

A Multilevel Monte Carlo Method for Computing Failure Probabilities*

Daniel Elfverson[†], Fredrik Hellman[‡], and Axel Målqvist[§]

Abstract. We propose and analyze a method for computing failure probabilities of systems modeled as numerical deterministic models (e.g., PDEs) with uncertain input data. A failure occurs when a functional of the solution to the model is below (or above) some critical value. By combining recent results on quantile estimation and the multilevel Monte Carlo method, we develop a method that reduces computational cost without loss of accuracy. We show how the computational cost of the method relates to error tolerance of the failure probability. For a wide and common class of problems, the computational cost is asymptotically proportional to solving a single accurate realization of the numerical model, i.e., independent of the number of samples. Significant reductions in computational cost are also observed in numerical experiments.

Key words. multilevel Monte Carlo, failure probability, uncertainty quantification, error analysis

AMS subject classifications. 65N15, 65C05, 60H35

DOI. 10.1137/140984294

1. Introduction. This paper is concerned with the computational problem of finding the probability for failures of a modeled system. The model input is subject to uncertainty with known distribution, and a failure is the event that a functional (quantity of interest (QoI)) of the model output is below (or above) some critical value. The goal of this paper is to develop an efficient and accurate multilevel Monte Carlo (MLMC) method to find the failure probability. We focus mainly on the case when the model is a partial differential equation (PDE), and we use terminology from the discipline of numerical methods for PDEs. However, the methodology presented here is also applicable in a more general setting.

An MLMC method inherits the nonintrusive and nonparametric characteristics from the standard Monte Carlo (MC) method. This allows the method to be used for complex black-box problems for which intrusive analysis is difficult or impossible. The MLMC method uses a hierarchy of numerical approximations on different accuracy levels. The levels in the hierarchy are typically directly related to a grid size or timestep length. The key idea behind the MLMC method is to use low accuracy solutions as control variates for high accuracy solutions in order to construct an estimator with lower variance. Savings in computational cost are achieved when the low accuracy solutions are cheap and are sufficiently correlated with the high

*Received by the editors August 29, 2014; accepted for publication (in revised form) January 28, 2016; published electronically April 5, 2016.

<http://www.siam.org/journals/juq/4/98429.html>

[†]Department of Mathematics and Mathematical Statistics, Umeå University, SE-901 87 Umeå, Sweden (daniel.elfverson@umu.se). The research of this author was supported by the Göran Gustafsson Foundation.

[‡]Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden (fredrik.hellman@it.uu.se). The research of this author was supported by the Centre for Interdisciplinary Mathematics, Uppsala University.

[§]Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, SE-412 96 Göteborg, Sweden (axel@chalmers.se). The research of this author was supported by the Swedish Research Council.

accuracy solutions. MLMC was first introduced in [10] for stochastic differential equations (SDEs) as a generalization of a two-level variance reduction technique introduced in [17]. The method has been applied to and analyzed for elliptic PDEs in [3, 5, 4, 19]. Further improvements of the MLMC method, such as work on optimal hierarchies, nonuniform meshes, and more accurate error estimates, can be found in [15, 6]. In the present paper, we are not interested in the expected value of the QoI, but instead are interested in a failure probability, which is essentially a single point evaluation of the cumulative distribution function (cdf). For extreme failure probabilities, related methods include importance sampling [14], importance splitting [13], and subset simulations [1]. Works more closely related to the present paper include the results on MLMC methods for computing payoffs of binary options [2] and non-parametric density estimation for PDE models in [9] and, in particular, [8]. In the latter, the selective refinement method for quantiles was formulated and analyzed.

In this paper, we seek to compute the cdf at a given critical value. The cdf at the critical value can be expressed as the expectation value of a Bernoulli distributed random variable Q that is equal to 1 if the QoI is smaller than the critical value, and 0 otherwise. The key idea behind selective refinement is that realizations with QoI far from the critical value can be solved to a lower accuracy than those close to the critical value, and still yield the same value of Q . The random variable Q lacks regularity with respect to the uncertain input data, and hence we are in an unfavorable situation for application of the MLMC method. However, with the computational savings from the selective refinement, it is still possible to obtain an asymptotic result for the computational cost where the cost for the full estimator is proportional to the cost for a single realization to the highest accuracy.

The paper is structured as follows. Section 2 presents the necessary assumptions and the precise problem description. It is followed by section 3, where our particular failure probability functional is defined and analyzed for the MLMC method. In sections 4 and 5 we revisit the MLMC and selective refinement methods adapted to this problem, and in section 6 we show how to combine MLMC with selective refinement to obtain optimal computational cost. In section 7 we give details on how to implement the method in practice. The paper is concluded with two numerical experiments in section 8.

2. Problem formulation. We consider a model problem \mathcal{M} , e.g., a (non)linear differential operator with uncertain data. We let u denote the solution to the model

$$\mathcal{M}(\omega, u) = 0,$$

where the data ω is sampled from a space Ω . In what follows we assume that there exists a unique solution u given any $\omega \in \Omega$ almost surely. It follows that the solution u to a given model problem \mathcal{M} is a random variable which can be parameterized in ω , i.e., $u = u(\omega)$.

The focus of this work is to compute failure probabilities; i.e., we are not interested in some pointwise estimate of the expected value of the solution, $\mathbb{E}[u]$, but rather the probability that a given QoI expressed as a functional, $X(u)$ of the solution u , is less (or greater) than some given critical value y . We let F denote the cdf of the random variable $X = X(\omega)$. The failure probability is then given by

$$(2.1) \quad p = F(y) = \Pr(X \leq y).$$

The following example illustrates how the problem description relates to real world problems.

Example 2.1. As an example, geological sequestration of carbon dioxide (CO₂) is performed by injection of CO₂ in an underground reservoir. The fate of the CO₂ determines the success or failure of the storage system. The CO₂ propagation is often modeled as a PDE with random input data, such as a random permeability field. Typical QoIs include reservoir breakthrough time or pressure at a fault. The value y corresponds to a critical value which the QoI may not exceed or go below. In the breakthrough time case, low values are considered failures. In the pressure case, high values are considered failures. In the latter case, one should negate the QoI to transform the problem into the form of (2.1).

The only regularity assumption on the model is the following Lipschitz continuity assumption of the cdf, which is assumed to hold throughout the paper.

Assumption 2.2. For any $x, y \in \mathbb{R}$,

$$(2.2) \quad |F(x) - F(y)| \leq C_L |x - y|.$$

To compute the failure probability, we consider the Bernoulli distributed variable $Q = \mathbb{1}(X \leq y)$ which takes the value 1 if $X \leq y$ and 0 otherwise. The cdf can be expressed as the expected value of Q , i.e., $p = F(y) = \mathbb{E}[Q]$. In practice we construct an estimator \widehat{Q} for $\mathbb{E}[Q]$, based on approximate sample values from X . As such, \widehat{Q} often suffers from numerical bias from the approximation in the underlying sample. Our goal is to compute the estimator \widehat{Q} to a given root mean square error (RMSE) tolerance ϵ , i.e., to compute

$$e[\widehat{Q}] = \left(\mathbb{E} \left[\left(\widehat{Q} - \mathbb{E}[Q] \right)^2 \right] \right)^{1/2} = \left(\mathbb{V}[\widehat{Q}] + \left(\mathbb{E}[\widehat{Q} - Q] \right)^2 \right)^{1/2} \leq \epsilon$$

to a minimal computational cost. The equality above shows a standard way of splitting the RMSE into a stochastic error and a numerical bias contribution.

The next section presents assumptions and results regarding the numerical discretization of the particular failure probability functional Q .

3. Approximate failure probability functional. We will not consider a particular approximation technique for computing \widehat{Q} , but instead make some abstract assumptions on the underlying discretization. We introduce a hierarchy of refinement levels $\ell = 0, 1, \dots$ and let X'_ℓ and $Q'_\ell = \mathbb{1}(X'_\ell \leq y)$ be an approximate QoI of the model, and we approximate failure probability, respectively, on level ℓ . One possible and natural way to define the accuracy on level ℓ is by assuming

$$(3.1) \quad |X - X'_\ell| \leq \gamma^\ell$$

for some $0 < \gamma < 1$. This means that the error of all realizations on level ℓ is uniformly bounded by γ^ℓ . In a PDE setting, typically an a priori error bound or a posteriori error estimate,

$$|X(\omega) - X_h(\omega)| \leq C(\omega)h^s,$$

can be derived for some constants $C(\omega)$, s and for a discretization parameter h . Then we can choose $X'_\ell = X_h$ with $h = (C(\omega)^{-1}\gamma^\ell)^{1/s}$ to fulfill (3.1). We denote by $\mathcal{C}[\cdot]$ the expected

computational cost to compute a certain quantity. We make the following assumption on the computational cost.

Assumption 3.1. *The expected computational cost for computing one realization of X'_ℓ is*

$$(3.2) \quad \mathcal{C}[X'_\ell] = \gamma^{-q\ell}$$

for some $q > 0$.

Typically finite element methods, finite volume methods, or finite difference schemes are used to compute X'_ℓ . Then q depends on the physical dimension of the computational domain, the convergence rate of the solution method, and the computational complexity for assembling and solving the linear system. Note that the unit of work is normalized according to (3.2).

For an accurate value of the failure probability functional, the condition in (3.1) is unnecessarily strong. This functional is very sensitive to perturbations of values close to y , but insensitive to perturbations for values far from y . This insensitivity can be exploited. We introduce a different approximation X_ℓ and impose the following, relaxed, assumption on this approximation of X , which allows for larger errors far from the critical value y . This assumption is illustrated in Figure 1.

Assumption 3.2. *The numerical approximation X_ℓ of X satisfies*

$$(3.3) \quad |X - X_\ell| \leq \gamma^\ell \quad \text{or} \quad |X - X_\ell| < |X_\ell - y|$$

for a fixed $0 < \gamma < 1$.

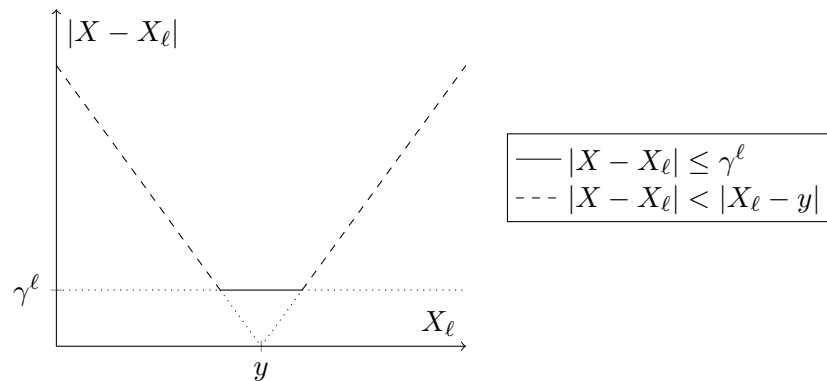


Figure 1. *Illustration of condition (3.3). The numerical error is allowed to be larger than γ^ℓ far away from y .*

We define $Q_\ell = \mathbb{1}(X_\ell \leq y)$ analogously to Q'_ℓ . Let us compare the implications of the two conditions (3.1) and (3.3) on the quality of the two respective approximations. Denote by X'_ℓ and Q'_ℓ stochastic variables obeying the error bound (3.1) and its corresponding approximate failure functional, respectively, and let X_ℓ obey (3.3). In a practical situation, Assumption 3.2 is fulfilled by iterative refinements of X_ℓ until condition (3.3) is satisfied. It is natural to use a similar procedure to achieve the stricter condition (3.1) for X'_ℓ . We express this latter assumption of using similar procedures for computing X_ℓ and X'_ℓ as

$$(3.4) \quad |X - X_\ell| \leq \gamma^\ell \text{ implies } X'_\ell = X_\ell;$$

i.e., for outcomes where X_ℓ is solved to accuracy γ^ℓ , X'_ℓ is equal to X_ℓ . Under that assumption, the following lemma shows that of the two approximations Q'_ℓ and Q_ℓ , Q'_ℓ is the most accurate.

Lemma 3.3. *Let X'_ℓ and X_ℓ fulfill (3.1) and (3.3), respectively, and assume (3.4) holds. Then $\Pr(Q_\ell = Q) \geq \Pr(Q'_\ell = Q)$.*

Proof. We split Ω into the events $A = \{\omega \in \Omega : |X - X_\ell| \leq \gamma^\ell\}$ and its complement $\Omega \setminus A$. For $\omega \in A$, using (3.4), we conclude that $Q'_\ell = Q_\ell$, and hence

$$\Pr(Q_\ell = Q \mid A) = \Pr(Q'_\ell = Q \mid A).$$

For $\omega \notin A$, we have $|X - X_\ell| > \gamma^\ell$, and from (3.3) we have that $|X - X_\ell| < |X_\ell - y|$. We now want to show that this implies $Q_\ell = Q$ or, equivalently, $X_\ell \leq y \Leftrightarrow X \leq y$. For the direction \Rightarrow ,

$$(3.5) \quad 0 \leq |X_\ell - y| = y - X_\ell \leq y - X + |X - X_\ell| \leq y - X + |X_\ell - y|,$$

i.e., $X \leq y$. The procedure is similar for \Leftarrow . Hence,

$$\Pr(Q_\ell = Q \mid \Omega \setminus A) = 1.$$

Since $\Pr(Q'_\ell = Q \mid \Omega \setminus A) \leq 1$, we get $\Pr(Q_\ell = Q) \geq \Pr(Q'_\ell = Q)$. ■

Under Assumption 3.2 we can prove the following lemma on the accuracy of the failure probability function Q_ℓ .

Lemma 3.4. *Under Assumptions 2.2 and 3.2, the following statements are satisfied, where C_1 and C_2 do not depend on ℓ :*

- M1. $|\mathbb{E}[Q_\ell - Q]| \leq C_1 \gamma^\ell$,
- M2. $\mathbb{V}[Q_\ell - Q_{\ell-1}] \leq C_2 \gamma^\ell$ for $\ell \geq 1$.

Proof. We split Ω into the events $B = \{\omega \in \Omega : \gamma^\ell \geq |X_\ell - y|\}$ and its complement $\Omega \setminus B$. In $\Omega \setminus B$, we have $Q_\ell = Q$, since $|X - X_\ell| < |X_\ell - y|$ from (3.3). See (3.5) in the previous proof. Also, we note that the event B implies $|X - X_\ell| \leq \gamma^\ell$, and hence $|X - y| \leq 2\gamma^\ell$. Then

$$\begin{aligned} |\mathbb{E}[Q_\ell - Q]| &= \left| \int_B Q_\ell(\omega) - Q(\omega) \, dP(\omega) \right| \leq \int_B 1 \, dP(\omega) \\ &\leq \Pr(|X - y| \leq 2\gamma^\ell) = F(y - 2\gamma^\ell) - F(y + 2\gamma^\ell) \\ &\leq 4C_L \gamma^\ell, \end{aligned}$$

which proves M1. Statement M2 follows directly from M1, since

$$\begin{aligned} \mathbb{V}[Q_\ell - Q_{\ell-1}] &= \mathbb{E}[(Q_\ell - Q_{\ell-1})^2] - \mathbb{E}[Q_\ell - Q_{\ell-1}]^2 \\ &\leq \mathbb{E}[Q_\ell - 2Q_\ell Q_{\ell-1} + Q_{\ell-1}] \\ &\leq |\mathbb{E}[Q_\ell - Q]| + |2\mathbb{E}[Q_\ell Q_{\ell-1} - Q]| + |\mathbb{E}[Q_{\ell-1} - Q]| \\ &\leq 2|\mathbb{E}[Q_\ell - Q]| + 2|\mathbb{E}[Q_{\ell-1} - Q]| \\ &\leq C_2 \gamma^\ell, \end{aligned}$$

where $(Q_\ell)^2 = Q_\ell$ was used. ■

It is interesting to note that with this particular failure probability functional, the convergence rate in M2 cannot be improved if the rate in M1 is already sharp, as the following lemma shows.

Lemma 3.5. *Let $0 < \gamma < 1$ be fixed. If there is a $0 < c \leq C_1$ such that the failure probability functional satisfies*

$$c\gamma^\ell \leq |\mathbb{E}[Q_\ell - Q]| \leq C_1\gamma^\ell$$

for all $\ell = 0, 1, \dots$, then

$$\mathbb{V}[Q_\ell - Q_{\ell-1}] \leq C_2\gamma^{\beta\ell},$$

where $\beta = 1$ is sharp in the sense that the relation will be violated for sufficiently large ℓ if $\beta > 1$.

Proof. Assume that $\mathbb{V}[Q_\ell - Q_{\ell-1}] \leq C\gamma^{\beta\ell}$ for some constant C and $\beta > 1$. For two levels $k < \ell$ such that $c\gamma^k > C_1\gamma^\ell$ we have that

$$|\mathbb{E}[Q_\ell - Q_k]| \geq ||\mathbb{E}[Q_\ell - Q]| - |\mathbb{E}[Q_k - Q]| \geq (c - C_1\gamma^{\ell-k})\gamma^k = \tilde{c}\gamma^k,$$

with $\tilde{c} = c - C_1\gamma^{\ell-k} > 0$. For such ℓ and k , we have

$$\begin{aligned} \tilde{c}\gamma^k &\leq |\mathbb{E}[Q_\ell - Q_k]| \leq \sum_{j=k}^{\ell-1} |\mathbb{E}[Q_{j+1} - Q_j]| \leq \sum_{j=k}^{\ell-1} \mathbb{E}[(Q_{j+1} - Q_j)^2] \\ &= \sum_{j=k}^{\ell-1} \left(\mathbb{V}[Q_{j+1} - Q_j] + (\mathbb{E}[Q_{j+1} - Q_j])^2 \right) \\ &\leq \sum_{j=k}^{\ell-1} \left(C\gamma^{\beta j} + \mathcal{O}(\gamma^{2j}) \right) \leq \tilde{C}\gamma^{\beta k} + \mathcal{O}(\gamma^{2k}). \end{aligned}$$

For $\ell, k \rightarrow \infty$ (keeping $\ell - k$ constant) we have a contradiction due to the mismatching rates, and hence $\beta \leq 1$, which proves that the bound cannot be improved. ■

4. Multilevel Monte Carlo method. In this section, we present the MLMC method in a general context. Because of the low convergence rate of the variance in M2, the MLMC method does not perform optimally for the failure probability functional. The results presented here will be combined with the results from section 5 to derive a new method to compute failure probabilities efficiently in section 6.

The (standard) MC estimator at refinement level ℓ of $\mathbb{E}[Q]$ using a sample $\{\omega_\ell^i\}_{i=1}^{N_\ell}$ reads as

$$\widehat{Q}_{N_\ell, \ell}^{MC} = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Q_\ell(\omega_\ell^i).$$

Note that the subscripts N_ℓ and ℓ control the statistical error and numerical bias, respectively. The expected value and variance of the estimator $\widehat{Q}_{N_\ell, \ell}^{MC}$ are $\mathbb{E}[\widehat{Q}_{N_\ell, \ell}^{MC}] = \mathbb{E}[Q_\ell]$ and $\mathbb{V}[\widehat{Q}_{N_\ell, \ell}^{MC}] = N_\ell^{-1}\mathbb{V}[Q_\ell]$, respectively. Referring to the goal of the paper, we want the MSE (square of the RMSE) to satisfy

$$e \left[\widehat{Q}_{N_\ell, \ell}^{MC} \right]^2 = N_\ell^{-1}\mathbb{V}[Q_\ell] + (\mathbb{E}[Q_\ell - Q])^2 \leq \epsilon^2/2 + \epsilon^2/2 = \epsilon^2;$$

i.e., both the statistical error and the numerical error should be less than $\epsilon^2/2$. The MLMC method is a variance reduction technique for the MC method. The MLMC estimator $\widehat{Q}_{\{N_\ell\},L}^{ML}$ at refinement level L is expressed as a sum of MC estimators $\widehat{Y}_{N_\ell,\ell}^{MC}$ of the correctors $Y_\ell = Q_\ell - Q_{\ell-1}$ for $\ell = 1, \dots, L$ and $Y_0 = Q_0$ for $\ell = 0$. The estimator reads as

$$\widehat{Q}_{\{N_\ell\},L}^{ML} = \sum_{\ell=0}^L \widehat{Y}_{N_\ell,\ell}^{MC} = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Y_\ell(\omega_\ell^i) = \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_\ell - Q_{\ell-1})(\omega_\ell^i),$$

where $Q_{-1} = 0$. There is one corrector for every refinement level $\ell = 0, \dots, L$, each with a specific MC estimator sample size N_ℓ . The expected value and variance of the MLMC estimator are

$$(4.1) \quad \begin{aligned} \mathbb{E} \left[\widehat{Q}_{\{N_\ell\},L}^{ML} \right] &= \sum_{\ell=0}^L \mathbb{E}[Q_\ell - Q_{\ell-1}] = \mathbb{E}[Q_L] \quad \text{and} \\ \mathbb{V} \left[\widehat{Q}_{\{N_\ell\},L}^{ML} \right] &= \sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Q_\ell - Q_{\ell-1}], \end{aligned}$$

respectively. Using (4.1) the MSE for the MLMC estimator can be expressed as

$$e \left[\widehat{Q}_{\{N_\ell\},L}^{ML} \right]^2 = \sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Q_\ell - Q_{\ell-1}] + (\mathbb{E}[Q_L - Q])^2,$$

and can be computed at expected cost

$$\mathcal{C} \left[\widehat{Q}_{\{N_\ell\},L}^{ML} \right] = \sum_{\ell=0}^L N_\ell c_\ell,$$

where $c_\ell = \mathcal{C}[Q_\ell] + \mathcal{C}[Q_{\ell-1}]$. Given that the variance of the MLMC estimator is $\epsilon^2/2$, the expected cost is minimized by choosing

$$(4.2) \quad N_\ell = 2\epsilon^{-2} \sqrt{\mathbb{V}[Q_\ell - Q_{\ell-1}]/c_\ell} \sum_{k=0}^L \sqrt{\mathbb{V}[Q_k - Q_{k-1}]c_k}$$

(see Appendix A), and hence the total expected cost is

$$(4.3) \quad \mathcal{C} \left[\widehat{Q}_{\{N_\ell\},L}^{ML} \right] = 2\epsilon^{-2} \left(\sum_{\ell=0}^L \sqrt{\mathbb{V}[Q_\ell - Q_{\ell-1}]c_\ell} \right)^2.$$

In the analysis, we assume N_ℓ can attain noninteger values. In practice, however, we use sample size $\lceil N_\ell \rceil$. If the product $\mathbb{V}[Q_\ell - Q_{\ell-1}]c_\ell$ increases (or decreases) with ℓ , then the dominating term in (4.3) will be $\ell = L$ (or $\ell = 0$). The values N_ℓ can be estimated on the fly in the MLMC algorithm using (4.2), while the cost c_ℓ can be estimated using an a priori model. The computational complexity to obtain an RMSE less than ϵ of the MLMC estimator for the

failure probability functional is given by the theorem below. In the following, the notation $a \lesssim b$ stands for $a \leq Cb$ with some constant C independent of ϵ and ℓ .

Theorem 4.1. *Let Assumptions 2.2 and 3.2 hold (so that Lemma 3.4 holds), and let $\mathcal{C}[Q_\ell] \lesssim \gamma^{-r\ell}$. Then there exists a constant L and a sequence $\{N_\ell\}$ such that the RMSE is less than ϵ , and the expected cost of the MLMC estimator is*

$$(4.4) \quad \mathcal{C}\left[\widehat{Q}_{\{N_\ell\},L}^{ML}\right] \lesssim \begin{cases} \epsilon^{-2}, & r < 1, \\ \epsilon^{-2}(\log \epsilon^{-1})^2, & r = 1, \\ \epsilon^{-1-r}, & r > 1. \end{cases}$$

Proof. For a proof see, e.g., [5, 10]. ■

The most straightforward procedure to fulfill Assumption 3.2 in practice is to refine all realizations on level ℓ uniformly to an error tolerance γ^ℓ , i.e., to compute X'_ℓ introduced in section 3, for which $|X - X'_\ell| \leq \gamma^\ell$. Computing Q'_ℓ using this procedure has the same cost as computing X'_ℓ . The cost rate from Assumption 3.1 directly applies to Q'_ℓ , and we have

$$(4.5) \quad \mathcal{C}[Q'_\ell] = \gamma^{-q\ell}.$$

Using Theorem 4.1, with Q'_ℓ instead of Q_ℓ (which is possible, since Q'_ℓ trivially fulfills Assumption 3.2) we obtain an RMSE of the expected cost less than $\epsilon^{-1-q} = \epsilon^{-1}\mathcal{C}[Q'_\ell]$ for the case $q > 1$.

In the next section we describe how the selective refinement algorithm computes X_ℓ (hence Q_ℓ) that fulfills Assumption 3.2 to a lower cost than its fully refined equivalent X'_ℓ . The theorem above can then be applied with $r = q - 1$ instead of $r = q$.

5. Selective refinement algorithm. In this section we modify the selective refinement algorithm proposed in [8] for computing failure probabilities (instead of quantiles) and for quantifying the error using the RMSE. The selective refinement algorithm computes X_ℓ so that

$$|X - X_\ell| \leq \gamma^\ell \quad \text{or} \quad |X - X_\ell| < |X_\ell - y|$$

in Assumption 3.2 is fulfilled without requiring the stronger (full refinement) condition

$$|X - X_\ell| \leq \gamma^\ell.$$

In contrast to the selective refinement algorithm in [8], Assumption 3.2 can be fulfilled by iterative refinement of realizations over all realizations independently. This allows for an efficient totally parallel implementation. We are particularly interested in quantifying the expected cost required by the selective refinement algorithm, and showing that the X_ℓ resulting from the algorithm fulfills Assumption 3.2.

Algorithm 1 exploits the fact that $Q_\ell = Q$ for realizations satisfying $|X - X_\ell| < |X_\ell - y|$. That is, even if the error of X_ℓ is greater than γ^ℓ , it might be sufficiently accurate to yield the correct value of Q_ℓ . The algorithm works on a per-realization basis, starting with an error tolerance 1. The realization is refined iteratively until Assumption 3.2 is fulfilled. The advantage is that many realizations are solved only with low accuracy, and hence the average

Algorithm 1 Selective refinement algorithm.

-
- 1: Input arguments: level ℓ , realization i , critical value y , and tolerance factor γ
 - 2: Compute $X'_0(\omega_\ell^i)$
 - 3: Let $j = 0$
 - 4: **while** $j < \ell$ and $\gamma^j \geq |X'_j(\omega_\ell^i) - y|$ **do**
 - 5: Let $j = j + 1$
 - 6: Compute $X'_j(\omega_\ell^i)$
 - 7: **end while**
 - 8: Let $X_\ell(\omega_\ell^i) = X'_j(\omega_\ell^i)$
-

cost per Q_ℓ is reduced. Lemma 5.1 shows that X_ℓ computed using Algorithm 1 satisfies Assumption 3.2.

Lemma 5.1. *Approximations X_ℓ computed using Algorithm 1 satisfy Assumption 3.2.*

Proof. At each iteration in the while-loop of Algorithm 1, γ^j is the error tolerance of $X_\ell(\omega_\ell^i)$, i.e., $|X(\omega_\ell^i) - X_\ell(\omega_\ell^i)| \leq \gamma^j$. The stopping criterion hence implies Assumption 3.2 for $X_\ell(\omega_\ell^i)$. ■

The expected cost for computing Q_ℓ using Algorithm 1 is given by the following lemma.

Lemma 5.2. *Under Assumption 3.1 the expected cost to compute the failure probability functional using Algorithm 1 can be bounded as*

$$\mathcal{C}[Q_\ell] \lesssim \sum_{j=0}^{\ell} \gamma^{(1-q)j}.$$

Proof. Consider iteration j , i.e., when $X_\ell(\omega_\ell^i)$ has been computed to tolerance γ^{j-1} . We denote by E_j the probability that a realization enters iteration j . For $j \leq \ell$,

$$\begin{aligned} \Pr(E_j) &= \Pr(y - \gamma^{j-1} \leq X'_{j-1} \leq y + \gamma^{j-1}) \\ &\leq \Pr(y - 2\gamma^{j-1} \leq X \leq y + 2\gamma^{j-1}) \\ &= F(y + 2\gamma^{j-1}) - F(y - 2\gamma^{j-1}) \\ &\leq 4C_L \gamma^{j-1}. \end{aligned}$$

Every realization is initially solved to tolerance 1. Using that the cost for solving a realization to tolerance γ^j is γ^{-qj} from Assumption 3.1, we get that the expected cost is

$$\mathcal{C}[Q_\ell] = 1 + \sum_{j=1}^{\ell} \Pr(E_j) \gamma^{-qj} \leq 1 + \sum_{j=1}^{\ell} 4C_L \gamma^{j-1} \gamma^{-qj} \lesssim \sum_{j=0}^{\ell} \gamma^{(1-q)j},$$

which concludes the proof. ■

6. Multilevel Monte Carlo using the selective refinement strategy. Combining the MLMC method with the algorithm for selective refinement, there can be further savings in computational cost. We call this method multilevel Monte Carlo with selective refinement

(MLMC-SR). In particular, for $q > 1$ we obtain from Lemma 5.2 that the expected cost for one sample can be bounded as

$$(6.1) \quad \mathcal{C}[Q_\ell] \lesssim \sum_{j=0}^{\ell} \gamma^{(1-q)j} \lesssim \gamma^{(1-q)\ell}.$$

Applying Theorem 4.1 with $r = q - 1$ yields the following result.

Theorem 6.1. *Let Assumptions 2.2, 3.1, and 3.2 hold. Further, suppose that Algorithm 1 is executed to compute Q_ℓ and that $\mathcal{C}[Q'_\ell] \lesssim \gamma^{-q\ell}$. Then there exists a constant L and a sequence $\{N_\ell\}$ such that the RMSE is less than ϵ , and the expected cost for the MLMC estimator with selective refinement is*

$$(6.2) \quad \mathcal{C}\left[\widehat{Q}_{\{N_\ell\},L}^{ML}\right] \lesssim \begin{cases} \epsilon^{-2}, & q < 2, \\ \epsilon^{-2}(\log \epsilon^{-1})^2, & q = 2, \\ \epsilon^{-q}, & q > 2. \end{cases}$$

Proof. For $q > 1$, using Lemmas 3.4 and 5.2, the result follows directly from Theorem 4.1 with $r = q - 1$. For $q \leq 1$, we use the rate ϵ^{-2} from the case $1 < q < 2$, since the cost cannot be worsened by making each sample cheaper to compute. ■

In a standard MC method we have $\epsilon^{-2} \sim N$, where N is the number of samples, and $\epsilon^{-q} \sim \mathcal{C}[Q'_L]$, where $\mathcal{C}[Q'_L]$ is the expected computational cost for solving one realization on the finest level without selective refinement. The MLMC-SR method then has the following cost:

$$(6.3) \quad \mathcal{C}\left[\widehat{Q}_{\{N_\ell\},L}^{ML}\right] \lesssim \begin{cases} N, & q < 2, \\ \mathcal{C}[Q'_L], & q > 2. \end{cases}$$

A comparison of MC, MLMC with full refinement (MLMC), and MLMC with selective refinement (MLMC-SR) is given in Table 1. To summarize, the best possible scenario is when the cost is ϵ^{-2} , which is equivalent to a standard MC method, where all samples can be obtained with cost 1. This complexity is obtained for the MLMC method when $q < 1$ and for the MLMC-SR method when $q < 2$. For $q > 2$, the MC method has the same complexity as solving N problems on the finest level $N\mathcal{C}[Q'_L]$, MLMC has the same complexity as solving $N^{1/2}$ problems on the finest level $N^{1/2}\mathcal{C}[Q'_L]$, and the MLMC-SR method has the same complexity as solving one problem on the finest level $\mathcal{C}[Q'_L]$.

Table 1

Comparison of work between MC, MLMC with full refinement (MLMC), and MLMC with selective refinement (MLMC-SR) for different q .

Method	$0 \leq q < 1$	$1 < q < 2$	$q > 2$
MC	ϵ^{-2-q}	ϵ^{-2-q}	ϵ^{-2-q}
MLMC	ϵ^{-2}	ϵ^{-1-q}	ϵ^{-1-q}
MLMC-SR	ϵ^{-2}	ϵ^{-2}	ϵ^{-q}

7. Heuristic algorithm. In this section, we present a heuristic algorithm for MLMC-SR. Contrary to Theorem 6.1, this algorithm does not guarantee that the RMSE is $\mathcal{O}(\epsilon)$, since in practice we lack a priori knowledge of the constants C_1 and C_2 in Lemma 3.4. Instead, the RMSE needs to be estimated. Recall the split of the MSE into a numerical and a statistical contribution:

$$(7.1) \quad \left(\mathbb{E}[Q - \widehat{Q}]\right)^2 \leq \frac{1}{2}\epsilon^2 \quad \text{and} \quad \mathbb{V}[\widehat{Q}] \leq \frac{1}{2}\epsilon^2.$$

With \widehat{Q} being the multilevel MC estimator $\widehat{Q}_{\{N_\ell\},L}^{ML}$, here we present heuristics for estimating the numerical and the statistical error of the estimator.

For both estimates and $\ell \geq 1$, we make use of the discrete variable $Y_\ell(\omega) = (Q_\ell - Q_{\ell-1})(\omega)$. We denote the probability of Y_ℓ being -1 , 0 , and 1 by p_{-1} , p_0 , and p_1 , respectively. For convenience, we drop the index ℓ for the probabilities; however, they do depend on ℓ . In order to estimate the numerical bias $\mathbb{E}[Q - \widehat{Q}_{\{N_\ell\},L}^{ML}] = \mathbb{E}[Q - Q_L]$, we assume that M1 holds approximately with equality, i.e., $|\mathbb{E}[Q - Q_\ell]| \approx C_1\gamma^\ell$. Then the numerical bias can be overestimated, $|\mathbb{E}[Q - Q_\ell]| \leq |\mathbb{E}[Y_\ell]|(\gamma^{-1} - 1)^{-1}$, since

$$\begin{aligned} |\mathbb{E}[Y_\ell]| &= |\mathbb{E}[Q_\ell - Q] - \mathbb{E}[Q_{\ell-1} - Q]| \\ &\geq ||\mathbb{E}[Q_\ell - Q]| - |\mathbb{E}[Q_{\ell-1} - Q]|| \\ &\approx \left| C_1\gamma^\ell - C_1\gamma^{\ell-1} \right| \\ &= C_1\gamma^\ell(\gamma^{-1} - 1). \end{aligned}$$

Hence, we concentrate our effort on estimating $|\mathbb{E}[Y_\ell]|$.

It has been observed that the accuracy of sample estimates of mean and variance of Y_ℓ might deteriorate for deep levels $\ell \gg 1$, and a continuation multilevel MC method was proposed in [6] as a remedy for this. That idea could be applied and specialized for this functional to obtain more accurate estimates. However, in this work we use the properties of Y_ℓ to construct a method with optimal asymptotic behavior, possibly with an increase of computational cost by a constant.

We consider the three Bernoulli distributions $[Y_\ell = 1]$, $[Y_\ell = -1]$, and $[Y_\ell \neq 0]$ which have parameters p_1 , p_{-1} , and $p_1 + p_{-1}$, respectively ($[\cdot]$ is the Iverson bracket notation). These parameters can be used in estimates for both the expectation value and the variance of the trinomially distributed Y_ℓ . Considering a general Bernoulli distribution with parameter p , we want to estimate p . For our distributions, as the level ℓ increases, p approaches zero, which is why we are concerned with finding stable estimates for small p . It is important that the parameter is not underestimated, since it is used to control the numerical bias and statistical error and could then cause premature termination. We propose an estimation method that is easy to implement, and that will overestimate the parameter in case of accuracy problems, rather than underestimate it, while keeping the asymptotic rates given in Lemma 3.4 for the estimators.

The standard unbiased estimator of p is $\hat{p} = xn^{-1}$, where x is the number of observed successes. The proposed alternative (and biased) estimator is $\tilde{p} = (x + k)(n + k)^{-1}$ for a

$k > 0$. This corresponds to a Bayesian estimate with prior beta distribution with parameters $(k + 1, 1)$. Observing that

$$(7.2) \quad \begin{aligned} |\mathbb{E}[Y_\ell]| &= |p_1 - p_{-1}|, \\ \mathbb{V}[Y_\ell] &= p_1 + p_{-1} - (p_1 - p_{-1})^2 \end{aligned}$$

and considering Lemma 3.4 (assuming equality with the rates), we conclude that all three parameters $p \propto \gamma^\ell$ (where \propto means “asymptotically proportional to” for $\ell \gg 1$). With the standard estimator \hat{p} , the relative variance can be expressed as $\mathbb{V}[\hat{p}](\mathbb{E}[\hat{p}])^{-2}$. This quantity should be less than 1 for an accurate estimate. We now examine its asymptotic behavior. The parameter n is the optimal number of samples at level ℓ (equation (4.2)) and can be expressed as

$$(7.3) \quad n \propto \gamma^{\frac{1}{2}\ell q - \frac{1}{2}L(2+q)},$$

where we used that $\epsilon \propto \gamma^L$, $\mathcal{C}[Y_\ell] \propto \gamma^{(1-q)\ell}$, and $\mathbb{V}[Y_\ell] \propto \gamma^\ell$. Then we have

$$\frac{\mathbb{V}[\hat{p}]}{\mathbb{E}[\hat{p}]^2} = \frac{n^{-1}p(1-p)}{p^2} = \frac{1-p}{np} \propto \gamma^{\frac{2+q}{2}(L-\ell)}.$$

In particular, for $\ell = L$, the relative variance is asymptotically constant, but we do not know a priori how big this constant is. When it is large (greater than 1), the relative variance of \hat{p} might be very large. An analogous analysis on \tilde{p} yields

$$(7.4) \quad \frac{\mathbb{V}[\tilde{p}]}{\mathbb{E}[\tilde{p}]^2} = \frac{(n+k)^{-2}np(1-p)}{(n+k)^{-2}(np+k)^2} = \frac{np(1-p)}{(np+k)^2} \leq \frac{np}{(np+k)^2}.$$

Maximizing the bound in (7.4) with respect to np gives

$$\frac{\mathbb{V}[\tilde{p}]}{\mathbb{E}[\tilde{p}]^2} \leq \frac{1}{4k}.$$

Choosing, for instance, $k = 1$ gives a maximum relative variance of $1/4$. Choosing a larger k gives larger bias but smaller relative variance. The bias of this estimator is significant if $np \ll k$; however, that is the case when we have too few samples to estimate the parameter accurately, and then \tilde{p} instead acts as a bound. The estimate \tilde{p} keeps the asymptotic behavior $\mathbb{E}[\tilde{p}] \propto \gamma^\ell$, since

$$\begin{aligned} \mathbb{E}[\tilde{p}] &= \frac{np+k}{n+k} \propto \frac{np+k}{n} = p + \frac{k}{n} \\ &\propto \gamma^\ell + \gamma^{-\frac{1}{2}\ell q + \frac{1}{2}L(2+q)} = \gamma^\ell(1 + \gamma^{\frac{1}{2}(L-\ell)(2+q)}) \leq 2\gamma^\ell \propto p, \end{aligned}$$

where we used that $\ell < L$ and k is constant.

Now, estimating the parameters p_1 , p_{-1} , and $p_1 + p_{-1}$ as \tilde{p}_1 , \tilde{p}_{-1} , and $\tilde{p}_{\pm 1}$, respectively, using the estimator \tilde{p} above (note that the sum $p_1 + p_{-1}$ is estimated separately from p_1 and p_{-1}) we can bound (approximately) the expected value and variance of Y_ℓ in (7.2):

$$(7.5) \quad |\mathbb{E}[Y_\ell]| \leq \max(p_1, p_{-1}) \approx \max(\tilde{p}_1, \tilde{p}_{-1})$$

and

$$(7.6) \quad \mathbb{V}[Y_\ell] \leq p_1 + p_{-1} \approx \tilde{p}_{\pm 1}$$

for $\ell \geq 1$. For $\ell = 0$, the sample size is usually large enough to use the sample mean and variance as accurate estimates. Since the asymptotic behavior of \tilde{p} is γ^ℓ , the rates in Lemma 3.4 still hold and Theorem 6.1 applies (however, with approximate quantities).

In practice, the optimal number of levels L and sample sizes N_ℓ in $\hat{Q}_{\{N_\ell\},L}^{ML}$ are not known. We present Algorithm 2, which is similar to the algorithm in [10], for estimating these quantities and computing the estimate. The difference between standard MLMC and MLMC-SR is that, in the latter, the corrector estimates $\hat{Y}_{N_\ell,\ell}^{MC}$ are computed using the selective refinement algorithm. According to Lemma 3.3 and Theorem 6.1, this gives an estimate of the same accuracy but to a lower cost. The termination criterion is the same as that used in the standard MLMC algorithm [10], i.e.,

$$(7.7) \quad \max(\gamma|\mathbb{E}[Y_{L-1}]|, |\mathbb{E}[Y_L]|) < \frac{1}{\sqrt{2}}(\gamma^{-1} - 1)\epsilon,$$

where $|\mathbb{E}[Y_{L-1}]|$ and $|\mathbb{E}[Y_L]|$ are estimated using the methods presented above. A difference from the standard MLMC algorithm is that the initial sample size for level L is $N_L = N\gamma^{-L}$ instead of $N_L = N$ for some N . This is what is predicted by (7.3) and is necessary to provide accurate estimates of the expectation value and variance of Y_ℓ for deep levels. Other differences from the standard MLMC algorithm are that the selective refinement algorithm (Algorithm 1) is used to compute $\hat{Q}_{N_\ell,L}^{MC}$, and that the estimates of expectation value and variance of Y_ℓ are computed according to the discussion above.

Algorithm 2 MLMC method using selective refinement.

- 1: Pick critical value y , cost model parameter q , tolerance factor γ , initial number of samples N , parameter k , and final tolerance ϵ
 - 2: Set $L = 0$
 - 3: **loop**
 - 4: Let $N_L = N\gamma^{-L}$ and compute $\hat{Y}_{N_\ell,L}^{MC}$ using selective refinement (Algorithm 1)
 - 5: Estimate $\mathbb{V}[Q_\ell - Q_{\ell-1}]$ using (7.6)
 - 6: Compute the optimal $\{N_\ell\}_{\ell=0}^L$ using (4.2) and cost model (6.1)
 - 7: Compute $\hat{Y}_{N_\ell,\ell}^{MC}$ for all levels $\ell = 0, \dots, L$ using selective refinement (Algorithm 1)
 - 8: Estimate $\mathbb{E}[Q_\ell - Q_{\ell-1}]$ using (7.5)
 - 9: Terminate if converged by checking inequality (7.7)
 - 10: Set $L = L + 1$
 - 11: **end loop**
 - 12: The MLMC-SR estimator is $\hat{Q}_{\{N_\ell\},L}^{ML} = \sum_{\ell=0}^L \hat{Y}_{N_\ell,\ell}^{MC}$
-

8. Numerical experiments. Two types of numerical experiments are presented in this section. The first experiment (in section 8.1) is performed on a simple and cheap model \mathcal{M} so that the asymptotic results of the computational cost, derived in Theorem 6.1, can be

verified. The second experiment (in section 8.2) is performed on a PDE model \mathcal{M} to show the method's applicability to realistic problems. In our experiments we made use of the software FEniCS [18] and SciPy [16].

8.1. Failure probability of a normal distribution. In this first demonstrational experiment, we let the QoI X belong to the standard normal distribution, and we seek to find the probability of $X \leq y = 0.8$. The true value of this probability is $\Pr(X \leq 0.8) = \Phi(0.8) \approx 0.78814$, and hence we have a reliable reference solution. We define approximations X_h of X as follows. First, we let our input data ω belong to the standard normal distribution, and let $X(\omega) = \omega$. Then, we let $X_h(\omega) = \omega + h(2U(\omega, h) - 1 + b)/(1 + b)$, where $b = 0.1$ and $U(\omega, h)$ is a uniformly distributed random number between 0 and 1. Since we have an error bound $|X_h - X| \leq h$, the selective refinement algorithm (Algorithm 1) can be used to construct a function X_ℓ satisfying Assumption 3.2. With this setup it is very cheap to compute X_h to any accuracy h ; however, for illustrative purposes we assume a cost model $\mathcal{C}[X_h] = h^{-q}$ with $q = 1, 2, 3$ to cover the three cases in Theorem 6.1.

For the three values of q , and eight logarithmically distributed values of ϵ between 10^{-3} and 10^{-1} , we performed 100 runs of Algorithm 2. All parameters used in the simulations are presented in Table 2.

Table 2

Parameters used for the demonstrational experiment.

Parameter	Value
y	0.8
q	1, 2, 3
γ	0.5
N	10
k	1
ϵ	$(10^{-3}, 10^{-1})$

For convenience, we denote by \widehat{Q}_i the MLMC-SR estimator $\widehat{Q}_{\{N_\ell\}, L}^{ML}$ of the failure probability from run $i = 1, \dots, M$ with $M = 100$. For each tolerance ϵ and cost parameter q , we estimated the RMSE of the MLMC-SR estimator by

$$e[\widehat{Q}_{\{N_\ell\}, L}^{ML}] = \left(\mathbb{E} \left[\left(\widehat{Q}_{\{N_\ell\}, L}^{ML} - \mathbb{E}[Q] \right)^2 \right] \right)^{1/2} \approx \left(\frac{1}{M} \sum_{i=1}^M \left(\widehat{Q}_i - \mathbb{E}[Q] \right)^2 \right)^{1/2}.$$

Also, for each of the eight tolerances ϵ , we computed the run-specific estimation errors $|\widehat{Q}_i - \mathbb{E}[Q]|$, $i = 1, \dots, M$. In Figure 2 we present three plots of the RMSE versus ϵ , one for each value of q . We can see that the method yields solutions with the correct accuracy.

In order to verify Theorem 6.1, we estimated the expected cost for each tolerance ϵ and value of q by computing the mean of the total cost over 100 runs. The cost for each realization was computed using the cost model in (4.5). The cost for realizations differs not only between levels ℓ , but also within a level ℓ owing to the selective refinement algorithm. For each run i , the costs of all realizations were summed to obtain the total cost for that run. We computed a mean of the total costs for the 100 runs. A plot of the result can be found in Figure 3. As

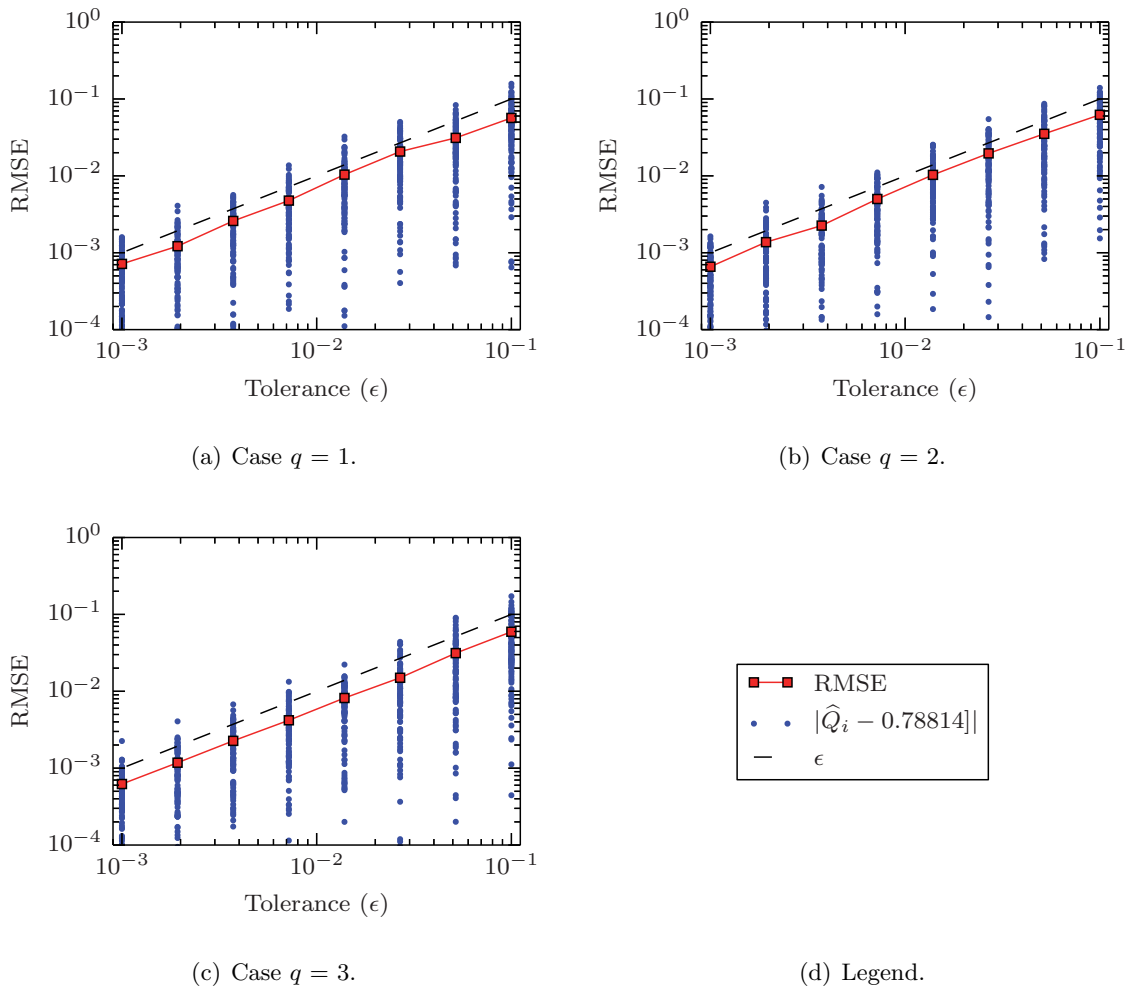


Figure 2. RMSE (square markers and line) plotted versus tolerance for the experiment described in section 8.1. The dashed line is the tolerance ϵ and the dots are the individual errors for the 100 runs at each tolerance.

the tolerance ϵ decreases, the expected cost approaches the rates given in Theorem 6.1. The reference costs are multiplied by constants to align well with the estimated expected costs.

8.2. Single-phase flow in media with log-normal permeability. We consider Darcy's law on a unit square $[0, 1]^2$ on which we have impervious upper and lower boundaries, high pressure on the left boundary (Γ_1), and low pressure on the right boundary (Γ_2). We define the spaces $H_f^1(\mathcal{D}) = \{v \in H^1(\mathcal{D}) : v|_{\Gamma_1} = f \text{ and } v|_{\Gamma_2} = 0\}$, and let n denote the unit normal of \mathcal{D} .

The weak form of the PDE reads as follows: find $u \in H_1^1(\mathcal{D})$ such that

$$(8.1) \quad (a(\omega, \cdot) \nabla u, \nabla v) = 0 \quad \text{in } \mathcal{D}$$

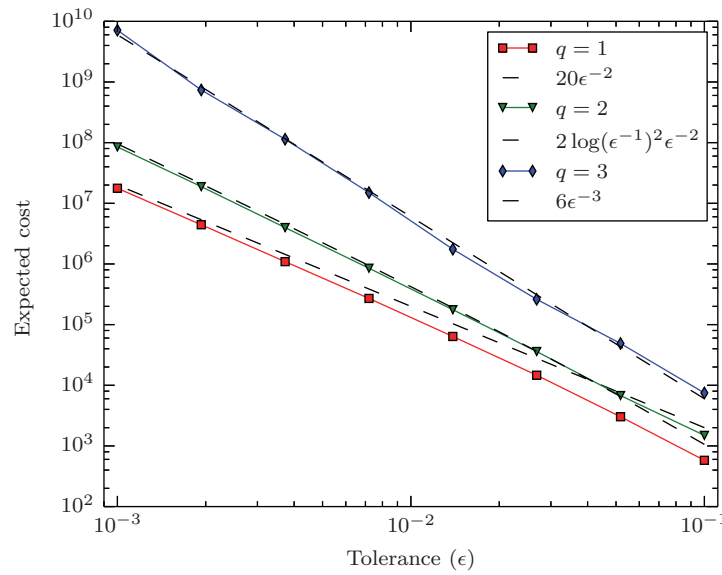


Figure 3. Computed mean total cost (diamond, triangle, square markers and lines) plotted with theoretical reference cost (dashed lines) for the experiment described in section 8.1. The reference costs for the three values of q are $20\epsilon^{-2}$ for $q = 1$, $2\log(\epsilon^{-1})^2\epsilon^{-2}$ for $q = 2$, and $6\epsilon^{-3}$ for $q = 3$.

for all $v \in H_0^1(\mathcal{D})$, and a is a stationary log-normal distributed random field

$$(8.2) \quad a(\omega, \cdot) = \exp(\kappa(\omega, \cdot))$$

over \mathcal{D} , where $\kappa(\cdot, x)$ has zero mean and is normal distributed with exponential covariance; i.e., for all $x_1, x_2 \in \mathcal{D}$ we have that

$$(8.3) \quad \mathbb{E}[\kappa(\cdot, x_1)\kappa(\cdot, x_2)] = \sigma^2 \exp\left(\frac{-\|x_1 - x_2\|_2}{\rho}\right).$$

We choose $\sigma = 1$ and $\rho = 0.1$ in the numerical experiment.

We are interested in the boundary flux on the right boundary, i.e., the functional $X(\omega) = \int_{\Gamma_2} n \cdot a(\omega)\nabla u \, dx = (a(\omega, \cdot)\nabla u, \nabla g)$, for any $g \in H^1(\mathcal{D})$, $g|_{\Gamma_1} = 0$, and $g|_{\Gamma_2} = 1$. The last equality comes by a generalized Green's identity; see [12, Chap. 1, Cor. 2.1].

To generate realizations of $a(\omega, \cdot)$, the circulant embedding method introduced in [7] is employed. The mesh resolution for the input data of the realizations generated on level ℓ in the MLMC-SR algorithm is chosen such that the finest mesh needed on level ℓ is not finer than the chosen mesh. For a fixed realization on level ℓ we do not know how fine our data should be because of the selective refinement procedure. This means that the complexity obtained for the MLMC-SR algorithm does not apply for the generation of data. The circulant embedding method has log-linear complexity. A remedy for the complexity of generating realizations is to use a truncated Karhunen–Loève expansion that can easily be refined. However, numerical experiments show that we are in a regime where the time spent on generating realizations using circulant embedding is negligible compared to the time spent in the linear solvers.

Table 3

Parameters used for the single-phase flow experiment. The parameters $y, q, \gamma, N, k, \epsilon$ are used in the MLMC-SR algorithm and ρ, σ to define the log-normal field.

Parameter	Value
y	1.5
q	2
γ	0.5
N	10
k	1
ϵ	$10^{-1}, 10^{-1.5}, 10^{-2}$
ρ	0.1
σ	1

The PDE is discretized using a finite element method (FEM) discretization with linear Lagrange elements. We have a family of structured nested meshes \mathcal{T}_{h_m} , where a mesh parameter h_m is the maximum element diameter of the given mesh. The data $a(\omega, \cdot)$ is defined in the grid points of the meshes. Using the circulant embedding we get an exact representation of the stochastic field in the grid points of the given mesh. This can be interpreted as not making any approximation of the stochastic field but instead making a quadrature error when computing the bilinear form.

The functional for a discretization on mesh m is defined as $X_{h_m}(\omega) = (a(\omega, \cdot) \nabla u_{h_m}, \nabla g)$. The convergence rates in energy norm for log-normal data is $h^{1/2-\delta}$ for any $\delta > 0$ [4]. Using postprocessing, it can be shown that the error in the functional converges twice as fast [11], i.e., $|X_{h_m} - X_{h_m}(\omega)| \leq Ch^{s-2\delta}$ for $s = 1$. We use a multigrid solver that has linear $\alpha = 1$ (up to log-factors) complexity. The work for one sample can then be computed as $\gamma^{-q\ell}$, where γ^ℓ is the numerical bias tolerance for the sample and $q \approx 2\alpha/s = 2$, which was also verified numerically. The error is estimated using the dual solution computed on a finer mesh. Since it can be quite expensive to solve a dual problem for each realization of the data, the error in the functional can also be computed by estimating the constant C and s either numerically or theoretically.

We choose $\gamma = 0.5$, $N = 10$, and $k = 1$ in the MLMC-SR algorithm; see section 7 for more information on the choices of parameters. The problem reads as follows: find the probability p for $X \leq y = 1.5$ to the given RMSE ϵ . We compute p for $\epsilon = 10^{-1}, 10^{-1.5}$, and 10^{-2} . All parameters used in the simulation are presented in Table 3.

To verify the accuracy of the estimator we compute 100 simulations of the MLMC-SR estimator for each RMSE ϵ and present the sample standard deviation (square root of the sample variance) of the MLMC-SR estimators in Table 4.

We see that in all three cases the sample std is smaller than the statistical contribution $\epsilon/\sqrt{2}$ of the RMSE ϵ . Since the exact flux is unknown, the numerical contribution in the estimator has to be approximated to be less than $\epsilon/\sqrt{2}$ as well, which is done in the termination criterion of the MLMC-SR algorithm, so it is not presented here. The mean number of samples computed to the different tolerances on each level of the MLMC-SR algorithm is computed from 100 simulations of the MLMC-SR estimator for $\epsilon = 10^{-2}$ and is shown in Table 5.

Table 5 shows that the selective refinement algorithm only refines a fraction of all problems to the highest accuracy level $j = \ell$. Using an MLMC method (without selective refinement)

Table 4

The mean failure probability p and sample standard deviation (std) computed using 100 MLMC-SR estimators and compared to the target std which is the statistical part of the RMSE error ϵ .

ϵ	Mean p	Sample std	Target std ($\epsilon/\sqrt{2}$)
10^{-1}	0.8834	$6.472 \cdot 10^{-2}$	$7.071 \cdot 10^{-2}$
$10^{-1.5}$	0.8890	$1.873 \cdot 10^{-2}$	$2.236 \cdot 10^{-2}$
10^{-2}	0.8933	$5.557 \cdot 10^{-3}$	$7.071 \cdot 10^{-3}$

Table 5

The distribution of realizations solved to different tolerance levels j for the case $\epsilon = 10^{-2}$. The table is based on the mean of 100 runs.

ℓ	0	1	2	3	4
Mean N_ℓ	16526.81	9045.41	4524.83	1471.63	738.63
$j = 0$	16526.81	4520.99	2265.23	734.21	366.90
$j = 1$		4524.42	1486.62	484.11	244.69
$j = 2$			772.98	232.33	116.77
$j = 3$				20.98	9.76
$j = 4$					0.51

the N_ℓ problem would be solved to the highest accuracy level. Using the cost model γ^{-q_ℓ} for $\epsilon = 10^{-2}$, we gain a factor ~ 6 in computational cost for this particular problem using MLMC-SR compared to MLMC. From Theorem 6.1 the computational cost for MLMC-SR and MLMC increases as $\epsilon^{-2} \log(\epsilon^{-1})^2$ and ϵ^{-3} , respectively.

Appendix A. Derivation of optimal level sample size. To determine the optimal sample level size N_ℓ in (4.2), we minimize the total cost, keeping the variance of the MLMC estimator equal to $\epsilon^2/2$, i.e.,

$$(A.1) \quad \begin{aligned} & \min \sum_{\ell=0}^L N_\ell c_\ell \\ & \text{subject to } \sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] = \epsilon^2/2, \end{aligned}$$

where $Y_\ell = Q_\ell - Q_{\ell-1}$. We reformulate the problem using a Lagrangian multiplier μ for the constraint. Define the objective function

$$(A.2) \quad g(N_\ell, \mu) = \sum_{\ell=0}^L N_\ell c_\ell + \mu \left(\sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] - \epsilon^2/2 \right).$$

The solution is a stationary point (N_ℓ, μ) such that $\nabla_{N_\ell, \mu} g(N_\ell, \mu) = 0$. Denoting by \hat{N}_ℓ and $\hat{\mu}$ the components of the gradient, we obtain

$$(A.3) \quad \nabla_{N_\ell, \mu} g(N_\ell, \mu) = (c_\ell - \mu N_\ell^{-2} \mathbb{V}[Y_\ell]) \hat{N}_\ell + \left(\sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] - \epsilon^2/2 \right) \hat{\mu}.$$

Choosing $N_\ell = \sqrt{\mu \mathbb{V}[Y_\ell]/c_\ell}$ makes the \hat{N}_ℓ components zero. The $\hat{\mu}$ component is zero when $\sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] = \epsilon^2/2$. Plugging in N_ℓ yields $2\epsilon^{-2} \sum_{\ell=0}^L \sqrt{\mathbb{V}[Y_\ell]c_\ell} = \sqrt{\mu}$, and hence the optimal sample size is

$$(A.4) \quad N_\ell = 2\epsilon^{-2} \sqrt{\mathbb{V}[Y_\ell]/c_\ell} \sum_{k=0}^L \sqrt{\mathbb{V}[Y_k]c_k}.$$

REFERENCES

- [1] S.-K. AU AND J. L. BECK, *Estimation of small failure probabilities in high dimensions by subset simulation*, Probab. Engrg. Mech., 16 (2001), pp. 263–277.
- [2] R. AVIKAINEN, *On irregular functionals of SDEs and the Euler scheme*, Finance Stoch., 13 (2009), pp. 381–401.
- [3] A. BARTH, C. SCHWAB, AND N. ZOLLINGER, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numer. Math., 119 (2011), pp. 123–161.
- [4] J. CHARRIER, R. SCHEICHL, AND A. L. TECKENTRUP, *Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods*, SIAM J. Numer. Anal., 51 (2013), pp. 322–352.
- [5] 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. Vis. Sci., 14 (2011), pp. 3–15.
- [6] N. COLLIER, A.-L. HAJI-ALI, F. NOBILE, E. VON SCHWERIN, AND R. TEMPONE, *A continuation multi-level Monte Carlo algorithm*, BIT, 55 (2015), pp. 399–432.
- [7] C. R. DIETRICH AND G. N. NEWSAM, *Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix*, SIAM J. Sci. Comput., 18 (1997), pp. 1088–1107. doi:10.1137/S1064827592240555.
- [8] D. ELFVERSON, D. J. ESTEP, F. HELLMAN, AND A. MÅLQVIST, *Uncertainty quantification for approximate p-quantiles for physical models with stochastic inputs*, SIAM/ASA J. Uncertain. Quantif., 2 (2014), pp. 826–850. doi:10.1137/140967039.
- [9] D. ESTEP, A. MÅLQVIST, AND S. TAVENER, *Nonparametric density estimation for randomly perturbed elliptic problems. I: Computational methods, a posteriori analysis, and adaptive error control*, SIAM J. Sci. Comput., 31 (2009), pp. 2935–2959.
- [10] M. B. GILES, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617.
- [11] M. B. GILES AND E. SÜLI, *Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality*, Acta Numer., 11 (2002), pp. 145–236.
- [12] V. GIRAULT AND P.-A. RAVIART, *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms*, Springer Ser. Comput. Math. 5, Springer-Verlag, Berlin, 1986.
- [13] P. GLASSERMAN, P. HEIDELBERGER, P. SHAHABUDDIN, AND T. ZAJIC, *Splitting for rare event simulation: Analysis of simple cases*, in Proceedings of the 1996 Winter Simulation Conference, IEEE Press, Piscataway, NJ, 1996, pp. 302–308.
- [14] P. GLYNN, *Importance sampling for Monte Carlo estimation of quantiles*, in Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation, S. M. Ermakov, ed., Publishing House of St. Petersburg University, St. Petersburg, Russia, 1996, pp. 180–185.
- [15] 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., (2015), pp. 1–37.
- [16] E. JONES, T. OLIPHANT, P. PETERSON, ET AL., *SciPy: Open Source Scientific Tools for Python*, 2001. <http://www.scipy.org/> (accessed 2016-04-01).
- [17] A. KEBAIER, *Statistical Romberg extrapolation: A new variance reduction method and applications to options pricing*, Ann. Appl. Probab., 14 (2005), pp. 2681–2705.
- [18] A. LOGG, K.-A. MARDAL, AND G. WELLS, *Automated Solution of Differential Equations by the Finite Element Method*, Lect. Notes Comput. Sci. Eng. 84, Springer, Berlin, Heidelberg, 2012.
- [19] A. L. TECKENTRUP, R. SCHEICHL, M. B. GILES, AND E. ULLMANN, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numer. Math., 125 (2013), pp. 569–600.