

RECYCLING SAMPLES IN THE MULTIGRID MULTILEVEL (QUASI-)MONTE CARLO METHOD*

PIETERJAN ROBBE[†], DIRK NUYENS[†], AND STEFAN VANDEWALLE[†]

Abstract. The multilevel Monte Carlo method is an efficient variance reduction technique. It uses a sequence of coarse approximations to reduce the computational cost in uncertainty quantification applications. The method is nowadays often considered to be the method of choice for solving PDEs with random coefficients when many uncertainties are involved. When using full multigrid to solve the deterministic problem, coarse solutions obtained by the solver can be recycled as samples in the multilevel Monte Carlo method, as was pointed out by Kumar, Oosterlee and Dwight [*Int. J. Uncertain. Quantif.*, 7 (2017), pp. 57–81]. In this article, an alternative approach is considered, using quasi-Monte Carlo points, to speed up convergence. Additionally, our method comes with an improved variance estimate which is also valid in case of the Monte Carlo based approach. The new method is illustrated on the example of an elliptic PDE with lognormal diffusion coefficient. Numerical results for a variety of random fields with different smoothness parameters in the Matérn covariance function show that sample recycling is more efficient when the input random field is nonsmooth.

Key words. multilevel Monte Carlo, multigrid, PDEs with random coefficients

AMS subject classifications. 60H15, 65C05, 68U20

DOI. 10.1137/18M1194031

1. Introduction and background. Mathematical models in science or engineering that are subject to uncertainty often involve spatially varying input data represented as a random field. The uncertainty characterized by the input random field leads to a corresponding uncertainty in the output of the model and in any quantity of interest derived thereof. The computational goal is then to find statistics of this quantity of interest, such as an expected value or a higher-order moment.

Throughout this article, we consider the model parametric elliptic partial differential equation (PDE)

$$(1.1) \quad -\nabla \cdot a(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}), \quad \mathbf{x} \in D,$$

where D is a bounded domain in \mathbb{R}^d , $d = 1, 2, 3$, with boundary ∂D . We assume deterministic boundary conditions are given. The diffusion coefficient $a(\mathbf{x}, \mathbf{y})$ is a random field (i.e., a random variable for which every realization is a function defined on D), and $\mathbf{y} = (y_j)_{j \geq 1}$ is an infinite sequence which characterizes an instance of the random field. In practice, this infinite number of random variables needs to be truncated.

Depending on the characteristics of the random field and the way it is represented, many uncertain parameters can be needed to accurately model the random field.

*Received by the editors June 14, 2018; accepted for publication (in revised form) January 9, 2019; published electronically October 29, 2019.

<https://doi.org/10.1137/18M1194031>

Funding: This work was supported by project IWT/SBO EUFORIA: “Efficient Uncertainty Quantification For Optimization in Robust design of Industrial Applications” (IWT-140068) of the Agency for Innovation by Science and Technology, Flanders, Belgium, and by the Statistical and Applied Mathematical Sciences Institute (SAMSI) 2017 Year-long Program on Quasi-Monte Carlo and High-Dimensional Sampling Methods for Applied Mathematics.

[†]KU Leuven, Department of Computer Science, NUMA Section, Celestijnenlaan 200A box 2402, 3001 Leuven, Belgium (pieterjan.robbe@kuleuven.be, dirk.nuyens@kuleuven.be, stefan.vandewalle@kuleuven.be).

Sampling-based methods, such as the Monte Carlo method, are the preferred tool to deal with these many uncertainties. They require the solution of problem (1.1) for every draw of a sufficiently large sample set $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(N)}$, where $\mathbf{y}^{(i)}$ is a finite-dimensional vector and N is the number of samples. From this sample set, conclusions are made about the statistics of a quantity of interest $F(u)$, such as the expected value, variance, or higher-order moments. Here, F is a functional that is applied to the solution u , for example, a point evaluation of u or the average value over a subset of the domain D . Note that uncertainty in the diffusion coefficient $a(\mathbf{x}, \mathbf{y})$ leads to uncertainty in the solution $u(\mathbf{x}, \mathbf{y})$ and hence in the quantity of interest $F(u)$. In the remainder of the text, we make this relation explicit by using the shorthand notation $F(\mathbf{y}) := F(u(\mathbf{x}, \mathbf{y}))$.

In 2008, the multilevel Monte Carlo (MLMC) method was introduced as an alternative for standard Monte Carlo sampling; see [12]. Instead of sampling the quantity of interest directly on the desired model resolution, sampling is done on the differences of a coarse-to-fine hierarchy of model approximations. Most samples will be taken on the coarse approximations, where samples are cheap and only few samples are required on the finest approximations. This explains the huge benefit that is commonly attributed to MLMC methods. For details, see, e.g., [2, 4, 13] and references therein.

Since then, many extensions of the multilevel method have been presented by various authors. A notable example is the multilevel quasi-Monte Carlo (MLQMC) method (see [14, 25, 32]), where the random sampling from the Monte Carlo method is replaced by deterministic sampling to speed up convergence. Another extension is the multi-index Monte Carlo (MIMC) method (see [19]), where the multilevel hierarchy is replaced by a multidimensional hierarchy of model approximations. See also [34] for its extension to QMC.

In [22], a combination of multigrid and multilevel Monte Carlo was presented. In this method, a full multigrid (FMG) approach is used to solve the deterministic PDE in every sample from (1.1). For every solution at a fine approximation, the method provides samples on all preceding coarser approximations, which can be reused in the estimator. In this work, we present an alternative approach, based on the aforementioned MLQMC method, which also allows the recycling of coarse samples. Additionally, our method comes with a mathematically sound estimate for the variance of the obtained estimator, which is also valid in the Monte Carlo based approach.

This article is organized as follows. In section 2, we discuss the multilevel sampling framework in more detail and introduce the multigrid multilevel Monte Carlo (MG-MLMC) method. We review the practical implementation, at which point we introduce the quasi-Monte Carlo variant (MG-MLQMC), and analyze the cost of the estimator in section 3. After that, we turn our focus to the PDE example outlined above. We discuss FMG in section 4, and random field generation using the Karhunen–Loève (KL) expansion in section 5. The article is concluded with various numerical experiments in section 6, which confirm the theory.

2. Multigrid Multilevel Monte Carlo. The classic sampling-based method is the Monte Carlo method. A Monte Carlo estimator for the expected value of the quantity of interest $\mathbb{E}[F]$ is simply an average over N realizations of the uncertain parameters, i.e.,

$$\mathcal{Q}^{\text{MC}} := \frac{1}{N} \sum_{n=1}^N F(\mathbf{y}^{(n)}),$$

where $\mathbf{y}^{(n)}$ denotes the n th independent realization. The Monte Carlo method has the well-known convergence of the root mean square error (RMSE) of order $O(1/\sqrt{N})$ for functions with a finite variance. This makes the method rather slow in real engineering applications, where many expensive samples are required to reach a certain accuracy. However, the use of multilevel sampling methods to reduce the cost of the estimator makes the Monte Carlo approach feasible in engineering practice. In the remainder of this section, we briefly discuss this multilevel extension, before introducing the MG-MLMC method.

2.1. Multilevel Monte Carlo. Monte Carlo sampling can be combined with a multilevel idea to yield an efficient sampling method. Assume there is a hierarchy of approximations, conveniently called *levels*, F_0, F_1, \dots, F_L that converge to the true quantity of interest $F(u)$ for $L \rightarrow \infty$.

An example of such a hierarchy is the solution of a PDE on a mesh with decreasing grid size. Instead of estimating the quantity of interest directly on the finest and most expensive grid, the multilevel method estimates the quantity of interest on the coarsest and cheapest grid, F_0 , together with a series of correction terms (or *differences*) $F_\ell - F_{\ell-1}$, $\ell = 1, \dots, L$. These differences constitute the telescoping sum

$$(2.1) \quad \mathbb{E}[F_L] = \sum_{\ell=0}^L \mathbb{E}[F_\ell - F_{\ell-1}]$$

that lies at the basis of any multilevel method and where we set $F_{-1} = 0$. The expectation of the differences in (2.1) can be computed using the standard Monte Carlo method, resulting in the MLMC method

$$\mathcal{Q}_L^{\text{MC}} := \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} \left(F_\ell(\mathbf{y}_\ell^{(n)}) - F_{\ell-1}(\mathbf{y}_\ell^{(n)}) \right),$$

where $\mathbf{y}_\ell^{(n)}$ denotes the n th realization of the random parameters at level ℓ . The subscript L in $\mathcal{Q}_L^{\text{MC}}$ denotes the finest level used in the estimator. The unknown number of samples on each level, $N_\ell \in \mathbb{N}$, and the total number of levels L , remain to be determined.

Notice that the same random numbers $\mathbf{y}_\ell^{(n)}$ are used to compute a sample of the difference $F_\ell - F_{\ell-1}$. The hope is that F_ℓ and $F_{\ell-1}$ are strongly positively correlated, and the variance of the difference $V_\ell := \mathbb{V}[F_\ell - F_{\ell-1}]$ is reduced; since then

$$V_\ell = \mathbb{V}[F_\ell] + \mathbb{V}[F_{\ell-1}] - 2\text{cov}(F_\ell, F_{\ell-1}) \ll \mathbb{V}[F_\ell] + \mathbb{V}[F_{\ell-1}].$$

Furthermore, as the level ℓ increases, the expectation of the difference $F_\ell - F_{\ell-1}$ decreases, and the variance reduction will be more pronounced. Hence, the total number of samples N_ℓ decreases to zero as ℓ increases, and fewer samples will be required on finer levels.

In a typical MLMC simulation, the error is controlled by bounding the mean square error (MSE)

$$(2.2) \quad \text{MSE}(\mathcal{Q}_L^{\text{MC}}) = \mathbb{E}[(\mathcal{Q}_L^{\text{MC}} - \mathbb{E}[F])^2] = \mathbb{V}[\mathcal{Q}_L^{\text{MC}}] + \text{Bias}(\mathcal{Q}_L^{\text{MC}})^2$$

by a fixed tolerance ϵ^2 . This condition is fulfilled if both terms in (2.2) are smaller than $\epsilon^2/2$.

Since the variance of the estimator is given by the sum of the variances over the levels, and by using the classic argument of Giles from [12], the cost of the estimator is minimized for a fixed variance satisfying

$$(2.3) \quad \mathbb{V}[\mathcal{Q}_L^{\text{MC}}] = \sum_{\ell=0}^L N_\ell^{-1} V_\ell \leq \epsilon^2/2$$

if the number of samples is chosen as

$$(2.4) \quad N_\ell = \left\lceil \frac{2}{\epsilon^2} \sqrt{\frac{V_\ell}{C_\ell}} \left(\sum_{\tau=0}^L \sqrt{V_\tau C_\tau} \right) \right\rceil,$$

where C_ℓ represents the cost to compute a single sample on level ℓ .

2.2. Formulation and main idea. Besides efficient sampling methods, such as the MLMC method outlined above, one also requires good deterministic solvers to compute the solution $u(\mathbf{x}, \mathbf{y})$ for every sample of the random field $a(\mathbf{x}, \mathbf{y})$ in (1.1). One particular choice are multigrid methods. An FMG method can compute a solution to discretization accuracy in $\mathcal{O}(M)$ operations, where M is the number of degrees of freedom. FMG uses a hierarchy of coarse approximations, much similar to the MLMC hierarchy, to accelerate convergence on the fine grid. Further details of the FMG method will be deferred to section 4.

The idea of the MG-MLMC method is essentially to use the same hierarchy of coarse approximations from the FMG solver as levels in the MLMC method. Every deterministic solution on level ℓ using FMG will additionally give free coarse solutions on all levels $0 \leq k < \ell$, which could be recycled in the MLMC estimator; however, these coarse solutions are not uncorrelated, as required by (2.3).

The MG-MLMC estimator, as it was first introduced in [22], can be written as

$$(2.5) \quad \mathcal{Q}_{L,\text{reuse}}^{\text{MC}} := \sum_{\ell=0}^L \left(\frac{1}{\sum_{i=\ell}^L N_i} \right) \sum_{k=\ell}^L \sum_{n=1}^{N_k} \left(F_\ell(\mathbf{y}_k^{(n)}) - F_{\ell-1}(\mathbf{y}_k^{(n)}) \right).$$

Here, the additional sum over k represents the sample recycling, and we assume $N_0 \geq N_1 \geq \dots \geq N_L$. Fully expanded for $L = 2$, the MG-MLMC estimator reads

$$\begin{aligned} \mathcal{Q}_{2,\text{reuse}}^{\text{MC}} &= \frac{1}{N_0 + N_1 + N_2} \left(\sum_{n=1}^{N_0} F_0(\mathbf{y}_0^{(n)}) + \sum_{n=1}^{N_1} F_0(\mathbf{y}_1^{(n)}) + \sum_{n=1}^{N_2} F_0(\mathbf{y}_2^{(n)}) \right) \\ &\quad + \frac{1}{N_1 + N_2} \left(\sum_{n=1}^{N_1} \left(F_1(\mathbf{y}_1^{(n)}) - F_0(\mathbf{y}_1^{(n)}) \right) + \sum_{n=1}^{N_2} \left(F_1(\mathbf{y}_2^{(n)}) - F_0(\mathbf{y}_2^{(n)}) \right) \right) \\ &\quad + \frac{1}{N_2} \left(\sum_{n=1}^{N_2} \left(F_2(\mathbf{y}_2^{(n)}) - F_1(\mathbf{y}_2^{(n)}) \right) \right). \end{aligned}$$

Equation (2.5) is a sum of $L + 1$ terms, where each term is given by

$$(2.6) \quad \Upsilon_\ell := \left(\frac{1}{\sum_{i=\ell}^L N_i} \right) \sum_{k=\ell}^L \sum_{n=1}^{N_k} \left(F_\ell(\mathbf{y}_k^{(n)}) - F_{\ell-1}(\mathbf{y}_k^{(n)}) \right),$$

which are estimators for the difference $F_\ell - F_{\ell-1}$, using independent random numbers $\mathbf{y}_k^{(n)}$. However, these estimators are not mutually uncorrelated. Hence, the variance of the estimator is

$$\begin{aligned}
 \mathbb{V}[\mathcal{Q}_{L,\text{reuse}}^{\text{MC}}] &= \sum_{\ell=0}^L \mathbb{V}[\Upsilon_{\ell}] + 2 \sum_{0 \leq \ell < \tau \leq L} \text{cov}(\Upsilon_{\ell}, \Upsilon_{\tau}) \\
 (2.7) \quad &= \sum_{\ell=0}^L \left(\frac{V_{\ell}}{\sum_{i=\ell}^L N_i} \right) + 2 \sum_{0 \leq \ell < \tau \leq L} \rho_{\ell\tau} \sqrt{\left(\frac{V_{\ell}}{\sum_{i=\ell}^L N_i} \right) \left(\frac{V_{\tau}}{\sum_{i=\tau}^L N_i} \right)},
 \end{aligned}$$

where $\text{cov}(\Upsilon_{\ell}, \Upsilon_{\tau})$ denotes the unknown covariance and $\rho_{\ell\tau}$ denotes the unknown correlation between Υ_{ℓ} and Υ_{τ} . Compare this to (2.3), which was a key element in allowing us to compute the required number of samples at each level. For the MG-MLMC estimator however, no such simple relationship exists, because of the covariance terms. In the next section, we will present a simple yet effective fix that allows one to compute the variance of the estimator in a mathematically rigorous way. Note that [22] just used (2.3) as an approximation for (2.7).

3. Practical aspects and implementation. In this section, we discuss a particular implementation of the MG-MLMC method and analyze its cost. We also present a practical algorithm for MG-MLMC simulation.

3.1. Obtaining an estimate for the variance. There are several ways to obtain an estimate for the variance in (2.7).

- As a first idea, we could assume that the correlation coefficients $\rho_{\ell\tau}$ in (2.7) are all equal to 1. This allows the computation of the optimal number of samples N_{ℓ} on each level as a function of V_{ℓ} and C_{ℓ} , similar to (2.4). However, we find numerically that the variance of the estimator, obtained in this way, is a huge overestimation. Hence, the required number of samples is too large, and we do not obtain an efficient estimator.
- Alternatively, we can use techniques known from adaptive MLMC methods; see, e.g., [8, 9, 21]. These methods deal with sample-dependent refinements, either using an a priori error estimate or a trial run to determine the adaptive mesh hierarchy. A variance estimate can be obtained using the debiasing technique from Rhee and Glynn; see [31]. See also [5] for details on how this can be applied in the usual MLMC setting.
- Finally, we can use the random shifting technique known from quasi-Monte Carlo (QMC) literature. Here, the idea is to use R independent estimators and use the sample variance over these estimators to replace (2.7). Since QMC methods have the potential to also obtain a faster convergence of the RMSE compared to standard Monte Carlo, we will choose this third approach.

3.2. Multigrid Multilevel Quasi-Monte Carlo. In the QMC method, the random samples from the Monte Carlo method are replaced by deterministically well-chosen sample points. These points are commonly known as QMC points; see, e.g., [6]. Popular choices include Sobol sequences [39] and digital nets [28]. In this article, we consider rank-1 lattice rules in the s -dimensional unit cube $[0, 1]^s$, where the points are given by

$$\mathbf{t}^{(n)} := \frac{n\mathbf{z} \bmod N}{N} = \left\{ \frac{n\mathbf{z}}{N} \right\}, \quad n = 0, 1, \dots, N-1$$

with $\mathbf{z} \in \mathbb{Z}_N^s$ as a generating vector and $\{\cdot\}$ denotes the mod 1 or fractional part operation; see, e.g., [29, 38].

To calculate the MSE in (2.2), we use multiple randomly shifted lattice rules. The r th shifted version of the n th point for level ℓ is then given by

$$\mathbf{t}_{\ell,r}^{(n)} := \left\{ \frac{n\mathbf{z}}{N} + \Delta_{\ell}^{(r)} \right\}, \quad r = 1, 2, \dots, R,$$

where R is the number of shifts and $\Delta_{\ell}^{(r)}$ are independent and identically distributed (i.i.d.) samples from a uniform distribution on $[0, 1]^s$. It can be shown that for functions with sufficient smoothness and dimensions that become progressively less important, there exist lattice rules for which the RMSE decays as $O(N^{-1+\epsilon})$; for arbitrary $\epsilon > 0$, see [26]. For the numerical experiments in this paper we will consider a log-normal random field with Matérn covariance which will be represented by an infinite KL expansion. The dimensionality s of the QMC points is then equal to the number of terms in the truncated KL expansion; this will be discussed in section 5. In general, QMC error bounds are specified for integration over the s -dimensional unit cube, however the KL expansion here uses i.i.d. standard normal variates. Luckily, for randomly shifted lattice rules mapped through the inverse cumulative density of the normal distribution, as we will use here, there is an equivalent theory which shows $O(N^{-1+\epsilon})$ convergence for the RMSE; see, e.g., [24].

We denote the multilevel estimator that uses the r th randomized point set by

$$\mathcal{Q}_{L,r}^{\text{QMC}} := \sum_{\ell=0}^L \Upsilon_{\ell,r}, \quad \text{where}$$

$$\Upsilon_{\ell,r} := \frac{1}{N_{\ell}} \sum_{n=1}^{N_{\ell}} \left(F_{\ell}(\mathbf{y}_{\ell,r}^{(n)}) - F_{\ell-1}(\mathbf{y}_{\ell,r}^{(n)}) \right), \quad r = 1, 2, \dots, R$$

with $\mathbf{y}_{\ell,r}^{(n)} := \Phi^{-1}(\mathbf{t}_{\ell,r}^{(n)})$ and where Φ^{-1} denotes the inverse cumulative density function (CDF) of the desired distribution of the parameter set \mathbf{y} . This provides the required mapping for the QMC points, defined on the unit cube, to the domain of interest.

The MLQMC estimator, as introduced in [14, 25, 32], is just the average over these R estimators, i.e.,

$$\bar{\mathcal{Q}}_{L,R}^{\text{QMC}} := \frac{1}{R} \sum_{r=1}^R \mathcal{Q}_{L,r}^{\text{QMC}}.$$

The sample variance over the shifts is used as an estimate for the variance, i.e.,

$$(3.1) \quad \mathbb{V}[\bar{\mathcal{Q}}_{L,R}^{\text{QMC}}] \approx \frac{1}{R(R-1)} \sum_{r=1}^R \left(\mathcal{Q}_{L,r}^{\text{QMC}} - \bar{\mathcal{Q}}_{L,R}^{\text{QMC}} \right)^2.$$

Note that in the limit $R \rightarrow \infty$ this sample variance is (almost surely) equal to the true variance of the estimator.

A similar approach is now used for the MG-MLMC estimator from (2.5), i.e., we define

$$\mathcal{Q}_{L,r,\text{reuse}}^{\text{QMC}} := \sum_{\ell=0}^L \Upsilon_{\ell,r,\text{reuse}},$$

$$(3.2) \quad \Upsilon_{\ell,r,\text{reuse}} := \left(\frac{1}{\sum_{i=\ell}^L N_i} \right) \sum_{k=\ell}^L \sum_{n=1}^{N_k} \left(F_{\ell}(\mathbf{y}_{k,r}^{(n)}) - F_{\ell-1}(\mathbf{y}_{k,r}^{(n)}) \right), \quad \text{and}$$

$$(3.3) \quad \bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}} := \frac{1}{R} \sum_{r=1}^R \mathcal{Q}_{L,r,\text{reuse}}^{\text{QMC}},$$

in effect turning the MG-MLMC estimator into an MG-MLQMC estimator. The variance of this estimator, that is, the equivalent of (2.7), is computed similar to (3.1), i.e.,

$$(3.4) \quad \mathbb{V}[\bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}}] \approx \frac{1}{R(R-1)} \sum_{r=1}^R \left(\mathcal{Q}_{L,r,\text{reuse}}^{\text{QMC}} - \bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}} \right)^2.$$

The MLQMC estimator (3.1) and the MG-MLQMC estimator (3.3) are essentially the same, the only difference being the recycling of samples on coarser levels.

3.3. Cost analysis. In this section, we will analyze the cost of the MG-MLQMC estimator (i.e., the MLQMC method with recycling) and compare it to the cost of the MLQMC method without recycling. For ease of exposition, assume that the quantity of interest F_ℓ at level ℓ corresponds to a discretization of the problem with step size h_ℓ . As in [20], we use a geometric hierarchy for the level, i.e.,

$$(3.5) \quad h_\ell = \varrho h_{\ell+1},$$

where $\varrho > 1$ is a scaling factor. In the PDE setting, a typical value is $\varrho = 2$.

As is usual in MLMC analysis (see, e.g., [4, Theorem 1]), we make the following assumptions:

$$(A1) \quad |\mathbb{E}[F_L - F]| \leq c_1 h_L^\alpha,$$

$$(A2) \quad V_\ell \leq c_2 h_\ell^\beta,$$

$$(A3) \quad C_\ell \leq c_3 h_\ell^{-\gamma}$$

for some constants c_1, c_2 and c_3 . Here, $\alpha > 0$ is the rate of the decrease of the bias in terms of L , $\beta > 0$ is the rate of the decrease of the variance of the differences in terms of the levels ℓ , and $\gamma > 0$ gives the increase in cost of solving the PDE. Additionally, we make an assumption on the convergence rate of the lattice rule used in the MLQMC estimator. That is, we define

$$(3.6) \quad \bar{\Upsilon}_\ell := \frac{1}{R} \sum_{r=1}^R \Upsilon_{\ell,r},$$

the approximation for the expected value of the difference $F_\ell - F_{\ell-1}$, using a randomly shifted lattice rule, averaged over all shifts, and assume

$$(A4) \quad \mathbb{V}[\bar{\Upsilon}_\ell] \leq c_4 N_\ell^{-1/\lambda} V_\ell$$

for some constant $c_4 > 0$ with $1/\lambda > 0$ the rate of convergence of the randomized lattice rule. Assumption (A4) corresponds to assumption (M2) in [25, Theorem 1].

We proceed as in section 2.1 and minimize the cost of the MLQMC estimator without recycling given the variance constraint, assuming the N_ℓ are continuous variables:

$$\begin{aligned} & \min_{N_\ell} \sum_{\ell=0}^L N_\ell c_3 h_\ell^{-\gamma} \\ & \text{subject to } \sum_{\ell=0}^L c_4 N_\ell^{-1/\lambda} c_2 h_\ell^\beta \leq \epsilon^2/2. \end{aligned}$$

To this end, we consider the Lagrangian

$$\mathcal{L}(N_0, N_1, \dots, N_L, \zeta) := c_3 \sum_{\ell=0}^L N_\ell h_\ell^{-\gamma} + \zeta \left(c_4 c_2 \sum_{\ell=0}^L N_\ell^{-1/\lambda} h_\ell^\beta - \epsilon^2/2 \right),$$

where ζ is the Lagrange multiplier, and look for its stationary point. This leads to the first-order, necessary optimality conditions

$$(3.7) \quad \begin{aligned} \frac{\partial \mathcal{L}}{\partial N_\ell} &= c_3 h_\ell^{-\gamma} - c_4 c_2 \frac{\zeta}{\lambda} N_\ell^{-1/\lambda-1} h_\ell^\beta = 0 \quad \text{for } \ell = 0, \dots, L, \quad \text{and} \\ \frac{\partial \mathcal{L}}{\partial \zeta} &= c_4 c_2 \sum_{\ell=0}^L N_\ell^{-1/\lambda} h_\ell^\beta - \epsilon^2/2 = 0. \end{aligned}$$

Rearranging (3.7), we find that

$$(3.8) \quad N_\ell = \left(\frac{c_4 c_2}{c_3} \frac{\zeta}{\lambda} h_\ell^{\beta+\gamma} \right)^{\frac{\lambda}{\lambda+1}}.$$

Note that this equation for N_ℓ is of limited practical use, because estimates for λ , and hence for the convergence rate $1/\lambda$ of the MSE of the lattice rule, are very unreliable [25]. A simple algorithm that adaptively increases the number of samples N_ℓ , but does not require knowledge of λ , will be presented in section 3.4.

PROPOSITION 3.1. *The cost reduction factor of the MG-MLQMC algorithm using a geometric refinement $h_\ell = \varrho h_{\ell+1}$ is given by*

$$\frac{\text{cost}(\bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}})}{\text{cost}(\bar{\mathcal{Q}}_{L,R}^{\text{QMC}})} = 1 - \varrho^{-(\beta+\gamma)\lambda/(\lambda+1)},$$

where β is the rate of the decrease in variance and γ the rate of the increase in cost per level ℓ (see assumptions (A2) and (A3)), and $1/\lambda$ is the convergence rate of the randomized QMC method (see (A4)), or $\lambda = 1$ when using Monte Carlo.

Proof. Suppose the algorithm requires one to take N_ℓ samples on level ℓ . In the MG-MLQMC method, where samples from higher levels $\ell+1, \dots, L$ are recycled, we already have $N_{\ell+1}$ samples available on level ℓ . The remaining

$$N'_\ell := N_\ell - N_{\ell+1}$$

samples still need to be computed. From (3.8), we find that

$$N_{\ell+1} = \left(\varrho^{-(\beta+\gamma)} \right)^{\frac{\lambda}{\lambda+1}} N_\ell,$$

and hence

$$N'_\ell = \left(1 - \left(\varrho^{-(\beta+\gamma)} \right)^{\frac{\lambda}{\lambda+1}} \right) N_\ell.$$

Therefore,

$$\begin{aligned} \text{cost}(\bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}}) &= \sum_{\ell=0}^L R N'_\ell C_\ell \\ &= \left(1 - \left(\varrho^{-(\beta+\gamma)} \right)^{\frac{\lambda}{\lambda+1}} \right) \sum_{\ell=0}^L R N_\ell C_\ell \\ &= \left(1 - \left(\varrho^{-(\beta+\gamma)} \right)^{\frac{\lambda}{\lambda+1}} \right) \text{cost}(\bar{\mathcal{Q}}_{L,R}^{\text{QMC}}). \quad \square \end{aligned}$$

The recycling of coarse samples is more efficient when the rates β and γ are small, corresponding to a slow decay of the number of samples N_ℓ as ℓ increases. This is consistent with the results in [22]. There is an additional effect in the cost reduction factor that comes from using QMC points. When using rank-1 lattice rules, λ varies between 1 (Monte Carlo) and $1/2$ (optimal rank-1 lattice rule), hence $1/3 < \lambda/(\lambda + 1) \leq 1/2$, where the lower bound is obtained for the lattice rule with the fastest convergence.

In conclusion, when the decay of the variance is slow, i.e., β is small because of a lack of smoothness in the problem, or when the QMC method has good performance, i.e., $1/\lambda$ is large, the sequence of N_ℓ given in (3.8) will decrease slowly and the benefit of the sample recycling will be apparent. Conversely, if the decay of the variance is fast, or when the QMC method has bad performance, the sequence of N_ℓ will decrease very rapidly, and there is not much to be gained by reusing samples across the different levels. This is indeed what is confirmed by our numerical results in section 6.

3.4. Practical algorithm. In [14], a simple doubling algorithm is presented for MLQMC simulation. With a slight modification, this algorithm can also be used for MG-MLQMC simulation (see Algorithm 3.1). First, let us define the analogue of (3.6),

$$(3.9) \quad \tilde{\Upsilon}_{\ell, \text{reuse}} := \frac{1}{R} \sum_{r=1}^R \Upsilon_{\ell, r, \text{reuse}}.$$

Algorithm 3.1 MG-MLQMC simulation

```

1: procedure MG-MLQMC( $\epsilon$ )
2:   set defaults  $N_\ell = N_\ell^{\text{old}} = 0$ ,  $\Upsilon_{\ell, r, \text{reuse}} = \tilde{\Upsilon}_{\ell, \text{reuse}} = 0$ ,  $\mathcal{V}_\ell = +\infty$ ,  $C_\ell = 0$  for all  $\ell$  and  $r$ 
3:
4:   procedure SAMPLEANDSOLVE( $\ell$ ) ▷ FMG sample routine on level  $\ell$ 
5:     start timer
6:     for  $r = 1, \dots, R$  do
7:       for  $n = N_\ell^{\text{old}}, \dots, N_\ell - 1$  do
8:         construct random field based on  $\mathbf{y}_{\ell, r}^{(n)}$ 
9:         solve PDE using multigrid resulting in solutions for all levels  $\tau = 0, \dots, \ell$ 
10:      end for
11:      update  $\Upsilon_{\tau, r, \text{reuse}}$  using (3.2) for all  $\tau = 0, \dots, \ell$ 
12:    end for
13:    stop timer and update  $C_\ell$  based on timing for  $(N_\ell - N_\ell^{\text{old}})R$  solves
14:    update  $\tilde{\Upsilon}_{\tau, \text{reuse}}$  using (3.9) for all  $\tau = 0, \dots, \ell$ 
15:    update  $\mathcal{V}_\tau$  using (3.10) for all  $\tau = 0, \dots, \ell$ 
16:    set  $N_\ell^{\text{old}} = N_\ell$ 
17:  end procedure
18:
19:  set  $L = -1$ 
20:  while  $L < 2$  or bias estimate (3.11)  $> \epsilon/\sqrt{2}$  do
21:    set  $L = L + 1$ 
22:    while variance estimate (3.4)  $> \epsilon^2/2$  do
23:      find  $\ell \leq L$  with maximum ratio  $\mathcal{V}_\ell/C_\ell$  and set  $N_\ell = \max(1, 2N_\ell)$ 
24:      call SAMPLEANDSOLVE( $\ell$ )
25:    end while
26:  end while
27:
28: end procedure
    
```

This is the approximation for the expected value of the difference $F_\ell - F_{\ell-1}$, using the MG-MLQMC estimator, averaged over all random shifts $r = 1, 2, \dots, R$. The sample variance of $\tilde{\Upsilon}_{\ell, \text{reuse}}$ with respect to the random shifts is a measure for the contribution of that level ℓ to the total variance of the estimator, i.e., define

$$(3.10) \quad \mathcal{V}_\ell := \frac{1}{R(R-1)} \sum_{r=1}^R (\Upsilon_{\ell, r, \text{reuse}} - \tilde{\Upsilon}_{\ell, \text{reuse}})^2.$$

The idea of the algorithm is to double the number of samples where the ratio of the sample variance \mathcal{V}_ℓ and the cost C_ℓ is largest, until the total variance of the estimator is smaller than the requested tolerance $\epsilon^2/2$. The main difference between the MLQMC algorithm and the MG-MLQMC algorithm in Algorithm 3.1 is on line 9, where the additional (free) coarse samples from the FMG solver are added.

The algorithm is level-adaptive, in the sense that extra levels are added to the estimator only if required to reach a certain accuracy. The bias of the estimator, defined in (2.2), is computed as

$$(3.11) \quad \text{Bias}(\bar{\mathcal{Q}}_{L, R, \text{reuse}}^{\text{QMC}}) \approx \frac{|\tilde{\Upsilon}_{L, \text{reuse}}|}{\varrho^\alpha + 1},$$

using an estimate for α obtained from a linear fit through the expected value of the differences, $\tilde{\Upsilon}_{\ell, \text{reuse}}$, on all available levels; cf. (A1); see [12]. The condition on L on line 20 of Algorithm 3.1 ensures that enough levels are available to estimate the rate α .

4. Full Multigrid. Before turning to some numerical experiments, we give some comments on the FMG method. The basics of a multigrid method for the iterative solution of discretizations of elliptic PDEs is the correction scheme. A two-grid correction scheme can be written as the following procedure.

1. Use an iterative method with μ_1 iterations on $A_\ell \mathbf{u}_\ell = \mathbf{b}_\ell$ to approximate \mathbf{u}_ℓ .
2. Restrict the residual $\mathbf{r}_\ell = \mathbf{b}_\ell - A_\ell \mathbf{u}_\ell$ to the coarse grid by $\mathbf{r}_{\ell-1} = R_{\ell-1}^{\ell-1} \mathbf{r}_\ell$.
3. Solve $A_{\ell-1} \mathbf{e}_{\ell-1} = \mathbf{r}_{\ell-1}$.
4. Interpolate the error $\mathbf{e}_{\ell-1}$ back to the fine grid by $\mathbf{e}_\ell = I_{\ell-1}^\ell \mathbf{e}_{\ell-1}$.
5. Correct the fine-grid approximation by $\mathbf{u}_\ell = \mathbf{u}_\ell + \mathbf{e}_\ell$.
6. Use an iterative method with μ_2 iterations on $A_\ell \mathbf{u}_\ell = \mathbf{b}_\ell$ to approximate \mathbf{u}_ℓ .

Here, A_ℓ is the original matrix that results from a discretization of a sample of (1.1) at level ℓ . Matrices $R_{\ell-1}^{\ell-1}$ and $I_{\ell-1}^\ell$ are the intergrid operators, transferring quantities from the fine grid to the coarse grid and vice versa. The integers μ_1 and μ_2 are cycling parameters that control the number of iterations performed before and after the coarse grid correction. Typical values are $\mu_1 = 2$ and $\mu_2 = 1$. The iterative method used in step 1 and 6 is called a smoother because it removes high frequency components from the error; see, e.g., [18]. A single application of the smoother is known as a relaxation step.

A multilevel iterative process is eventually obtained by repeating this two-grid process of relaxation and coarsening on ever coarser grids. This is known as a multigrid V-cycle. Details on the multigrid method and why it works can be found in, e.g., [3, 17, 40].

Multigrid methods come in many flavors. One notable example is FMG, which combines the correction scheme with nested iteration; see Algorithm 4.1. First, the

Algorithm 4.1 FMG

```

1: procedure FMG( $A_\ell, \mathbf{b}_\ell$ )
2:   if on coarsest grid ( $\ell = 0$ ) then
3:     set  $\mathbf{u}_\ell = \mathbf{0}$  as initial guess
4:   else
5:     restrict the right-hand side  $\mathbf{b}_\ell$  to the next coarser grid by  $\mathbf{b}_{\ell-1} = R_\ell^{\ell-1} \mathbf{b}_\ell$ 
6:     call FMG( $A_{\ell-1}, \mathbf{b}_{\ell-1}$ ) to calculate  $\mathbf{u}_{\ell-1}$ 
7:     interpolate the solution  $\mathbf{u}_{\ell-1}$  back to the fine grid by  $\mathbf{u}_\ell = I_{\ell-1}^\ell \mathbf{u}_{\ell-1}$ 
8:   end if
9:   repeat  $\mu_0$  times
10:    call V-CYCLE( $\ell$ )
11:  end
12:  return the solution  $\mathbf{u}_\ell$ 
13: end procedure
    
```

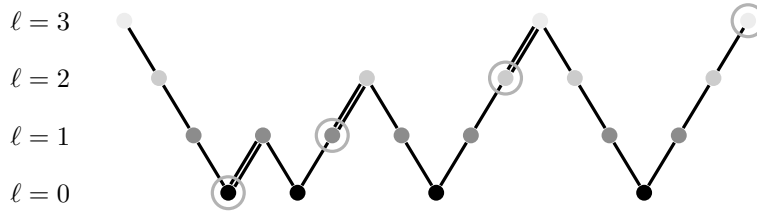


FIG. 1. Schematic representation of the sequence of grids visited in the FMG solver for $\mu_0 = 1$ and $\ell = 3$. The coarse solutions at the end of every V-cycle, encircled on the figure, are eventually reused in the MLMC estimator. The FMG interpolation scheme is indicated by the double line.

right-hand side \mathbf{b}_ℓ is transferred to the coarser grids using the restriction operator $R_\ell^{\ell-1}$. Starting from the coarse grid on level ℓ , a good initial guess for the next finer grid on level $\ell + 1$ is obtained by interpolation of the inexpensive-to-compute coarse-grid solution. Typically, a higher-order interpolation scheme is used in this step. After that, a number of multilevel correction cycles are performed. We denote this number by μ_0 . The process is then repeated until a good initial guess for an iterative method on the finest grid (level L) is obtained. As a result, we do not only solve the PDE on the finest grid, but we also get solutions on all coarser grids, as required. The sequence of levels visited, starting from the finest level, are visualized in Figure 1. The discrete solutions $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_\ell$ that are eventually used in the estimator are encircled on the figure. The FMG interpolation scheme (line 7 in Algorithm 4.1) is indicated by the double line.

A large number of efficient (black-box) multigrid solvers can be found in the literature; see, e.g., [36, 41]. Typically, these codes do not provide access to the solutions on coarser grid levels. However, a multigrid solver for elliptic problems is easy to construct.

5. Random field generation using the KL expansion. In this section, we focus on how to obtain a sample of the diffusion coefficient in (1.1), represented as the random field $a(\mathbf{x}, \mathbf{y})$, given the set of parameters \mathbf{y} . We assume the diffusion coefficient is derived from a Gaussian random field $z(\mathbf{x}, \mathbf{y})$ that is fully characterized by its mean $\bar{z}(\mathbf{x})$ and covariance function $C(\mathbf{x}, \mathbf{x}')$. In our numerical experiments later, we will use a so-called lognormal random field $a(\mathbf{x}, \mathbf{y}) = \exp(z(\mathbf{x}, \mathbf{y}))$; see [4].

This ensures that (1.1) has physical meaning, i.e., the diffusion coefficient remains positive for all samples.

The KL expansion is a common approach to represent $z(\mathbf{x}, \mathbf{y})$; see, e.g., [27]. The idea is to express the random field via a sequence of independent random variables \mathbf{y} in the following way:

$$(5.1) \quad z(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{\infty} y_j \sqrt{\theta_j} \psi_j(\mathbf{x}).$$

The eigenvalues θ_j and eigenfunctions $\psi_j(\mathbf{x})$ are the solutions of the eigenvalue problem

$$(5.2) \quad \int_D C(\mathbf{x}, \mathbf{x}') \psi_j(\mathbf{x}') d\mathbf{x}' = \theta_j \psi_j(\mathbf{x}).$$

Truncating the series in (5.1) at the s th term gives the truncated KL expansion

$$(5.3) \quad z_s(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^s y_j \sqrt{\theta_j} \psi_j(\mathbf{x}).$$

In the special case of Gaussian random fields, it can be shown that the random parameters \mathbf{y} in (5.1) are i.i.d. and standard normally distributed.

It is well-known that the KL expansion is optimal in the mean square sense, i.e., when truncated after a finite number of terms s , the resulting approximation $z_s(\mathbf{x}, \mathbf{y})$ minimizes the MSE; see, for instance, [11]. The mean square truncation error decreases monotonically to zero with the number of terms in the expansion, at a rate that depends on the covariance function $C(\mathbf{x}, \mathbf{x}')$.

If the eigenpairs (θ_j, ψ_j) in the KL expansion are not available, we need to approximate them using a suitable numerical scheme. One possibility is to perform collocation on (5.2), i.e., solve

$$(5.4) \quad \int_D C(\mathbf{x}_k, \mathbf{x}') \psi_j(\mathbf{x}') d\mathbf{x}' = \theta_j \psi_j(\mathbf{x}_k), \quad k = 1, 2, \dots, M$$

in some well-chosen integration points \mathbf{x}_k . In the Nyström method [1], the integral in (5.4) is approximated by a suitable numerical integration scheme, which uses the collocation points as quadrature nodes:

$$(5.5) \quad \sum_{q=1}^M w_q C(\mathbf{x}_k, \mathbf{x}_q) \tilde{\psi}_j(\mathbf{x}_q) = \tilde{\theta}_j \tilde{\psi}_j(\mathbf{x}_k), \quad k = 1, 2, \dots, M.$$

In matrix notation,

$$(5.6) \quad \Sigma W \tilde{\Psi}_j = \tilde{\theta}_j \tilde{\Psi}_j,$$

where Σ is a symmetric positive semidefinite matrix with entries $\Sigma_{k,q} = C(\mathbf{x}_k, \mathbf{x}_q)$, W is a diagonal matrix with the weights w_q on the diagonal, and $\tilde{\Psi}_j$ is a vector with entries $\tilde{\Psi}_{j,q} = \tilde{\psi}_j(\mathbf{x}_q)$. In general, ΣW is a dense and nonsymmetric matrix. However, the matrix eigenvalue problem (5.6) can be reformulated into an equivalent matrix eigenvalue problem

$$B \tilde{\Psi}_j^* = \tilde{\theta}_j \tilde{\Psi}_j^*,$$

where $\tilde{\Psi}_j^* = \sqrt{W}\tilde{\Psi}_j$, and $B = \sqrt{W}\Sigma\sqrt{W}$ is symmetric positive semidefinite. Hence, the eigenvalues $\tilde{\theta}_j$ are real and nonnegative and the eigenvectors $\tilde{\Psi}_j^*$ are orthogonal to each other.

Using (5.5), we obtain the so-called Nyström interpolation formula for the eigenfunctions $\tilde{\psi}_j(\mathbf{x})$, i.e.,

$$\tilde{\psi}_j(\mathbf{x}) = \frac{1}{\tilde{\theta}_j} \sum_{q=1}^M \sqrt{w_q} \tilde{\Psi}_{j,q}^* C(\mathbf{x}, \mathbf{x}_q),$$

with $\tilde{\Psi}_{j,q}^*$ the q th element of the eigenvector $\tilde{\Psi}_j^*$. After a suitable normalization, these eigenvalues and eigenfunctions can be used as an approximate eigenpair in the KL expansion from (5.3).

6. Numerical results. In this section we perform numerical experiments on problem (1.1) that illustrate the power of the proposed algorithm.

Discretization. Consider the model elliptic PDE from (1.1) defined on a two-dimensional computational domain $D = [0, 1]^2$ with source term $f(\mathbf{x}) = 1$ and homogenous boundary conditions of Dirichlet-type. We discretize the equation using finite differences on a hierarchy of regular grids with nodes

$$(x_1^{(i)}, x_2^{(j)}) = \left(\frac{i}{m_\ell}, \frac{j}{m_\ell} \right), \quad i, j = 0, \dots, m_\ell,$$

where $m_\ell = 1/h_\ell$ and h_ℓ satisfies (3.5) with $h_0 = 1/4$ and $\varrho = 2$.

Modeling of uncertain parameters. The diffusion coefficient $a(\mathbf{x}, \mathbf{y})$ is modeled as a lognormal random field, i.e., $a(\mathbf{x}, \mathbf{y}) = \exp(z(\mathbf{x}, \mathbf{y}))$ with $\bar{z}(\mathbf{x}) = 0$, and $C(\mathbf{x}, \mathbf{x}')$ is the Matérn covariance function with correlation length λ_c and smoothness ν , defined as

$$(6.1) \quad C(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu\rho} \right)^\nu K_\nu \left(\sqrt{2\nu\rho} \right), \quad \rho = \frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\lambda_c}.$$

In the following, we consider the parameter sets $\lambda_c = \{0.1, 0.3, 0.5\}$ and $\nu = \{0.5, 1, 2\}$. A typical realization of the random field $z(\mathbf{x}, \mathbf{y})$ for all parameter combinations of correlation length λ_c and smoothness ν is shown in Figure 2. Each of the random fields is represented using the truncated KL expansion in (5.3), where the number of terms in the expansion is chosen according to Table 1. In particular, we used the identity

$$\sum_{j=1}^{\infty} \theta_j = \int_D d\mathbf{x} = 1$$

to ensure that at least 99.8% of the variability in the random field is captured by the first s terms; see [10]. The eigenvalues and eigenfunctions are approximated numerically using the Nyström method outlined in section 5.

Figure 3 shows the decay of the eigenvalues $\tilde{\theta}_j$ for each parameter combination. Note that as the smoothness ν increases, the eigenvalues decay faster and fewer terms are required in the expansion. For decreasing correlation lengths λ_c , there is a growing preasymptotic phase where the KL eigenvalues do not decay. Note also that there is an end effect for the smallest eigenvalues, where the asymptotic decay rate is not preserved. Similar numerical results in case of a circulant embedding method and corresponding analysis can be found in [15].

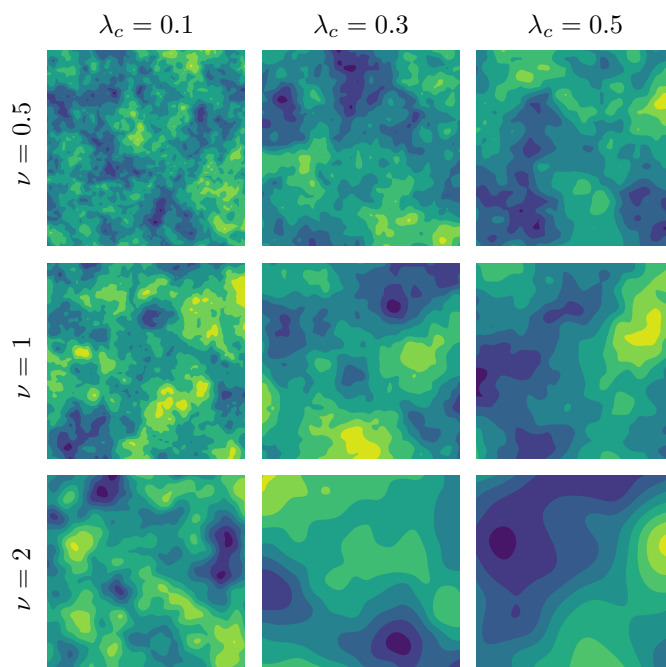


FIG. 2. Typical realizations of some Gaussian random fields for various combinations of the correlation length λ_c and smoothness ν in Matérn covariance function (6.1). These realizations are computed using the truncated KL expansion as outlined in section 5, where the number of terms in the expansion is chosen according to Table 1.

TABLE 1

Number of terms in the KL expansion used to represent the different Gaussian random fields in the numerical experiments. As the smoothness ν in the Matérn covariance increases, fewer terms are needed to accurately model the random fields using a KL expansion.

	$\lambda_c = 0.1$	$\lambda_c = 0.3$	$\lambda_c = 0.5$
$\nu = 0.5$	3,500	1,000	500
$\nu = 1$	3,500	1,000	500
$\nu = 2$	1,000	250	100

Quantity of interest. In the numerical experiments below, we investigate two different quantities of interest. As a first quantity of interest, we choose the value of $u(\mathbf{x}, \cdot)$ at $\mathbf{x} = (1/2, 1/2)$. This simple point evaluation has also been considered in [16, section 6], where it was shown to have some special properties because of the symmetry in the problem. As a second quantity of interest, we consider the averaged value of u over the domain $D_{\text{sub}} = [1/4, 1/2]^2$, that is,

$$F(u) = \int_{D_{\text{sub}}} u(\mathbf{x}, \mathbf{y}) d\mathbf{x} \bigg/ \int_{D_{\text{sub}}} d\mathbf{x} = 16 \int_{D_{\text{sub}}} u(\mathbf{x}, \mathbf{y}) d\mathbf{x}.$$

In our numerical experiments, the integral in the second quantity of interest is approximated using the two-dimensional trapezoidal quadrature rule.

Deterministic solver. Every sample of (1.1) is solved using the FMG approach outlined in section 4. Some further details are given below.

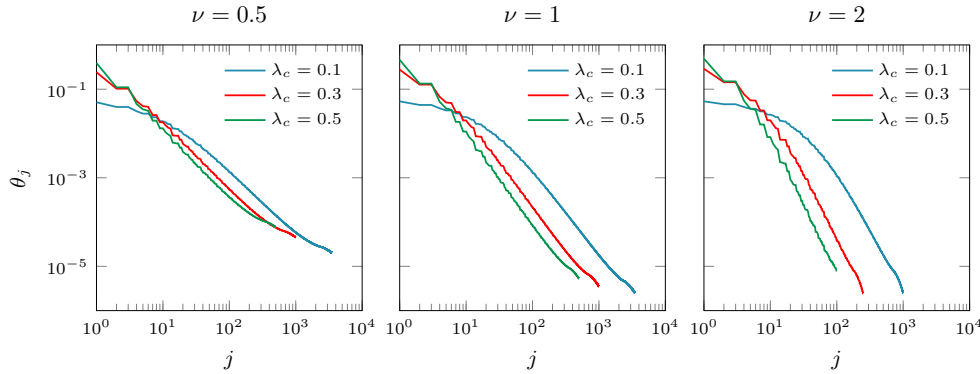


FIG. 3. Decay of the eigenvalues in the KL expansion from section 5 for various combinations of the correlation length λ_c and smoothness ν in the Matérn covariance function.

Coarse-grid operators. The *direct-discretization* approach is used to obtain the coarse-grid operators A_ℓ . In particular, we apply finite differences to the random field computed using the discrete version of the KL expansion from section 5. In this way, the samples that are recycled from the FMG method, correspond to valid samples of the problem defined on a coarser mesh. Alternatively, one can use a so-called (*Petrov*)–*Galerkin* approach to obtain the coarse grid problems, i.e., $A_{\ell-1} = R_\ell^{\ell-1} A_\ell I_{\ell-1}^\ell$. However, in that case, one must ensure that the coarse problems correspond to discrete versions of the original problem on those grids.

Pre- and postsmoothing. We used symmetric Gauss–Seidel (GS) [37] as a smoother, with multigrid cycle parameters $\mu_1 = 3$ and $\mu_2 = 2$ for random fields with smoothness $\nu = 1$ and $\nu = 2$ (rows 2 and 3 in Table 1), and with multigrid cycle parameters $\mu_1 = 6$ and $\mu_2 = 4$ for random fields with smoothness $\nu = 0.5$ (the top row in Table 1).

Transfer operators. Standard (full weighting) restriction and (bilinear) interpolation is used to transfer quantities from the fine grid to the coarse grid and back. In stencil format, we have

$$(6.2) \quad [R_{\ell+1}^\ell] = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{\ell+1}^\ell \quad \text{and} \quad [I_\ell^{\ell+1}] = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_\ell^{\ell+1};$$

see, e.g., [40] for more details. In the FMG method, cubic interpolation is used to transfer the coarse-grid solutions to the finer grid.

Coarse grid correction. We found that a small damping factor in the multigrid V-cycle algorithm significantly improves the convergence rate of the method. That is, line 5 in the two-grid correction scheme becomes $\mathbf{u}_\ell = \mathbf{u}_\ell + \delta \mathbf{e}_\ell$, where $\delta = 0.95$ for random fields with smoothness $\nu = 1$ and $\nu = 2$, and $\delta = 0.85$ for random fields with smoothness $\nu = 0.5$.

FMG. The number of FMG correction cycles μ_0 is chosen adaptively by controlling the (discrete) norm of the residual $\mathbf{r}_\ell = \mathbf{b}_\ell - A_\ell \mathbf{u}_\ell$. In particular, we require $\|\mathbf{r}_\ell\|_{h_\ell} \leq 1/h_\ell^2$. In this way, we ensure that solutions on coarser FMG levels converged to discretization accuracy. In [23], it is shown that the number of multigrid correction cycles μ_0 depends on both the grid size and the smoothness parameter ν of the Matérn covariance function.

Cost estimate. The actual time needed to compute a solution on each level was used as an estimate for the cost C_ℓ .

QMC method. We used a randomly shifted rank-1 lattice rule with a generating vector computed using the fast component-by-component construction from the QMC4PDE website [30]. We used a total of $R = 20$ independent random shifts. In principle, this number could be made dependent on the level, i.e., $R = R_\ell$; see the analysis in [25]. In practice, however, it is easier to keep the number of shifts fixed; see also [12, 25] which use respectively 16 and 32 shifts. Note that, since the total number of points in the lattice rule is not fixed a priori, the lattice rule is actually a lattice sequence; see [6] for details.

Computing environment. All our simulations are performed on a workstation with 20 Intel Xeon E5645 processors running at 2.4 GHz and using a total of 48 GB of RAM. All samples are taken in parallel. Since the Monte Carlo sampling is embarrassingly parallel, we observed almost linear speedup with respect to the number of cores. For the QMC-based methods, we ran an independent estimator on each core, where the number of shifts equaled the number of computational cores. For details on work scheduling of MLMC samplers on parallel machines, we refer to [7].

The MG-MLQMC method is implemented within the framework of the Julia package `MultilevelEstimators.jl` (see [33]), and we used this software in all our experiments.

6.1. Numerical rate determination. We begin our numerical tests by determining the rates α , β , γ , and λ in assumptions (A1)–(A4).

First quantity of interest. Figures 4 and 5 are standard figures used in MLMC literature. They plot the behavior of expected value (Figure 4) and variance (Figure 5) of the quantity of interest F_ℓ and the difference $F_\ell - F_{\ell-1}$ as a function of the level parameter ℓ . Observe that the estimated rate α in assumption (A1) is approximately constant for all parameter combinations listed in Table 1. The estimated rate β in assumption (A2) increases as the smoothness ν and correlation length λ_c in the Matérn covariance function increase. In addition, note that for $\lambda_c = 0.1$, the graphs of $\mathbb{V}[F_\ell]$ and $\mathbb{V}[F_\ell - F_{\ell-1}]$ intersect between $\ell = 1$ and $\ell = 2$, indicating that the coarse level uses too few degrees of freedom. In this case, there is no benefit in including the coarsest mesh into the hierarchy. However, for the sake of uniformity, we will use the same mesh hierarchy for all test cases.

Figure 6 shows the variance of the QMC estimator for the expected value of the differences $F_\ell - F_{\ell-1}$ as the number of samples N_ℓ is increased. Note that the numerically observed rate λ from assumption (A4) seems to decrease (that is, the lattice rule performs better) as the correlation length λ_c increases. However, from our analysis in section 3.3, and specifically Proposition 3.1, it follows that this effect is negligible compared to the increase of the rate β . Observe also the variance reduction from level to level, i.e., the offset between the lines.

Finally, the rate γ in assumption (A3) is fixed by the FMG solver. We found numerically that $\gamma \approx 2.12$.

For the case of MLQMC with a Matérn covariance, [25, Remark 10] shows that for the QMC convergence theory to be valid we need $\nu > d + 1$. This would require us to choose $\nu > 3$ in our experiments. However, in [25] it is also remarked that even when this condition is violated in the experiments the MLQMC algorithm still showed consistent advantage.

Second quantity of interest. The numerically observed rates α , β , and λ are shown in Table 2. Again, we observe that the estimated rate β in assumption (A2) increases as the smoothness ν and correlation length λ_c in the Matérn covariance function increase, while the estimated rate α in assumption (A1) remains approximately con-

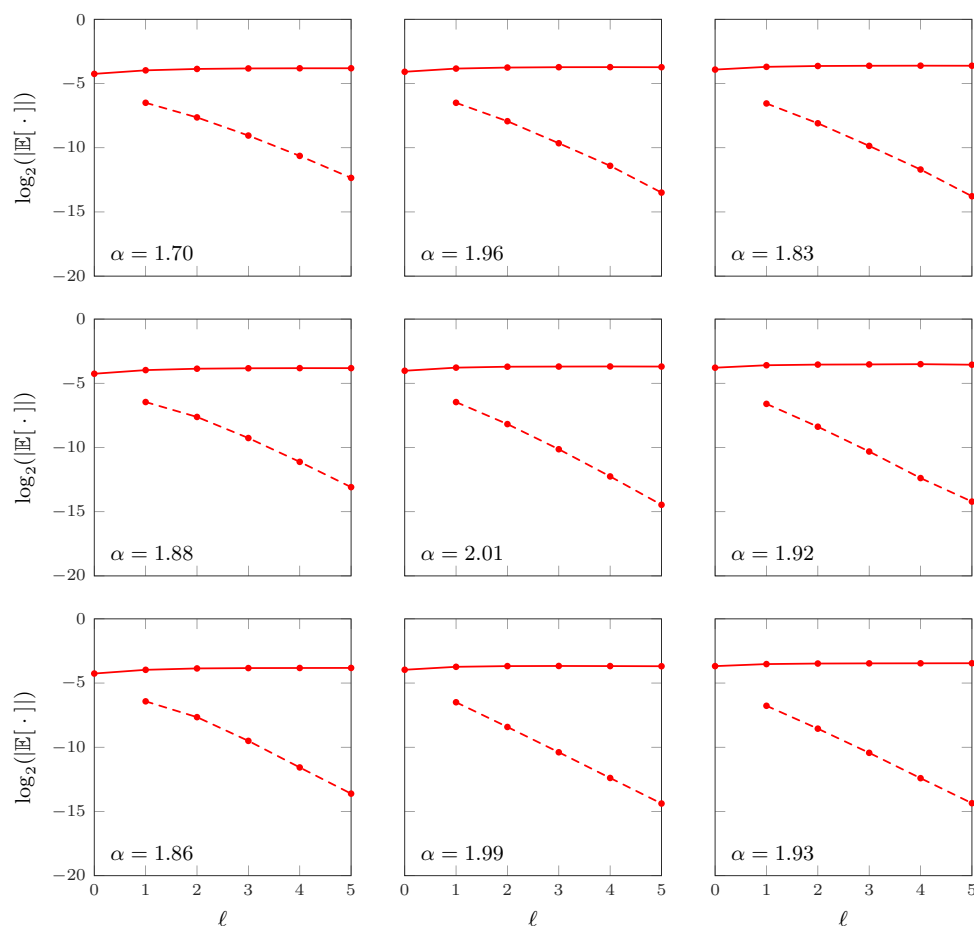


FIG. 4. Expected values $\mathbb{E}[F_\ell]$ (—•—) and $\mathbb{E}[F_\ell - F_{\ell-1}]$ (- -• -) for $\lambda_c = 0.1$ (left column), $\lambda_c = 0.3$ (middle column), $\lambda_c = 0.5$ (right column), and $\nu = 0.5$ (top row), $\nu = 1$ (middle row), $\nu = 2$ (bottom row) for the first quantity of interest (point evaluation). The estimated rate α corresponds to the rate in assumption (A1).

stant. The numerically observed rate λ from assumption (A4) decreases (that is, the lattice rule performs better), as the smoothness ν increases.

Based on the analysis from section 3.3, we thus expect that the sample recycling is less efficient when

- (i) the smoothness ν of the covariance function increases (i.e., moving from top to bottom in Table 1), and
- (ii) the correlation length λ_c of the covariance function increases (i.e., moving from left to right in Table 1).

6.2. Results and comparison. We compare the performance of MLQMC, with and without recycling, against a basic MLMC simulation.

First quantity of interest. We ran all simulations for a sequence of decreasing tolerances, starting from $\epsilon = 0.001$. Convergence plots are shown in Figure 7. Observe that the asymptotic complexity rate of the QMC methods is increasingly better (that is, the slope of the lines decrease) as the smoothness and correlation length in the Matérn covariance function increase, i.e., when moving from top to bottom and left

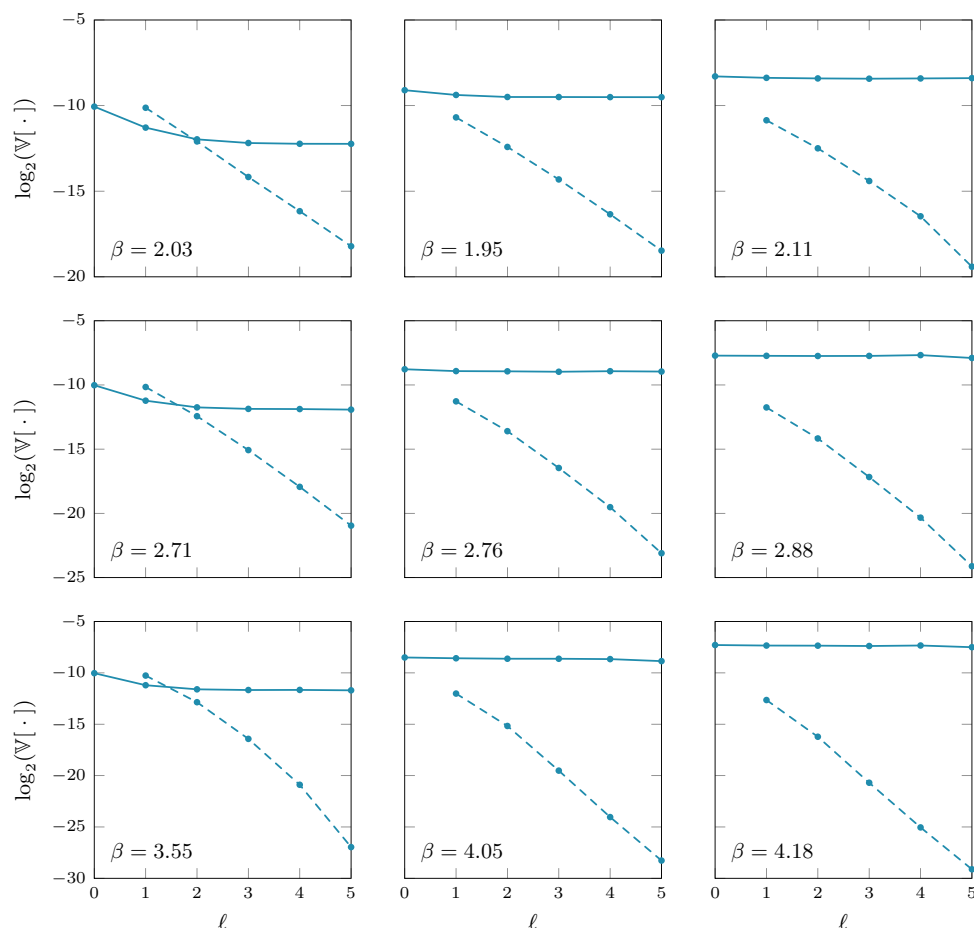


FIG. 5. Variances $\mathbb{V}[F_\ell]$ (—•—) and $\mathbb{V}[F_\ell - F_{\ell-1}]$ (-•-) for $\lambda_c = 0.1$ (left column), $\lambda_c = 0.3$ (middle column), $\lambda_c = 0.5$ (right column), and $\nu = 0.5$ (top row), $\nu = 1$ (middle row), $\nu = 2$ (bottom row) for the first quantity of interest (point evaluation). The estimated rate β corresponds to the rate in assumption (A2).

to right. In particular, this means that fewer computational resources are needed to achieve a certain tolerance on the MSE (2.2) (note the logarithmic time axis). Also, and consistent with our previous analysis, we observe that the benefit of the sample recycling decreases accordingly. That is, the lines for MG-MLQMC and MLQMC are closer together when moving towards the bottom right of Figure 7.

These results are summarized in Table 3 (left), where the speedup is shown for all parameter combinations, averaged over all tolerances ϵ . The first column shows the speedup of MLQMC over MLMC, i.e.,

$$S_1 = \frac{\text{time}(\mathcal{Q}_L^{\text{MC}})}{\text{time}(\bar{\mathcal{Q}}_{L,R}^{\text{QMC}})}.$$

Columns 2 and 3 show

$$S_2 = \frac{\text{time}(\mathcal{Q}_L^{\text{MC}})}{\text{time}(\bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}})} \quad \text{and} \quad S_3 = \frac{\text{time}(\bar{\mathcal{Q}}_{L,R}^{\text{QMC}})}{\text{time}(\bar{\mathcal{Q}}_{L,R,\text{reuse}}^{\text{QMC}})},$$

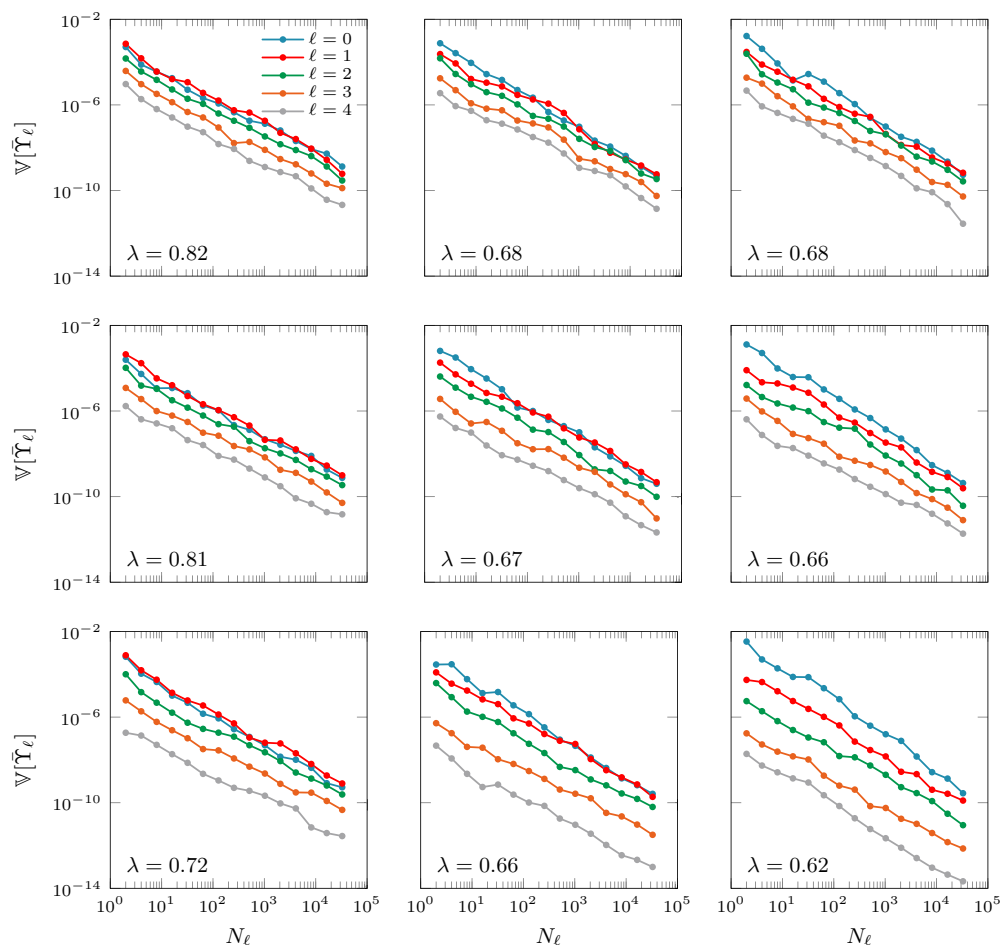


FIG. 6. Variance of the QMC estimator for the expected value of the difference $F_\ell - F_{\ell-1}$, averaged over all random shifts, for different levels and as function of the number of samples N_ℓ for $\lambda_c = 0.1$ (left column), $\lambda_c = 0.3$ (middle column), $\lambda_c = 0.5$ (right column), and $\nu = 0.5$ (top row), $\nu = 1$ (middle row), $\nu = 2$ (bottom row) for the first quantity of interest (point evaluation). The estimated rate λ (averaged over all levels) corresponds to the rate in assumption (A4).

TABLE 2

Numerical rate estimation in assumptions (A1)–(A4) for the second quantity of interest (an average over a subdomain). These rates have been determined in a similar way as in Figures 4–6.

	$\lambda_c = 0.1$	$\lambda_c = 0.3$	$\lambda_c = 0.5$
$\nu = 0.5$	$\alpha = 2.09$	$\alpha = 1.94$	$\alpha = 1.86$
	$\beta = 3.46$	$\beta = 3.45$	$\beta = 3.87$
	$\lambda = 0.92$	$\lambda = 0.87$	$\lambda = 0.94$
$\nu = 1$	$\alpha = 1.92$	$\alpha = 1.92$	$\alpha = 2.17$
	$\beta = 3.51$	$\beta = 3.93$	$\beta = 4.46$
	$\lambda = 0.88$	$\lambda = 0.85$	$\lambda = 0.80$
$\nu = 2$	$\alpha = 2.05$	$\alpha = 2.11$	$\alpha = 2.02$
	$\beta = 3.40$	$\beta = 4.32$	$\beta = 4.56$
	$\lambda = 0.83$	$\lambda = 0.76$	$\lambda = 0.80$

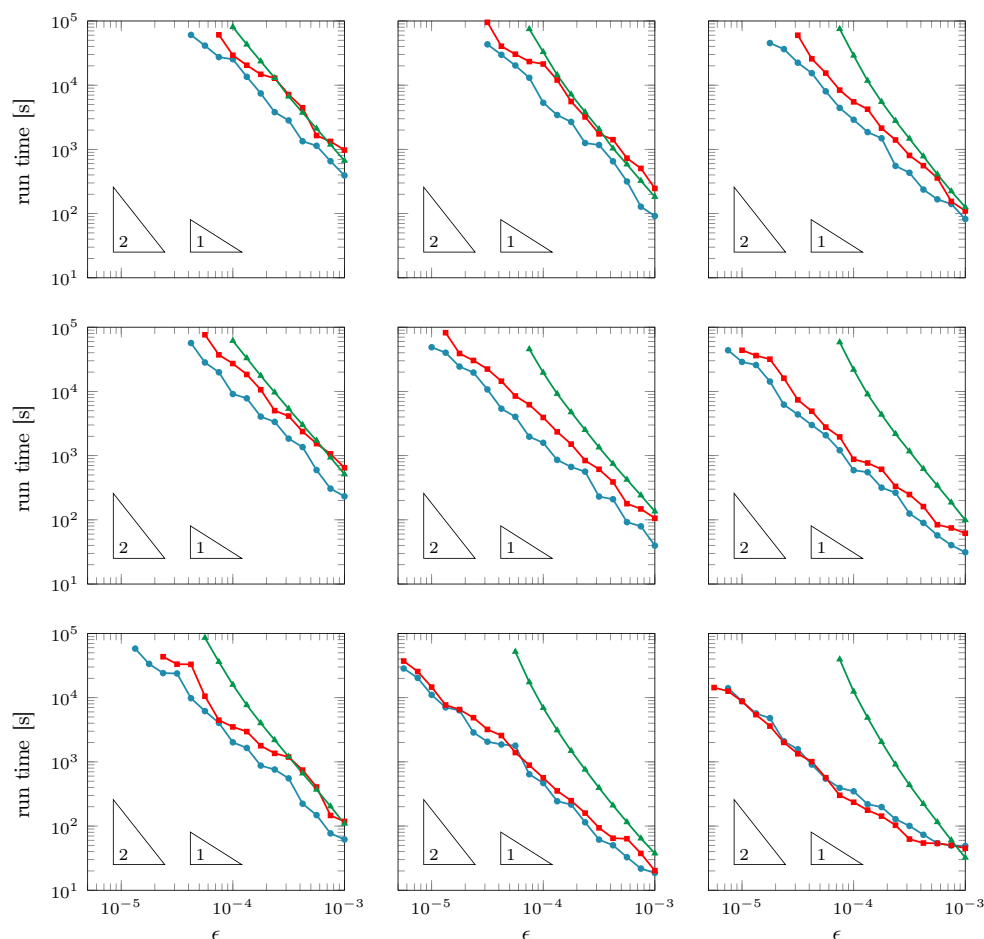


FIG. 7. Comparison of the run time between MLMC (—●—), MLQMC (—●—), and MG-MLQMC (—●—) for $\lambda_c = 0.1$ (left column), $\lambda_c = 0.3$ (middle column), $\lambda_c = 0.5$ (right column), and $\nu = 0.5$ (top row), $\nu = 1$ (middle row), $\nu = 2$ (bottom row) for the first quantity of interest (point evaluation).

the speedup of MG-MLQMC over MLMC and MG-MLQMC over MLQMC, respectively. From this table, we conclude that the MLQMC algorithm is a very efficient algorithm compared to the basic MLMC method. The method performs better as the covariance function becomes smoother. However, by using an FMG solver and reusing samples on coarser levels, we can gain another factor two in computational work reduction.

To better understand the benefit of the sample recycling, we show the sample distribution across the levels for the finest computed tolerance in the MG-MLQMC method in Figure 8. First, notice that the N_ℓ is a decreasing sequence with ℓ , as required. Secondly, the number of samples decreases faster when moving from top to bottom, indicating that there are fewer samples to be reused on coarser levels. Also indicated on the figure is the percentage of recycled samples, i.e., $N_{\ell+1}/N_\ell \cdot 100\%$. The higher this value, the better the method with sample recycling will perform, compared to standard MLQMC without recycling. These numbers should be compared to the results in Figure 7 and Table 3 (left).

TABLE 3

Speedup of MLQMC over MLMC (S_1) and MG-MLQMC over MLMC (S_2) and MLQMC (S_3) for the first (left) and second (right) quantity of interest.

λ_c	ν	S_1	S_2	S_3	λ_c	ν	S_1	S_2	S_3
0.1	0.5	1.34	2.60	2.04	0.1	0.5	1.18	1.75	1.53
0.3	0.5	1.25	3.16	2.43	0.3	0.5	0.89	1.45	1.66
0.5	0.5	2.85	5.42	1.87	0.5	0.5	1.07	1.49	1.40
0.1	1.0	1.44	3.49	2.35	0.1	1.0	1.25	2.08	1.98
0.3	1.0	3.17	7.82	2.17	0.3	1.0	1.26	2.40	2.00
0.5	1.0	9.68	15.32	1.67	0.5	1.0	1.27	2.26	1.68
0.1	2.0	2.94	4.98	1.93	0.1	2.0	2.14	2.69	1.16
0.3	2.0	9.22	10.57	1.35	0.3	2.0	2.83	3.68	1.32
0.5	2.0	41.93	36.94	0.90	0.5	2.0	4.10	7.25	1.49

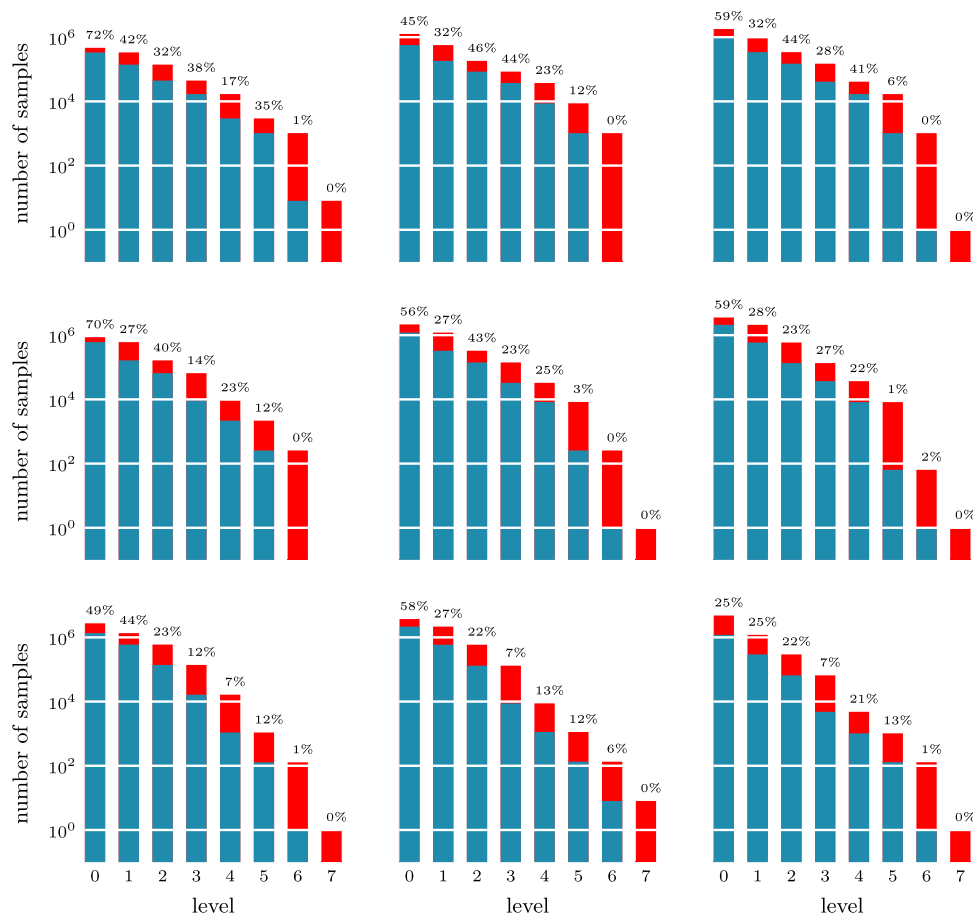


FIG. 8. Number of original (red) and recycled (blue) samples taken on each level in the MG-MLQMC algorithm for $\lambda_c = 0.1$ (left column), $\lambda_c = 0.3$ (middle column), $\lambda_c = 0.5$ (right column), and $\nu = 0.5$ (top row), $\nu = 1$ (middle row), $\nu = 2$ (bottom row) for the first quantity of interest (point evaluation). The numbers indicate the percentage of recycled samples.

Second quantity of interest. Again, we perform simulations for a sequence of decreasing tolerances, now starting from $\epsilon = 0.1$. The results of these simulations,

TABLE 4
Percentage of recycled samples on each level for the second quantity of interest.

λ_c	ν	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$
0.1	0.5	30%	8%	6%	5%	5%	0%
0.3	0.5	23%	13%	3%	5%	5%	0%
0.5	0.5	35%	14%	2%	5%	5%	0%
0.1	1.0	27%	6%	2%	5%	0%	
0.3	1.0	12%	13%	2%	5%	0%	
0.5	1.0	35%	3%	5%	5%	5%	0%
0.1	2.0	21%	11%	3%	0%	0%	
0.3	2.0	19%	8%	5%	1%	0%	
0.5	2.0	9%	5%	6%	14%	0%	

using the more challenging quantity of interest (an average over a subdomain), are summarized in Table 3 (right). We computed the same speedup measures S_1 – S_3 and observe similar behavior as for the first quantity of interest (a point evaluation). In this case, the speedup of MLQMC over MLMC (S_1) is smaller but can be improved by recycling samples (S_2).

Finally, we list the percentage of recycled samples for the second quantity of interest in Table 4, similar to Figure 8. Although the percentage of recycled samples is smaller, the results are in agreement with Proposition 3.1, indicating a larger cost reduction because of sample recycling when the smoothness and correlation length of the random field are small.

7. Conclusions and future work. In this work, we have introduced an algorithm that combines MLMC with a multigrid method, similar to the work in [22]. Our goal was to recycle coarse approximations from the FMG method as samples in the MLMC method, in such a way that no additional error is introduced. We presented a practical algorithm that achieves this goal in section 3.4 and analyzed its cost in section 3.3.

In the numerical experiments with our new method, we confined ourselves to the well-known two-dimensional elliptic PDE with random coefficients, where the uncertain coefficients are modeled as a lognormal random field with a covariance function of Matérn type. We performed various simulations with different combinations of correlation length and smoothness in this covariance function and different quantities of interest. We observed that, as the problem becomes more difficult (i.e., the covariance function becomes less smooth), there is more gain from the sample recycling. Hence, our method works in tandem with the QMC method, which works better as the smoothness parameter in the covariance function increases.

A next step would be to replace the multigrid method by an algebraic multigrid (AMG) method. Contrary to the geometric multigrid method we used in this work, AMG does not require a physical grid to be associated with each level. Instead, coarsening happens fully based on the matrix entries, resulting in a completely autonomous solver.

Another way to combine MLMC with multigrid is to use the multi-index setting from [19], where multiple directions of refinement are allowed. One direction would then correspond to the physical discretization, the other direction would be the number of V-cycles in the multigrid method. This could then be combined with the adaptive approach from [35].

REFERENCES

- [1] K. ATKINSON AND W. HAN, *Numerical solution of Fredholm integral equations of the second kind*, in Theoretical Numerical Analysis, Texts in Applied Mathematics, Vol. 39, Springer, New York, NY, 2009, pp. 473–549, https://doi.org/10.1007/978-1-4419-0458-4_12.
- [2] A. BARTH, C. SCHWAB, AND N. ZOLLINGER, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numerische Mathematik, 119 (2011), pp. 123–161, <https://doi.org/10.1007/s00211-011-0377-0>.
- [3] W. BRIGGS, V. HENSON, AND S. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, PA, 2000, <https://doi.org/10.1137/1.9780898719505>.
- [4] K. A. CLIFFE, M. B. GILES, R. SCHEICHL, AND A. L. TECKENTRUP, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Visualization Sci., 14 (2011), pp. 3–15, <https://doi.org/10.1007/s00791-011-0160-x>.
- [5] G. DETOMMASO, T. DODWELL, AND R. SCHEICHL, *Continuous Level Monte Carlo and Sample-Adaptive Model Hierarchies*, preprint, arXiv:1802.07539, 2018.
- [6] J. DICK, F. Y. KUO, AND I. H. SLOAN, *High-dimensional integration: The quasi-Monte Carlo way*, Acta Numerica, 22 (2013), pp. 133–288, <https://doi.org/10.1017/S0962492913000044>.
- [7] D. DRZISGA, B. GMEINER, U. RÜDE, R. SCHEICHL, AND B. WOHLMUTH, *Scheduling massively parallel multigrid for multilevel Monte Carlo methods*, SIAM J. Sci. Comput., 39 (2017), pp. S873–S897, <https://doi.org/10.1137/16M1083591>.
- [8] M. EIGEL, C. MERDON, AND J. NEUMANN, *An adaptive multilevel Monte Carlo method with stochastic bounds for quantities of interest with uncertain data*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 1219–1245, <https://doi.org/10.1137/15M1016448>.
- [9] D. ELFVBERG, F. HELLMAN, AND A. MÅLQVIST, *A multilevel Monte Carlo method for computing failure probabilities*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 312–330, <https://doi.org/10.1137/140984294>.
- [10] O. G. ERNST, C. E. POWELL, D. J. SILVESTER, AND E. ULLMANN, *Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data*, SIAM J. Sci. Comput., 31 (2009), pp. 1424–1447, <https://doi.org/10.1137/070705817>.
- [11] R. G. GHANEM AND P. D. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, Springer, New York, NY, 1991, <https://doi.org/10.1007/978-1-4612-3094-6>.
- [12] M. B. GILES, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617, <https://doi.org/10.1287/opre.1070.0496>.
- [13] M. B. GILES, *Multilevel Monte Carlo methods*, Acta Numerica, 24 (2015), pp. 259–328, <https://doi.org/10.1017/S096249291500001X>.
- [14] M. B. GILES AND B. J. WATERHOUSE, *Multilevel quasi-Monte Carlo path simulation*, in Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics, 2009, pp. 165–181, <https://doi.org/10.1515/9783110213140.165>.
- [15] I. GRAHAM, F. KUO, D. NUYENS, R. SCHEICHL, AND I. SLOAN, *Analysis of circulant embedding methods for sampling stationary random fields*, SIAM J. Numer. Anal., 56 (2018), pp. 1871–1895, <https://doi.org/10.1137/17M1149730>.
- [16] 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*, J. Comput. Phys., 230 (2011), pp. 3668–3694, <https://doi.org/10.1016/j.jcp.2011.01.023>.
- [17] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, Heidelberg, 1985, <https://doi.org/10.1007/978-3-662-02427-0>.
- [18] W. HACKBUSCH, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1994, <https://doi.org/10.1007/978-3-319-28483-5>.
- [19] A.-L. HAJI-ALI, F. NOBILE, AND R. TEMPONE, *Multi-index Monte Carlo: When sparsity meets sampling*, Numerische Mathematik, 132 (2016), pp. 767–806, <https://doi.org/10.1007/s00211-015-0734-5>.
- [20] A.-L. HAJI-ALI, F. NOBILE, E. VON SCHWERIN, AND R. TEMPONE, *Optimization of mesh hierarchies in multilevel Monte Carlo samplers*, Stoch. Partial Differ. Equ. Anal. Comput., 4 (2016), pp. 76–112, <https://doi.org/10.1007/s40072-015-0049-7>.
- [21] H. HOEL, E. VON SCHWERIN, A. SZEPESSY, AND R. TEMPONE, *Adaptive multilevel Monte Carlo simulation*, in Numerical Analysis of Multiscale Computations, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 217–234, https://doi.org/10.1007/978-3-642-21943-6_10.
- [22] P. KUMAR, C. W. OOSTERLEE, AND R. P. DWIGHT, *A multigrid multilevel Monte Carlo method using high-order finite-volume scheme for lognormal diffusion problems*, Int. J. Uncertain. Quantif., 7 (2017), pp. 57–81, <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2016018677>.
- [23] P. KUMAR, C. RODRIGO, F. J. GASPAR, AND C. W. OOSTERLEE, *On Cell-Centered Multigrid Methods and Local Fourier Analysis for PDEs with Random Coefficients*, preprint, arXiv:1803.08864, 2018.

- [24] F. Y. KUO AND D. NUYENS, *Application of quasi-Monte Carlo Methods to elliptic PDEs with random diffusion coefficients: A survey of analysis and implementation*, Found. Comput. Math., 16 (2016), pp. 1631–1696, <https://doi.org/10.1007/s10208-016-9329-5>.
- [25] F. Y. KUO, R. SCHEICHL, C. SCHWAB, I. H. SLOAN, AND E. ULLMANN, *Multilevel quasi-Monte Carlo methods for lognormal diffusion problems*, Math. Comput., 86 (2017), pp. 2827–2860, <https://doi.org/10.1090/mcom/3207>.
- [26] F. Y. KUO, C. SCHWAB, AND I. H. SLOAN, *quasi-Monte Carlo methods for high-dimensional integration: The standard (weighted Hilbert space) setting and beyond*, ANZIAM J., 53 (2011), pp. 1–37, <https://doi.org/10.1017/S1446181112000077>.
- [27] O. LE MAÎTRE AND O. M. KNIO, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Springer Science & Business Media, 2010, <https://doi.org/10.1007/978-90-481-3520-2>.
- [28] H. NIEDERREITER, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, PA, 1992, <https://doi.org/10.1137/1.9781611970081>.
- [29] D. NUYENS AND R. COOLS, *Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces*, Math. Comput., 75 (2006), pp. 903–920, <https://doi.org/10.1090/s0025-5718-06-01785-6>.
- [30] D. NUYENS AND F. Y. KUO, *QMC4PDE: A Practical Guide to the Software for Constructing Point Sets and Point Generator Code*, 2016, <https://people.cs.kuleuven.be/~dirk.nuyens/qmc4pde/>.
- [31] C.-H. RHEE AND P. W. GLYNN, *Unbiased estimation with square root convergence for SDE models*, Oper. Res., 63 (2015), pp. 1026–1043, <https://doi.org/10.1287/opre.2015.1404>.
- [32] P. ROBBE, *A Practical Multilevel Quasi-Monte Carlo Method for Elliptic PDEs with Random Coefficients*, master thesis, “Een Parallele Multilevel Monte-Carlo-Methode voor de Simulatie van Stochastische Partiële Differentiaalvergelijkingen”, Nuyens, D. and Vandewalle, S. (supervisors), 2016.
- [33] P. ROBBE, *MultilevelEstimators.jl: Multilevel Monte Carlo for Julia*, 2018, <https://github.com/PieterjanRobbe/MultilevelEstimators.jl>.
- [34] P. ROBBE, D. NUYENS, AND S. VANDEWALLE, *A multi-index quasi-Monte Carlo algorithm for lognormal diffusion problems*, SIAM J. Sci. Comput., 39 (2017), pp. S851–S872, <https://doi.org/10.1137/16M1082561>.
- [35] P. ROBBE, D. NUYENS, AND S. VANDEWALLE, *A dimension-adaptive multi-index Monte Carlo method applied to a model of a heat exchanger*, in Monte Carlo and Quasi-Monte Carlo Methods, A. B. Owen and P. W. Glynn, eds., Springer International Publishing, New York, 2018, pp. 429–445.
- [36] J. RUGE AND K. STÜBEN, *Algebraic multigrid methods (AMG)*, in Multigrid Methods, Frontiers in Applied Mathematics 5, SIAM, Philadelphia, 1987, <https://doi.org/10.1137/1.9781611971057.ch4>.
- [37] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, <https://doi.org/10.1137/1.9780898718003>.
- [38] I. H. SLOAN AND S. JOE, *Lattice Methods for Multiple Integration*, Oxford University Press, New York and Oxford, 1994, <https://doi.org/10.1137/1038021>.
- [39] I. SOBOLOV, *On the distribution of points in a cube and the approximate evaluation of integrals*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 86–112, [https://doi.org/10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9).
- [40] U. TROTTEBERG, C. W. OOSTERLEE, AND A. SCHULLER, *Multigrid*, Academic Press Inc., San Diego, 2000.
- [41] H. VAN EMDEN AND U. M. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, [https://doi.org/10.1016/S0168-9274\(01\)00115-5](https://doi.org/10.1016/S0168-9274(01)00115-5).