A Practical Multilevel Quasi-Monte Carlo Method for PDEs with Random Coefficients

P. Robbe[†] D. Nuyens[†] S. Vandewalle[†]

Abstract

We present a Multilevel Quasi-Monte Carlo algorithm for the solution of elliptic partial differential equations with random coefficients and inputs. By combining the multilevel sampling idea with randomly shifted rank-1 lattice rules, the algorithm constructs an estimator for the expected value of some functional of the solution. The error analysis of this estimator provides a formula for the optimal number of samples at each level. The efficiency of the method is illustrated on a three-dimensional subsurface flow problem with lognormal diffusion coefficient. This example is particularly challenging because of the small correlation length, and thus the large number of uncertainties that must be included. We numerically show that it is possible to achieve a cost almost inversely proportional to the requested tolerance on the root-mean-square error.

1 Introduction

In a mathematical model for a real-life process, the parameters are often unknown or subject to uncertainty. Many models take the form of a partial differential equation (PDE), and the parameters can be random variables, processes or fields. Monte Carlo (MC) techniques are a standard method for solving such stochastic PDEs, however, they are often considered impractical due to their expense. Recently, the Multilevel Monte Carlo (MLMC) method has been successfully applied to many different problems, showing significant gains over classical MC, see [7] or [2]. The MLMC method is a combination of classical Monte Carlo sampling and a multigrid idea. The method is quite general, and does not impose strong restrictions on the estimators used to compute the contributions on each of these grids. This has resulted in many possible variants of the method, such as level-dependent estimators [22] and Quasi-Monte Carlo (QMC) estimators [9]. Here, we present a practical QMC variant, based on a heuristic for the optimal number of samples at each grid. We apply our algorithm to a subsurface flow example in three dimensions with a small correlation length and several thousand uncertainties.

The outline of this paper is as follows. After introducing the model problem below, section 3 introduces the classical Multilevel Monte Carlo method. In section 4, we discuss the basics of Quasi-Monte Carlo quadrature and introduce so-called *rank-1 lattice rules*. Next, we formulate a new Multilevel Quasi-Monte Carlo method and a complexity theorem that bounds the cost of this estimator. Finally, in section 6, we present some numerical results that demonstrate the superiority of the new Multilevel Quasi-Monte Carlo method.

2 The Model Problem

A central problem in groundwater studies is the steady-state flow in random porous media. This flow can be described by the random elliptic partial differential equation

$$-\nabla \cdot (k(\boldsymbol{x};\omega)\nabla p(\boldsymbol{x};\omega)) = f(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in D,\ \omega \in \Omega, \tag{1}$$

where D is a bounded domain in $\mathbb{R}^d$, with $d = (1, 2, 3)$, and Ω a sample space. The diffusion coefficient $k(\boldsymbol{x};\omega)$ represents the permeability of the porous medium and is modelled as a random field on $D \times \Omega$. A random field is the generalisation of a stochastic process into multiple (spatial) dimensions. The solution $p(\boldsymbol{x};\omega)$ represents the unknown hydrostatic pressure head and is itself also a random field on $D \times \Omega$.

A commonly used model for $k(\boldsymbol{x};\omega)$ is a lognormal distribution, $k = \log Z$, where Z is an underlying Gaussian field. The spatial covariance between two locations $\boldsymbol{x}$ and $\boldsymbol{y}$ of this random field is given by a

[†] Department of Computer Science, KU Leuven - University of Leuven, Celestijnenlaan 200A, B3001 Leuven, Belgium; e-mail: {`pieterjan.robbe`, `dirk.nuyens`, `stefan.vandewalle`}`@kuleuven.be` .

covariance function or kernel. A covariance function that is often used in literature for application (1) is the exponential covariance

$$\operatorname{cov}(Z(\boldsymbol{x}), Z(\boldsymbol{y})) \coloneqq \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_p}{\lambda}\right) \quad \boldsymbol{x}, \boldsymbol{y} \in D. \tag{2}$$

The parameters λ and σ^2 are the correlation length and variance of the random field $k(\boldsymbol{x};\omega)$. For subsurface flow, the parameter range of interest for the correlation length is $\lambda \ll |D|$. The variance σ^2 depends on the material.

Several techniques exist to produce samples of a random field, such as the polynomial chaos expansion [23], circulant embedding [11] or the Karhunen-Loève (KL) expansion [6]. We will focus on the latter approach. The KL expansion

$$Z(\boldsymbol{x};\omega) = \sum_{n=1}^{\infty} \sqrt{\theta_n} f_n(\boldsymbol{x}) \xi_n(\omega) \tag{3}$$

represents the random field $Z(\boldsymbol{x};\omega)$ as a linear combination of a number of eigenvalues θ_n and eigenfunctions f_n, with $\mathcal{N}(0,1)$-distributed random numbers $\xi_n(\omega)$ as coefficients. The eigenvalues θ_n and eigenfunctions f_n are the eigenvalues and eigenfunction of the covariance integral operator associated with (2). In practice, the infinite sum in equation (3) must be truncated after s terms, with $s \in \mathbb{N}$ called the stochastic dimension of the problem. The s-dimensional vector of random variables $\{\xi_n\}_{n=1}^{s}$ will be denoted as $\boldsymbol{\xi} \in \mathbb{R}^s$. Throughout this paper, we will use the 1-norm ($p = 1$) in (2). Then, analytical expressions for the eigenvalues θ_n and eigenfunctions f_n of the covariance operator are available, see [2] or [6].

Our goal is to compute certain statistical quantities, such as a mean or variance, of some functional $\mathcal{G}$ of the solution $p(\boldsymbol{x};\omega)$ to (1). This functional $G \coloneqq \mathcal{G}(p)$ is known as the *quantity of interest*. We will focus on the expected value of the quantity of interest, $\mathbb{E}[\mathcal{G}(p)]$.

The classical Monte Carlo method approximates the expected value of the quantity of interest G by a sample average. Taking samples of G typically requires three steps:

1. Generate a random vector $\boldsymbol{\xi} \in \mathbb{R}^s$ and take a sample of $k(\boldsymbol{x};\omega) = \log Z(\boldsymbol{x};\omega)$.

2. Solve the resulting deterministic PDE with a method of choice. In the numerical experiments further on, we will use a finite volume discretisation, combined with an algebraic multigrid solver.

3. Compute the quantity of interest by applying $\mathcal{G}$ to the computed solution.

Note that these three steps compute a truncated and discretised version of the quantity of interest $G = \mathcal{G}(p)$, denoted as $G_h^s \coloneqq \mathcal{G}(p_h^s)$, with h the discretisation parameter or set of such parameters.

3 Multilevel Monte Carlo Methods

Here, we review the main idea of Multilevel Monte Carlo methods [7, 13]. Consider a sequence $\{h_\ell\}_{\ell=0}^{L}$, $h_\ell = h_0 2^{-\ell}$. With every ℓ we associate a discretisation grid of PDE (1), with discretisation parameter h_ℓ. For example, in the finite volume method considered in the numerical experiments, h_ℓ corresponds to the size of the cells involved in the discretisation. The integers $\ell = 0 \ldots L$ are called *levels*. As the level number ℓ increases, the PDE is solved on an increasingly finer grid. Correspondingly, we index the approximations to the quantity of interest G on level ℓ as G_ℓ.

In a classical sampling method, the discretisation grid of the PDE is fixed, and samples are only taken on a sufficiently fine grid. In a multilevel sampling method, one avoids estimating $\mathbb{E}[G]$ directly on level L, but instead take samples on all levels $\ell = 0 \ldots L$.

We define the difference operator, denoted as Δ, by

$$\Delta G_\ell = \begin{cases} G_\ell - G_{\ell-1} & \text{if } \ell > 0 \\ G_\ell & \text{if } \ell = 0 \end{cases}.$$

Note that, since G_ℓ is a random variable, also ΔG_ℓ is a random variable. Using this definition, the expectation of G_L can be expressed as

$$\mathbb{E}[G_L] = \sum_{\ell=0}^{L} \mathbb{E}[\Delta G_\ell].$$

Let $\Delta\mathcal{Q}_\ell$ be an unbiased estimator for $\mathbb{E}[\Delta G_\ell]$, then the multilevel estimator can be expressed as

$$\mathcal{M} = \sum_{\ell=0}^{L} \Delta\mathcal{Q}_\ell \approx \mathbb{E}[G]. \tag{4}$$

If the sequence of approximations $G_0, G_1, \ldots$ convergences in mean square to G, then the variance of the differences $\mathbb{V}[\Delta G_\ell] \to 0$ as $\ell \to \infty$. Thus, fewer and fewer samples are needed on the finer grids, where samples become increasingly more expensive. Assuming the variance will be the largest at $\ell = 0$, most samples will be taken on the coarse grid, where samples are cheap. The cost reduction associated with multilevel sampling methods is due to the fact that most of the uncertainty is captured by realisations on the coarse grid, and only few realisations are needed on the fine grids.

In the Multilevel Monte Carlo (MLMC) method, a standard Monte Carlo estimator is used for each of the terms $\mathbb{E}[\Delta G_\ell]$, i.e.,

$$\Delta\mathcal{Q}_\ell = \frac{1}{N_\ell}\sum_{n=0}^{N_\ell-1} \Delta G(\xi_n), \quad \text{and, hence,} \quad \mathcal{M}_{\text{MC}} \coloneqq \sum_{\ell=0}^{L}\frac{1}{N_\ell}\sum_{n=0}^{N_\ell-1} \Delta G(\xi_n), \tag{5}$$

where we stressed the dependence of G on $\boldsymbol{\xi}$. Denote $V_\ell \coloneqq \mathbb{V}[\Delta G_\ell]$ and let W_ℓ be the amount of work to compute a single realisation of ΔG_ℓ. Then, by independence, we find that for the MLMC estimator (5)

$$\mathbb{E}[\mathcal{M}_{\text{MC}}] = \mathbb{E}[G_L] \quad \text{and} \quad \mathbb{V}[\mathcal{M}_{\text{MC}}] = \sum_{\ell=0}^{L}\frac{V_\ell}{N_\ell}.$$

It is important to note that $\Delta G(\xi_n)$ in (5) comes from using the same random numbers $\boldsymbol{\xi}$ on both levels ℓ and $\ell - 1$, in order to assure that the variance V_ℓ is low.

The objective of estimator (4) is to compute $\mathcal{M}$ to sufficient accuracy, for example with a *root-mean-square* error (RMSE) bounded by a tolerance parameter $\epsilon > 0$:

$$\sqrt{\mathbb{E}\left[(\mathcal{M} - \mathbb{E}[G])^2\right]} \leq \epsilon. \tag{6}$$

The quantity under the square root is the mean-square error (MSE), which can be expanded as

$$\text{MSE}(\mathcal{M}) = \mathbb{E}\left[(\mathcal{M} - \mathbb{E}[G])^2\right] = \mathbb{E}[(\mathcal{M} - \mathbb{E}[\mathcal{M}])^2] + (\mathbb{E}[\mathcal{M} - G])^2 = \mathbb{V}[\mathcal{M}] + \text{Bias}(\mathcal{M}, G)^2. \tag{7}$$

The first term in (7) is the variance of the estimator, often referred to as the statistical error. The second term is the square of the bias of the estimator, $\text{Bias}(\mathcal{M}, G)$. A sufficient condition to satisfy (6) is that both terms in the MSE are smaller than $\epsilon^2/2$. Alternatively, following [3], a splitting $\theta \in (0, 1)$ is often proposed between bias and statistical error:

$$\text{Bias}(\mathcal{M}, G) \leq (1 - \theta)\epsilon \qquad \text{(bias constraint)} \tag{8}$$

$$\text{prob}[|\mathcal{M} - \mathbb{E}[\mathcal{M}]| \leq \theta\epsilon] \geq 1 - \nu, \qquad \text{(statistical constraint)} \tag{9}$$

where prob stands for probability. This allows control of both accuracy ϵ and failure probability ν. Using this θ, the statistical constraint can be relaxed when the estimated bias on a certain level is smaller than $\epsilon/2$.

By the Central Limit Theorem and by the assumption that the estimator (5) has an asymptotic normal distribution, one can rewrite the statistical constraint (9) as

$$\mathbb{V}[\mathcal{M}] \leq \left(\frac{\theta\epsilon}{\Phi^{-1}(1 - \nu/2)}\right)^2 = \left(\theta\ \text{TOL}_{\epsilon,\nu}\right)^2, \quad \text{with} \quad \text{TOL}_{\epsilon,\nu} \coloneqq \frac{\epsilon}{\Phi^{-1}(1 - \nu/2)}, \tag{10}$$

where Φ is the cumulative distribution function of $\mathcal{N}(0, 1)$. See [3] for details and a proof of the asymptotic normality assumption.

Central to the success of multilevel sampling methods, are effective techniques to determine the number of samples N_ℓ, required on each level ℓ, in order to achieve a specified accuracy. For MLMC, N_ℓ can be computed by minimising the total amount of work subject to the statistical constraint (10),

$$\min_{N_\ell} \text{Total Work} = \sum_{\ell=0}^{L} N_\ell W_\ell \quad \text{s.t.} \quad \sum_{\ell=0}^{L}\frac{V_\ell}{N_\ell} \leq \left(\theta \text{TOL}_{\epsilon,\nu}\right)^2. \tag{11}$$

```
begin
  L := −1; $\theta$ := 1/2; converged := 0;
  repeat
    L ← L + 1;
    take $N^0$ samples at level L and compute sample variance and bias;
    if bias < $\epsilon/2$ then $\theta \leftarrow 1 - \text{bias}/\epsilon$; fi;
    compute optimal number of samples at each $\ell <$ L with (12);
    update samples at each $\ell <$ L;
    if L > 2 then recompute bias and check for convergence with (8); fi;
  until converged;
end
```

Algorithm 1: Algorithm for Multilevel Monte Carlo (MLMC) simulation.

By using the method of the Lagrange multipliers, one finds that for all $\ell = 0, \ldots, L$

$$N_\ell = (\theta \mathrm{TOL}_{\epsilon,\nu})^{-2} \sqrt{\frac{V_\ell}{W_\ell}} \sum_{m=0}^{L} \sqrt{V_m W_m} \eqsim \sqrt{V_\ell / W_\ell}, \tag{12}$$

where the notation $x \eqsim y$ means $x \lesssim y$ and $y \lesssim x$. The notation $x \lesssim y$ means that $x \leq cy$ with $c > 0$. In numerical simulations, one rounds (12) to $\lceil N_\ell \rceil$, the smallest integer not less than N_ℓ.

A complete MLMC algorithm is formulated as Algorithm 1. We clarify some of the remaining components. The bias can be estimated as

$$\mathrm{Bias}(\mathcal{M}, G) = \frac{|\Delta G_L|}{2^\alpha - 1}, \tag{13}$$

where α is defined in Theorem 1, see [7] for details. A certain number of samples, the so-called warm-up samples, are needed to get an initial estimate for the bias and the mean in (12) and (13). If this number of samples N^* exceeds the optimal number of samples, performance detoriation may arise, see [21]. This often happens on the fine grids, where the required number of samples is small. We find in our numerical examples that $N^* = 16$ is a good trade-off. Next, note that the splitting parameter $\theta \in (0, 1)$ will be active only in the last iteration, where the (estimated) bias is smaller then $\epsilon/2$. In all other iterations, $\theta = 0.5$. This means that the algorithm will spend less time in the last iteration. Finally, note that the algorithm can also return an error estimate,

$$\mathrm{error} = \mathrm{Bias}(\mathcal{M}, G) + \Phi^{-1}(1 - \alpha/2)\sqrt{\mathbb{V}[\mathcal{M}_{\mathrm{MC}}]}.$$

In the next two sections, we will propose a Quasi-Monte Carlo adaptation of the MLMC method. The method differs from the method in [16], in that a heuristic for the number of samples N_ℓ at each level will be used, instead of doubling the required number of samples when the statistical error is too big. This also allows for efficient parallel execution, hence further improving performance.

4 Quasi-Monte Carlo Quadrature

The Quasi-Monte Carlo (QMC) method is an equal-weight cubature rule to approximate high-dimensional integrals over the unit cube $[0, 1]^s$:

$$I_s(f) = \int_{[0,1]^s} f(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \approx \mathcal{Q}_{s,N}(f) := \frac{1}{N} \sum_{n=0}^{N-1} f(\boldsymbol{t}_n).$$

The formula is seemingly identical to the Monte Carlo estimator. However, instead of $\boldsymbol{t}_n \in [0, 1]^s$ being i.i.d. uniform random numbers, the points $\boldsymbol{t}_n$ are chosen deterministically to be better than random. Some common techniques for generating these points are *rank-1 lattice rules* [4] and *digital nets* [5]. Rather than the usual $\mathcal{O}(N^{-1/2})$ convergence behaviour for Monte Carlo methods, QMC can, under certain conditions, achieve an integration error $\mathcal{O}(N^{-\lambda})$ with $\lambda > 1/2$, see [18]. In our work, we will use the *rank-1 lattice rule* approach.

Definition 1. *An N-point* rank-1 lattice rule *in s dimensions is a QMC method with cubature points*

$$\boldsymbol{t}_n = \left\{ \frac{n\boldsymbol{z}}{N} \right\}, \quad n = 0, \ldots, N-1, \tag{14}$$

where $\boldsymbol{z} \in \mathbb{Z}^s$ *is an s-dimensional* generating vector*, and* $\{\cdot\}$ *denotes the fractional part.*

For integrands with sufficient smoothness and progressively less important dimensions, there exists lattice rules for which the integration error decays as $\mathcal{O}(N^{-1+\epsilon})$ for all $\epsilon > 0$ [4]. A generating vector $\boldsymbol{z}$ can be constructed using a *component-by-component* (CBC) algorithm with cost $\mathcal{O}(sN\log N + s^2N)$ for so-called POD-weights, see [20] for details on the CBC construction and [17] for POD weights. Our cost model for the proposed MLQMC estimator does not incorporate the cost of the CBC algorithm. This is justified by the fact that the same lattice rule can be used for any PDE in the same problem class. Hence, it can be constructed in advance.

The classical MC method comes with a probabilistic error bound of the form $\sigma(f)/\sqrt{N}$, where $\sigma^2 \coloneqq I_s(f^2) - (I_s(f))^2$ is the variance of f. This error estimate comes for free, since $\sigma(f)$ can be estimated using the same samples used to approximate $I_s(f)$. Unfortunately, QMC methods do not provide such an error bound, since the points are chosen deterministically. However, this feature can be recovered when using *random shifts*: each point in the lattice rule (14) is shifted by a vector of $\mathcal{U}(0,1)$-distributed random numbers, i.e.,

$$\boldsymbol{t}_n = \left\{\frac{n\boldsymbol{z}}{N} + \boldsymbol{\Delta}\right\}, \quad n = 0\ldots N-1.$$

We will denote the corresponding lattice rule as $\mathcal{Q}_{s,N}(f;\boldsymbol{\Delta})$. A probabilistic error estimate for the QMC method can be obtained by choosing K of these shifts $\boldsymbol{\Delta}_1,\ldots,\boldsymbol{\Delta}_K$. The approximation for the integral becomes

$$\mathcal{Q}_{s,N}^K(f) \coloneqq \frac{1}{K}\sum_{k=1}^K \mathcal{Q}_{s,N}(f;\boldsymbol{\Delta}_k).$$

Since $\mathcal{Q}_{s,N}(f;\boldsymbol{\Delta}_1),\ldots,\mathcal{Q}_{s,N}(f;\boldsymbol{\Delta}_K)$ are i.i.d. random variables, the variance $\mathbb{V}_{\boldsymbol{\Delta}}$ over these K estimators can be used to construct a confidence interval for $\mathcal{Q}_{s,N}^K(f)$, see [15].

A tool for analysing convergence of QMC methods is the so-called *worst-case error*, defined as

$$e_{\text{wor}}(\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n) \coloneqq \sup_{\substack{f\in\mathscr{H}\\ \|f\|_{\mathscr{H}}\le 1}} |I_s(f) - \mathcal{Q}_{s,N}| \tag{15}$$

for some Banach space of functions $\mathscr{H}$. In the case of randomly-shifted rank-1 lattice rules, $\mathscr{H}$ is the *weighted* and *unanchored* Sobolev space of functions on $[0,1]^s$, with square integrable mixed first-order derivatives. Its norm is given by

$$\|f\|_{s,\boldsymbol{\gamma}}^2 \coloneqq \sum_{\mathfrak{u}\subseteq\{1:s\}} \frac{1}{\gamma_{\mathfrak{u}}} \int_{[0,1]^{|\mathfrak{u}|}} \left(\int_{[0,1]^{s-|\mathfrak{u}|}} \frac{\partial^{|\mathfrak{u}|}f}{\partial\boldsymbol{y}_{\mathfrak{u}}}(\boldsymbol{y}_{\mathfrak{u}};\boldsymbol{y}_{\{1:s\}\setminus\mathfrak{u}})\mathrm{d}\boldsymbol{y}_{\mathfrak{u}}\right)^2 \mathrm{d}\boldsymbol{y}_{\mathfrak{u}},$$

with $\{1:s\}$ as shorthand notation for $\{1,2\ldots s\}$ and $\partial^{|\mathfrak{u}|}f/\partial\boldsymbol{y}_{\mathfrak{u}}$ the mixed first-order derivative with respect to the *active* variables $\boldsymbol{y}_{\mathfrak{u}}$ and the *inactive* variables $\boldsymbol{y}_{\{1:s\}\setminus\mathfrak{u}}$ integrated out. For details, we refer to [15].

Using (15), we have the QMC error bound

$$|I_s(f) - \mathcal{Q}_{s,N}(f)| \le e_{\text{wor}}(\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n)\|f\|_{\mathscr{H}} \quad \text{for all } f\in\mathscr{H}.$$

There is also a probabilistic counterpart of this error bound, using random shifts:

$$\sqrt{\mathbb{E}[|I_s(f) - \mathcal{Q}_{s,N}^K(f)|^2]} \le \frac{1}{\sqrt{K}} e_{\text{wor}}^{\text{sh}}(\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n)\|f\|_{\mathscr{H}} \quad \text{for all } f\in\mathscr{H}, \tag{16}$$

with the *shift-averaged worst-case error* defined as

$$e_{\text{wor}}^{\text{sh}}(\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n) \coloneqq \sqrt{\int_{[0,1]^s} e_{\text{wor}}^2(\boldsymbol{t}_1+\boldsymbol{\Delta},\ldots,\boldsymbol{t}_n+\boldsymbol{\Delta})\mathrm{d}\boldsymbol{\Delta}}.$$

To apply the QMC method to compute the expectation of a quantity of interest, one must reformulate that expectation as an integral over the unit cube $[0,1]^s$. This can be done by means of a change of variables, see [10]. Unfortunately, after this mapping, the resulting function no longer belongs to the proposed function space $\mathscr{H}$. Strictly speaking, a different function space setting must be considered, see [15], for which specific generating vectors $\boldsymbol{z}$ can be constructed. However, motivated by [19], we will assume that a similar bound as (16) is true for functions defined on $\mathbb{R}^s$ and pick a standard generating vector from [14].

```
begin
  L := $-1$; $\theta := 1/2$; converged := 0;
  repeat
    L ← L + 1;
    generate $K$ random shifts $\boldsymbol{\Delta}_{k,\mathsf{L}}$ $k = 1 \dots K$ and take $N^*$ samples at level L;
    compute sample variance $\mathbb{V}_{\boldsymbol{\Delta}}[\Delta G_{\mathsf{L}}]$ and bias;
    if bias < $\epsilon/2$ then $\theta \leftarrow 1 - \mathsf{bias}/\epsilon$; fi;
    compute optimal number of samples at each $\ell <$ L with (20);
    update samples at each $\ell <$ L using the random shifts $\boldsymbol{\Delta}_{k,\ell}$;
    if L > 2 then recompute bias and check for convergence with (8); fi;
  until converged;
end
```

Algorithm 2: Algorithm for Multilevel Quasi-Monte Carlo (MLQMC) simulation.

5 Multilevel Quasi-Monte Carlo Quadrature

In this section, we construct the Multilevel Quasi-Monte Carlo (MLQMC) estimator by combining the Multilevel Monte Carlo method with the randomly-shifted rank-1 lattice rules introduced in the previous section. To obtain an expression for the optimal number of samples at each level, we must first find an expression for the variance of the new estimator. Next, we can formulate the same optimisation problem as (11) and solve for N_ℓ. Finally, we introduce an algorithm for MLQMC simulation and derive a theoretical cost estimate.

First, we extend (4) to a Multilevel Quasi-Monte Carlo estimator, i.e.,

$$\mathcal{M}_{\mathrm{QMC}} \coloneqq \sum_{\ell=0}^{L} \mathcal{Q}^K_{s,N_\ell}(\Delta G_\ell) = \sum_{\ell=0}^{L} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_\ell} \sum_{n=0}^{N_\ell - 1} \Delta G_\ell \left(\boldsymbol{t}_n + \boldsymbol{\Delta}_{k,\ell} \right). \tag{17}$$

This involves a total of $K(L+1)$ different shifts $\boldsymbol{\Delta}_{k,\ell}$. Note that $\mathcal{Q}^K_{s,N_\ell}(\Delta G_\ell)$ is an unbiased estimator for ΔG_ℓ. By analysing the error of the MLQMC estimator $\mathcal{M}_{\mathrm{QMC}}$, we will obtain a heuristic for the number of lattice points N_ℓ at each level. The variance of the estimator can be elaborated as follows,

$$\mathbb{V}_{\boldsymbol{\Delta}}[\mathcal{M}_{\mathrm{QMC}}] = \mathbb{V}_{\boldsymbol{\Delta}} \left[\sum_{\ell=0}^{L} \mathcal{Q}^K_{s,N_\ell}(\Delta G_\ell) \right] = \sum_{\ell=0}^{L} \mathbb{V}_{\boldsymbol{\Delta}} \left[\mathcal{Q}^K_{s,N_\ell}(\Delta G_\ell) \right], \tag{18}$$

by independence and unbiasedness of the shifts $\boldsymbol{\Delta}_{k,\ell}$. Using the definition of variance and assuming a bound like (16) also holds here under the correct function space setting, we can bound (18) as

$$\begin{aligned} \mathbb{V}_{\boldsymbol{\Delta}}[\mathcal{M}_{\mathrm{QMC}}] &= \sum_{\ell=0}^{L} \mathbb{E}_{\boldsymbol{\Delta}} \left[|\mathcal{Q}^K_{s,N_\ell}(\Delta G_\ell) - \mathbb{E}_{\boldsymbol{\Delta}}[\Delta G_\ell]|^2 \right] \\ &\leq \sum_{\ell=0}^{L} \frac{1}{K} e^{\mathrm{sh}}_{\mathrm{wor}}(\boldsymbol{t}_1, \dots, \boldsymbol{t}_n)^2 \|\Delta G_\ell\|^2_{\mathscr{H}} \leq \sum_{\ell=0}^{L} \frac{1}{K} \left(\frac{\mathcal{C}_{s,\lambda}}{N_\ell^\lambda} \right)^2 \|\Delta G_\ell\|^2_{\mathscr{H}}. \end{aligned}$$

We assumed the use of a QMC method that converges as $\mathcal{O}(1/N^\lambda)$ with $\mathcal{C}_{s,\lambda}$ a constant that depends only on the dimension of the integral and the point set. Given this bound, we can now formulate the optimisation problem

$$\min_{N_\ell} \text{ Total Work} = K \sum_{\ell=0}^{L} N_\ell W_\ell \quad \text{s.t.} \quad \sum_{\ell=0}^{L} \frac{1}{K} \left(\frac{\mathcal{C}_{s,\lambda}}{N_\ell^\lambda} \right)^2 \|\Delta G_\ell\|^2_{\mathscr{H}} \leq \left(\theta \mathrm{TOL}_{\epsilon,\nu} \right)^2. \tag{19}$$

Using the method of the Lagrange multipliers, (19) can be written as a problem without constraints. The solution of the original optimisation problem is a critical point of the Lagrangian

$$\mathcal{L}(N_\ell, \mu) = K \sum_{\ell=0}^{L} N_\ell W_\ell + \mu \left(\sum_{\ell=0}^{L} \frac{1}{K} \left(\frac{\mathcal{C}_{s,\lambda}}{N_\ell^\lambda} \right)^2 \|\Delta G_\ell\|^2_{\mathscr{H}} - (\theta \mathrm{TOL}_{\epsilon,\nu})^2 \right).$$

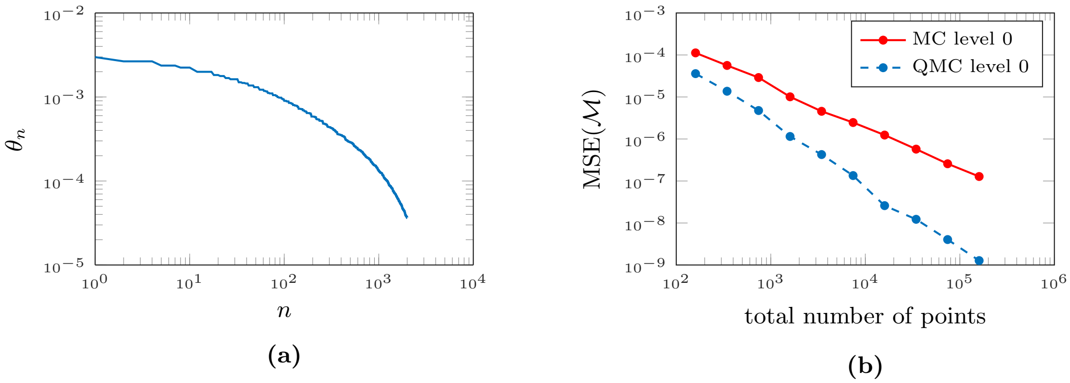


Figure 1: **(a)** Decay of the eigenvalues θ_n in the KL expansion (3). In our numerical tests, we will use $s = 2000$ terms. **(b)** Estimation of the convergence parameter λ of the QMC lattice rule. We find $\lambda \approx 0.85$.

First-order optimality conditions stipulate that

$$\frac{\partial \mathcal{L}}{\partial N_\ell} = K W_\ell + \mu \frac{1}{K}(-2\lambda)\frac{\mathcal{C}_{s,\lambda}^2}{N_\ell^{2\lambda+1}}\|\Delta G_\ell\|_{\mathscr{H}}^2 = 0,$$

and hence,

$$N_\ell \eqsim \sqrt[2\lambda+1]{\frac{2\lambda \mathcal{C}_{s,\lambda}^2 \|\Delta G_\ell\|_{\mathscr{H}}^2}{K^2 W_\ell}}.$$

The above derivation extends the one for the classical Monte Carlo method, which corresponds to $\lambda = 1/2$, $N_\ell = 1$ and K the number of i.i.d. samples. From the analysis of the MLMC estimator in §3, one has that in that case $K \eqsim \sqrt{V_\ell / W_\ell}$, see (12). Hence, the extended analysis is valid for both MLMC and MLQMC if we propose $\mathcal{C}_{s,\lambda}^2 \|\Delta G_\ell\|_{\mathscr{H}}^2 \lesssim V_\ell$. This leads to the following heuristic estimate for the optimal number of samples at each level:

$$N_\ell \geq \sqrt[2\lambda]{(\theta \mathrm{TOL}_{\epsilon,\nu})^{-1} \frac{1}{K}\left(\frac{V_\ell}{W_\ell}\right)^{\frac{2\lambda}{2\lambda+1}} \sum_{m=0}^{L} \sqrt[2\lambda+1]{V_\ell W_\ell^{2\lambda}}}. \tag{20}$$

Note that this value is in agreement with equation (12) for $\lambda = 1/2$.

A theoretical cost estimate for a MLQMC algorithm has been proposed and proven in [16]. The results can be adapted in a straightforward way to cover our algorithm. The only (minor) difference in the following theorem is that we neglect the influence of s on the cost (i.e., we keep s fixed).

Theorem 1. *Suppose there are nonnegative constants α, β and γ such that*

(i) $|\mathbb{E}[Q_L - Q]| \lesssim h_L^\alpha$

(ii) $\mathbb{E}_{\boldsymbol{\Delta}}[\mathcal{Q}_{s,N_\ell}^K(\Delta G_\ell)] = \mathbb{E}[\Delta G_\ell]$

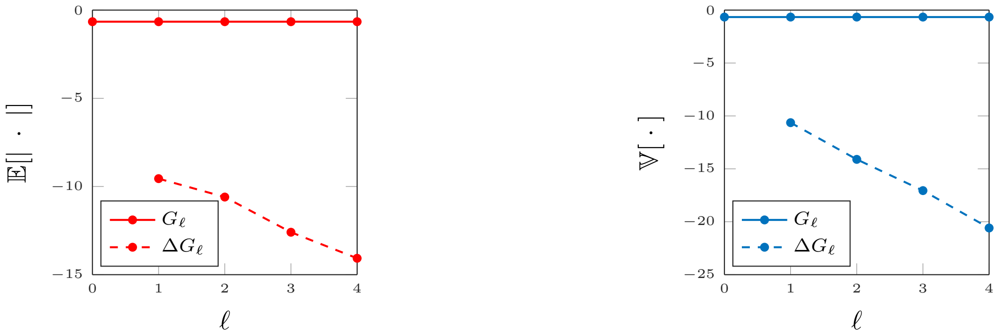
(iii) $\mathbb{V}_{\boldsymbol{\Delta}}[\mathcal{Q}_{s,N_\ell}^K(\Delta G_\ell)] \lesssim K^{-1} N_\ell^{-2\lambda} h_{\ell-1}^\beta$

(iv) $\mathrm{cost}(\mathcal{Q}_{s,N_\ell}^K(\Delta G_\ell)) \lesssim K N_\ell h_\ell^{-\gamma}$

Then, for any $\epsilon > 0$, there exists a choice of L and $N_0, \ldots, N_L$ such that

$$\mathrm{MSE}(\mathcal{M}) \lesssim \epsilon^2 \textit{ and } \mathrm{cost}(\mathcal{M}) \lesssim \begin{cases} \epsilon^{-1/\lambda} & \textit{when } \beta > 2d\lambda \\ \epsilon^{-1/\lambda}(\log \epsilon{-}1)^{1/(2\lambda)+1} & \textit{when } \beta = 2d\lambda \\ \epsilon^{-1/\lambda-(d-\beta/(2\lambda))/\alpha} & \textit{when } \beta < 2d\lambda \end{cases}. \tag{21}$$

All components are now in place to formulate an algorithm for MLQMC simulation, see Algorithm 2. The same remarks given for Algorithm 1 also apply here.



(a)

(b)

Figure 2: (a) and **(b)**: Estimation of the parameters $\alpha \approx 1.55$ and $\beta \approx 3.28$ in Theorem 1. Using $\gamma \approx 1$, we derive that the asymptotic cost bound is $\mathcal{O}(\epsilon^{-2})$ for MLMC and $\mathcal{O}(\epsilon^{-1.36})$ for MLQMC.

6 Numerical Results

We numerically demonstrate the performance of the MLMC and MLQMC estimators and illustrate their superiority over standard MC estimation.

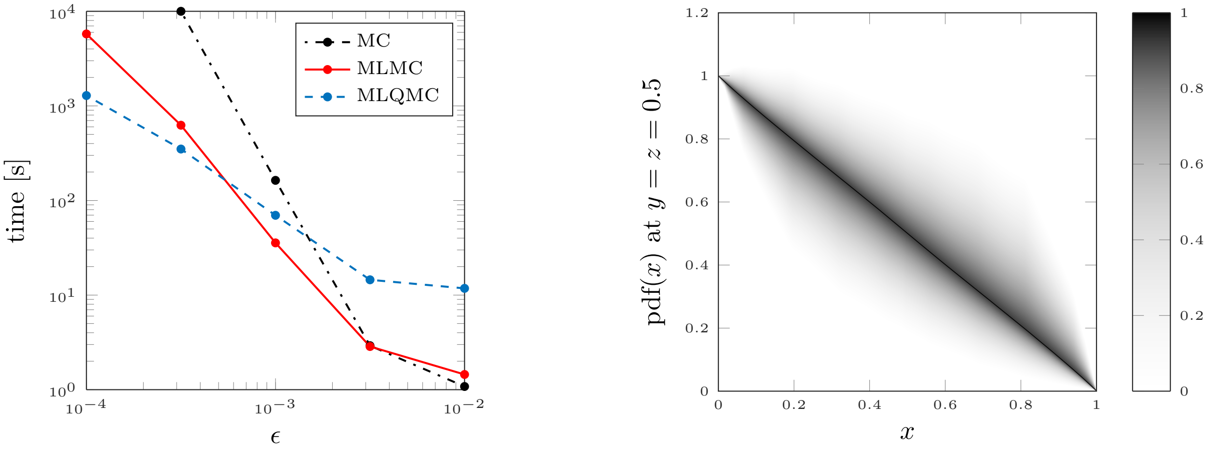
Consider the 3-dimensional PDE (1) with a diffusion coefficient characterised by (2) with correlation length $\lambda = 0.075$, variance $\sigma^2 = 1$ and coarsest mesh size $h_0 = 1/4$. This is a particularly challenging problem because of the slow decay of the eigenvalues θ_n in expansion (3), see Figure 1(a). A lot of terms must be kept to assure a sufficiently small truncation error. Here, we will use $s = 2000$, i.e., the random space is of dimension 2000. The spatial geometry is a simple flow cell $D = [0,1]^3$ with Dirichlet boundary conditions $p(0) = 1$ and $p(1) = 0$. There is no outflow through the other boundaries, i.e., we assume homogeneous Neumann conditions, and set the source term $f = 0$. As a quantity of interest, we will consider a point evaluation of the pressure at $\boldsymbol{x}^* = [0.214, 0.369, 0.857]$. Further parameters are $N^* = 16$ warm-up samples and a failure probability $\alpha = 10\%$. The number of shifts is chosen as $K = 16$. In our application, this appears to be a good trade-off between an accurate variance estimation without affecting the convergence rate of the method. We pick a generating vector $\boldsymbol{z}$ from [14]. All simulations are performed on a shared memory computer with 20 2.8GHz processors and 64GB RAM.

To predict the asymptotic bound on the cost of both MLMC and MLQMC in Theorem 1, we need values for the parameters γ, λ, α and β. The discretised PDE problem is solved using a conjugate gradient method with the algebraic multigrid preconditioner from the HSL Mathematical Software Library, see [1]. This choice leads to a numerically determined parameter value of $\gamma \approx 1.0753$. The parameter λ, also needed in the MLQMC algorithm, is estimated as $\lambda \approx 0.85$, see Figure 1(b). Figure 2(a) shows the behaviour of the expected value of the quantity of interest G_ℓ and the difference ΔG_ℓ for different levels. The slope of the line is approximately -1.55, hence $\mathbb{E}[\Delta G_\ell] \approx (1/h_\ell)^{-1.55}$, or $\alpha \approx 1.55$. Similarly, Figure 2(b) shows the behaviour of the variance of G_ℓ and ΔG_ℓ. We find $\beta \approx 3.28$. Since $\gamma \approx 1$, we expect from Theorem 1 a cost $\mathcal{O}(\epsilon^{-2})$ for MLMC and $\mathcal{O}(\epsilon^{-1.36})$ for MLQMC.

To check these bounds on the cost, we plot a cost versus accuracy graph in Figure 3(a). Note the logarithmic scale on the time-axis: for $\epsilon =$ `1e-4` we reduced the simulation time from 1h 36min for MLMC to only 21min for MLQMC. We compare the experimentally achieved cost bounds with the derived bounds from Theorem 1:

method	theoretical cost bound	experimental cost bound
MLMC	$\epsilon^{-2.00}$	$\epsilon^{-2.23}$
MLQMC	$\epsilon^{-1.36}$	$\epsilon^{-1.31}$

Notice that there is a good agreement between the experimental and theoretical error bounds, both for MLMC and MLQMC. For reference we include the results of a classical Monte Carlo method for the first few tolerances in Figure 3(a). Note that the observed cost $\epsilon^{-3.53}$ is far from the expected ϵ^{-2}, since the MC simulations are performed at different finest levels, to get a sufficient decrease of the bias.



(a)

(b)

Figure 3: **(a)**: Cost versus accuracy graph for MLMC and MLQMC. **(b)**: Normalized probability density function (pdf) along the x-axis, for $y = z = 0.5$.

The MLMC and MLQMC estimators still have all the benefits of the classical Monte Carlo method: samples can be taken in parallel, one can obtain a *probability density function* (pdf) of the quantity of interest and the expected value of multiple quantities of interest can be obtained from a single simulation. For the latter, the quantity of interest with the highest variance V_ℓ will determine the number of samples to be taken at level ℓ, see [8]. Figure 3(b), for example, shows the (normalized) pdf of a point evaluation of the pressure along the x-axis, for $y = z = 0.5$.

7 Conclusions and Further Work

We proposed a Multilevel Quasi-Monte Carlo (MLQMC) algorithm for simulating partial differential equations with random coefficients. We focused on problems with lognormal diffusion coefficient arising in subsurface flow problems. The MLQMC method combines ideas from the classical Multilevel Monte Carlo (MLMC) method with rank-1 lattice rules.

This gives a new improved complexity result that achieves a cost almost proportional to the requested tolerance on the expected value of some quantity of interest. As opposed to the work in [16] and [9], we derived an explicit expression for the optimal number of samples at each of the levels. All of these samples can be taken in parallel, further boosting performance. Moreover, our algorithm uses the advanced error splitting presented in [3], allowing control of both error and failure probability.

Numerical results illustrate the superiority of the new MLQMC algorithm over classical Monte Carlo and Multilevel Monte Carlo simulation. In our experiments, we obtained a decrease in simulation time w.r.t. MLMC by a factor 4.5 for the smallest achievable tolerance. With respect to classical Monte Carlo, the improvement goes up to several orders of magnitude.

Future research will focus on combining the MLQMC estimator with other variance reduction techniques, such as higher-order digital nets [5], and the recently proposed Multi-Index Monte Carlo method [12].

References

[1] J. Boyle, M. Mihajlović, and J. Scott. HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D. *International journal for numerical methods in engineering*, 82(1):64–98, 2010.

[2] K. Cliffe, M. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

[3] N. Collier, A.-L. Haji-Ali, F. Nobile, E. von Schwerin, and R. Tempone. A continuation Multilevel Monte Carlo algorithm. *BIT Numerical Mathematics*, pages 1–34, 2014.

[4] J. Dick, F. Y. Kuo, and I. H. Sloan. High dimensional integration: the Quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

[5] J. Dick and F. Pillichshammer. *Digital nets and sequences: Discrepancy Theory and Quasi-Monte Carlo Integration.* Cambridge University Press, Cambridge, 2010.

[6] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: a Spectral Approach.* Courier Corporation, Chelmsford, 2003.

[7] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[8] M. B. Giles. Multilevel Monte Carlo methods. In J. Dick, F. Y. Kuo, G. W. Peters, and I. H. Sloan, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 83–103. Springer, New York, 2013.

[9] M. B. Giles and B. J. Waterhouse. Multilevel Quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, 8(1):165–181, 2009.

[10] I. Graham, F. Y. Kuo, J. Nichols, R. Scheichl, C. Schwab, and I. H. Sloan. Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients. *Numerische Mathematik*, pages 1–40, 2015.

[11] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *Journal of Computational Physics*, 230(10):3668–3694, 2011.

[12] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-Index Monte Carlo: When sparsity meets sampling. *arXiv preprint arXiv:1405.3757*, 2014.

[13] S. Heinrich. Multilevel Monte Carlo methods. In I. Lirkov, S. D. Margenov, and J. Wasniewski, editors, *Large-scale scientific computing*, pages 58–67. Springer, Berlin, Heidelberg, 2001.

[14] F. Y. Kuo. Lattice rule generating vectors. `http://web.maths.unsw.edu.au/~fkuo/lattice/index.html`.

[15] F. Y. Kuo and D. Nuyens. Application of Quasi-Monte Carlo methods to PDEs with random coefficients - analysis and implementation, (submitted).

[16] F. Y. Kuo, R. Scheichl, C. Schwab, I. H. Sloan, and E. Ullmann. Multilevel Quasi-Monte Carlo methods for lognormal diffusion problems. *arXiv preprint arXiv:1507.01090*, 2015.

[17] F. Y. Kuo, C. Schwab, and I. H. Sloan. Multi-level Quasi-Monte Carlo Finite Element Methods for a Class of Elliptic PDEs with Random Coefficients. *Foundations of Computational Mathematics*, 15(2):1–39, April 2015.

[18] F. Y. Kuo and I. H. Sloan. Lifting the curse of dimensionality. *Notices of the AMS*, 52(11):1320–1328, 2005.

[19] J. A. Nichols and F. Y. Kuo. Fast CBC construction of randomly shifted lattice rules achieving O(n-1+δ) convergence for unbounded integrands over Rs in weighted spaces with POD weights. *Journal of Complexity*, 30(4):444–468, 2014.

[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. *Mathematics of Computation*, 75(254):903–920, 2006.

[21] S. Pauli, R. N. Gantner, P. Arbenz, and A. Adelmann. Multilevel Monte Carlo for the Feynman–Kac formula for the Laplace equation. *BIT Numerical Mathematics*, pages 1–19, 2014.

[22] A. Teckentrup, R. Scheichl, M. Giles, and E. Ullmann. Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numerische Mathematik*, 125(3):569–600, 2013.

[23] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.