of statistics with asymptotically normal distribution. Ann. Math. Statist. 19, 293–325 (1948)

Kuo, F. Y., Sloan, I. H., Wasilkowski, G., Woźniakowski, H.: On decompositions of multivariate functions. Math. Comp. 79, 953–966 (2010)
Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems. Volume II: Standard Information for Functionals. EMS Tracts in Mathematics, 12. European Mathematical Society, Zürich (2010)

Owen, A. B.: Monte Carlo Theory, Methods and Examples. [http://statweb.stanford.edu/~owen/mc/](http://statweb.stanford.edu/~owen/mc/) (2019) [Last accessed 4 March 2020]

Saad, Y.: Iterative Methods for Sparse Linear Systems. 2nd Ed., SIAM, Philadelphia (2003)

Sloan, I. H., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high-dimensional integrals? J. Complexity 14, 1–33 (1998)

Sobol’, I. M.: Sensitivity estimates for nonlinear mathematical models. Math. Mod. Comp. Exp. 1, 407–414 (1993)

Teckentrup, A. L., Scheichl, R., Giles, M. B., Ullmann, E.: Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. Numer. Math. 125, 569–600 (2013)