Water Resources Research

RESEARCH ARTICLE
10.1029/2019WR026984

Accelerated Multilevel Monte Carlo With Kernel-Based Smoothing and Latinized Stratification

Søren Taverniers[1], Sebastian B. M. Bosma[1], and Daniel M. Tartakovsky[1]

[1]Department of Energy Resources Engineering, Stanford University, Stanford, CA, USA

Key Points:
- Kernel-based smoothing and Latinized stratified sampling accelerate multilevel Monte Carlo (MLMC) estimators for distributions
- Kernel-based MLMC is up to 2 orders of magnitude faster than standard Monte Carlo simulations
- MLMC with Latinized stratified sampling yields comparable levels of speedup without introducing a smoothing error

Correspondence to:
D. M. Tartakovsky,
tartakovsky@stanford.edu

Citation:
Taverniers, S., Bosma, S. B. M., & Tartakovsky, D. M. (2020). Accelerated multilevel Monte Carlo with kernel-based smoothing and Latinized stratification. *Water Resources Research*, *56*, e2019WR026984. https://doi.org/10.1029/2019WR026984

Received 19 DEC 2019
Accepted 24 JUL 2020
Accepted article online 28 JUL 2020

Abstract Heterogeneity and a paucity of measurements of key material properties undermine the veracity of quantitative predictions of subsurface flow and transport. For such model forecasts to be useful as a management tool, they must be accompanied by computationally expensive uncertainty quantification, which yields confidence intervals, probability of exceedance, and so forth. We design and implement novel multilevel Monte Carlo (MLMC) algorithms that accelerate estimation of the cumulative distribution functions (CDFs) of quantities of interest, for example, water breakthrough time or oil production rate. Compared to standard non-smoothed MLMC, the new estimators achieve a significant variance reduction at each discretization level by smoothing the indicator function with a Gaussian kernel or replacing standard Monte Carlo (MC) with the recently developed hierarchical Latinized stratified sampling (HLSS). After validating the kernel-smoothed MLMC and HLSS-enhanced MLMC methods on a single-phase flow test bed, we demonstrate that they are orders of magnitude faster than standard MC for estimating the CDF of breakthrough times in multiphase flow problems.

1. Introduction

Physics-based models of subsurface flow and transport play a critical role in management of groundwater, hydrocarbon, and geothermal resources. A typical model takes the form of a numerical solution of (a coupled system of) partial differential equations (PDEs) representing relevant conservation laws. Such models are parametrized with a set of material (and fluid) properties, such as intrinsic and relative permeabilities, porosity, and dispersivity. Since most subsurface environments exhibit a large degree of heterogeneity on the multiplicity of scales, their properties vary in space and cannot be represented exactly in a numerical model due to incomplete and/or inaccurate measurements. Hence, values of any or all of these parameters should be considered uncertain (Tartakovsky & Winter, 2008), and this input uncertainty leads to uncertainty in output quantities of interest (QoIs).

A probabilistic framework for quantification of predictive uncertainty treats uncertain input parameters and model outputs (QoIs) as random variables/fields/processes. Thus, a single choice of parameter values, and a resulting prediction of QoIs, is thought of as a sample from corresponding probability distributions of the model's input and output. Probabilistic formulation of a subsurface model consists of specifying input parameters in terms of their probability density functions (PDFs) or cumulative distribution functions (CDFs). A solution of this problem takes the form of PDFs/CDFs of the system state or derived QoIs.

Monte Carlo (MC) simulations (Fishman, 1996) are routinely used to compute such solutions or their moments (e.g., means and variances of QoIs). The popularity of MC stems from its ease of use and nonintrusive character, that is, the ability to use existing solvers and "off-the-shelf" software. On a more technical level, MC benefits from a convergence (i.e., the number of realizations, N, needed to achieve a required sampling accuracy) that is independent of the number of random inputs (the so-called stochastic dimension). Unfortunately, this convergence is slow: The standard deviation of an MC estimator of the QoI's expected value (aka mean or average) is inversely proportional to $\sqrt{N}$. This renders MC computationally demanding, and often prohibitively so, when each model run is expensive (e.g., when a high spatial and/or temporal resolution is required).

To achieve the same sampling error (estimator variance) with fewer realizations, standard MC may be replaced with a more computationally efficient sampling design, which is one of the main drivers of uncertainty quantification (UQ) research (Tartakovsky, 2017). UQ techniques based on stochastic finite elements, including stochastic Galerkin and stochastic collocation, outperform MC simulations in problems with

relatively low stochastic dimensions (Tartakovsky, 2017, and the references therein). However, they become less efficient than MC for problems with either high stochastic dimensions (Taverniers & Tartakovsky, 2017)—a feature commonly referred to as the curse of dimensionality—or strong nonlinearities and moderately large parametric uncertainty (Barajas-Solano & Tartakovsky, 2016). Other methods, such as moment differential equations (Neuman et al., 1996) and the method of distributions (Tartakovsky et al., 2009; Venturi et al., 2013), are not affected by the curse of dimensionality but require closure approximations that are often derived via perturbation expansions in the variance of a random input. This formally limits their applicability to problems with small parametric uncertainty or mild heterogeneity, even though the applicability range is often significantly larger than the theory suggests (Z. Lu et al., 2002; Ye et al., 2004) and can be extended further by means of random domain decompositions (Winter & Tartakovsky, 2002; Winter et al., 2003). Just like the stochastic Galerkin method, these algorithms are intrusive, that is, require one to solve a set of deterministic equations that differ from the underlying PDEs with random coefficients.

Unlike the aforementioned UQ techniques, variance-reduction sampling methods aim to preserve MC's attractive features while improving upon its poor convergence. This class of methods includes antithetic sampling, control variates, importance sampling, Latin hypercube sampling (LHS), and stratification (Fishman, 1996). An alternative strategy to controlling the MC cost is to minimize the overall error of an MC estimator of a QoI's expected value for a given amount of available computational resources (Moslehi et al., 2015). Following the general philosophy of the resource-constrained model selection (Sinsbeck & Tartakovsky, 2015), this approach subdivides the overall mean square error (MSE) of the estimator into sampling (variance) and discretization (bias) components. The former is estimated via a sample variance, while the latter is approximated by a polynomial in powers of the discrete spatial and/or temporal mesh size.

Another approach to variance reduction combines standard MC with the multigrid method for solving PDEs (Giles, 2008; Heinrich, 1998, 2001). This method, which we adopt in the current study, has become known as multilevel MC (MLMC). It seeks to outperform MC by correcting cheaper-to-compute realizations on a coarse spatial grid with more expensive samples at finer levels of discretization. While originally designed to perform standard MC at each level, MLMC may be accelerated by replacing the latter with a modified sampling strategy such as Quasi-MC (Crevillén-García & Power, 2017; Kuo et al., 2017) or one of the variance-reduction schemes listed above (Kebaier & Lelong, 2018).

Most MLMC studies focus either on the estimation of statistical moments of a QoI (Kumar et al., 2019; Linde et al., 2017; Müller et al., 2013, 2016) or on the single-point evaluation of its CDF to estimate rare events, for example, probability of failure (Ullmann & Papaioannou, 2015). Much less work has been done on MLMC for estimation of the full CDF/PDF of a QoI. A key challenge here is the slow decay of the variance of the indicator function with discretization level, which may render MLMC less efficient than standard MC at the finest resolution (D. Lu et al., 2016). Polynomial smoothing of the indicator function can improve computational efficiency for estimating CDFs (Giles et al., 2015; D. Lu et al., 2016), as can approximation of PDFs via a truncated moment sequence (Bierig & Chernov, 2016). Indirect estimation of a CDF via an appropriate primitive function (Krumscheid & Nobile, 2018) provides yet another tool to speed up the computation.

We employ the hierarchical Latinized stratified sampling (HLSS) method (Shields, 2016) to design a more efficient MLMC algorithm for estimation of the CDF or, equivalently, exceedance probability of a QoI. We also replace the polynomial smoothing (Giles et al., 2015) with a kernel-based smoothing in order to regularize the indicator function within a standard multilevel framework (i.e., using standard MC at each level). Inspired by the framework developed in Moslehi et al. (2015) and following Giles et al. (2015) and D. Lu et al. (2016), the MSE between the estimated CDF and its exact counterpart is decomposed into a sampling error, a bias error, and (if applicable) a smoothing error. Rather than fixing the computational cost and minimizing the MSE (Moslehi et al., 2015), we specify the MSE tolerance and minimize the computational cost via a combination of the standard Lagrange multiplier approach to estimate the optimal numbers of samples at each discretization level (for the kernel-smoothed standard MLMC algorithm) and/or a tunable parameter to control the relative magnitudes of the allowable sampling error and bias.

In section 2, we discuss various MC-based approaches to CDF estimation and introduce two complementary strategies for MLMC acceleration: standard MLMC with kernel-based smoothing and HLSS-enhanced MLMC. Section 3 contains a description of the single- and two-phase flow problems used to assess the

performance of these MLMC algorithms. Section 4 describes the results of our numerical experiments. Main conclusions and future research directions are presented in section 5.

2. MC Estimation of CDFs

Consider a QoI $Q \in \mathbb{R}$ that depends on p continuous random input variables $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)$, that is, $Q = Q(\boldsymbol{\xi})$. Each input variable $\xi_i : \Omega_i \to \mathbb{R}$ is a measurable function with the sample space Ω_i. The CDF $F(q)$ of Q can be defined as the expected value $\mathbb{E}[\mathcal{I}_{(-\infty, q]}(Q)]$ of the indicator function

$$\mathcal{I}_{(-\infty, q]}(s) = \begin{cases} 1 & \text{for } s \in (-\infty, q] \\ 0 & \text{for } s \in (q, +\infty), \end{cases} \tag{1}$$

which establishes its relation to the PDF $f(q)$ of Q

$$F(q) = \int_{-\infty}^{\infty} \mathcal{I}_{(-\infty, q]}(s)\, f(s)\, ds = \int_{-\infty}^{q} f(s)\, ds. \tag{2}$$

Our goal is to estimate $F(q)$, at any point q in some compact interval $[a, b] \subset \mathbb{R}$, from its values at a set of $S+1$ equidistant points $\mathcal{S}_h = \{a = q_0 < q_1 < \ldots < q_S = b\}$ with separation distance h. We do so by employing piecewise polynomial interpolation of degree $\max\{d, 1\}$ (Giles et al., 2015), where $d \in \mathbb{N}_0$ is related to the smoothness of $f(q)$, so that $f(q)$ is at least d times continuously differentiable on $[a - \xi_0, b + \xi_0]$ for some $\xi_0 > 0$. We use cubic spline interpolation, in which case $d = 3$. An alternative is to employ Lagrange basis polynomials ϕ_n ($n = 0, \ldots, S$) (D. Lu et al., 2016), for which the approximation $F_h(q)$ of $F(q)$ in (2) is given by (In our simulations, we compute the cubic spline interpolant using a built-in MATLAB® function)

$$F_h(q) = \sum_{n=0}^{S} \mathbb{E}[\mathcal{I}_n(Q)]\phi_n(q), \qquad \mathcal{I}_n(Q) \equiv \mathcal{I}_{(-\infty, q_n]}(Q). \tag{3}$$

In hydrogeological applications and beyond, the QoI Q is an output computed from the numerical solution of a PDE with, for example, finite difference or finite volume methods. These strategies require the computational domain to be discretized with a spatial grid $\mathcal{T}_M$ consisting of M cells. Subsequent solution of the discretized PDE yields a QoI approximation Q_M, which converges to Q as M increases. We assume this convergence to hold both in the mean and in the sense of distribution, such that

$$\mathbb{E}[Q_M - Q] = \mathcal{O}(M^{-\alpha_1}), \quad \mathbb{E}[\mathcal{I}_{(-\infty, q]}(Q_M) - \mathcal{I}_{(-\infty, q]}(Q)] = \mathcal{O}(M^{-\alpha_2}) \quad \text{as } M \to \infty, \tag{4}$$

for $\alpha_1, \alpha_2 \in \mathbb{R}^+$ independent of M and q. Following (3), an approximation of the CDF $F_M(q)$ of Q_M on $[a,b]$ is given by

$$F_{h,M}(q) = \sum_{n=0}^{S} \tau_{n,M}\phi_n(q), \qquad \tau_{n,M} \equiv \mathbb{E}[\mathcal{I}_n(Q_M)]. \tag{5}$$

Another approximation stems from the replacement of ensemble means with sample means, that is, $\mathbb{E}[\mathcal{I}_n(Q_M)] \approx \hat{\mathcal{I}}_{n,M}$, yielding the CDF estimator

$$\hat{F}_{h,M}(q) = \sum_{n=0}^{S} \hat{\mathcal{I}}_{n,M}\phi_n(q). \tag{6}$$

To sum up, the estimation error introduced by the above approximations, that is, the discrepancy between the true CDF $F(q)$ and its estimator $\hat{F}_{h,M}$, has two sources: the discretization error (or *bias*) related to approximating F_h by $F_{h,M}$ and the sampling error related to approximating $F_{h,M}$ by $\hat{F}_{h,M}$. (For all estimators $\hat{F}_{h,M}$ considered in this work, we assume that the number of interpolation points $S+1$ is large enough for the interpolation error to be negligible.) The MSE of this estimation, $\varepsilon^2_{\text{est}}$, is bounded by

$$\underbrace{\mathbb{E}[||F_h - \hat{F}_{h,M}||_\infty^2]}_{\varepsilon_{\text{est}}^2} \leq \mathbb{E}[||\hat{F}_{h,M} - \mathbb{E}[\hat{F}_{h,M}]||_\infty^2] + ||F_{h,M} - F_h||_\infty^2 \leq \underbrace{\max_{0 \leq n \leq S} \mathbb{V}[\hat{\mathcal{F}}_{n,M}]}_{\varepsilon_{\text{sam}}^2} + \underbrace{\max_{0 \leq n \leq S} |\mathbb{E}[\mathcal{F}_n(Q_M) - \mathcal{F}_n(Q)]|^2}_{\varepsilon_{\text{dis}}^2}, \tag{7}$$

where $||\cdot||_\infty$ denotes the L^∞ norm; $\mathbb{V}[\cdot]$ refers to the variance operator; and ε_{sam} and ε_{dis} are, respectively, the sampling and discretization error, in the root mean square sense.

In the remainder of this section, we consider the random input variables $\xi_1, \ldots, \xi_p$ to be mutually uncorrelated and characterized by their respective CDFs F_{ξ_i}. As discussed in section 4, these variables can be used to build a correlated permeability field via a Karhunen-Loève (KL) expansion.

2.1. Standard MLMC With Kernel-Based Smoothing

The standard MLMC estimator (Giles et al., 2015) of $F(q)$ is described in Appendix A1. The jump discontinuity in the indicator function used to construct a CDF may lead to a slow decay of its variance with increasing spatial resolution. This may render MLMC less efficient than fine-resolution MC for sufficiently large values of the error tolerance ϵ (D. Lu et al., 2016). To obviate this problem, one can replace the indicator function with a pth degree polynomial of a certain bandwidth $\delta_{G,l}$ at level l (see Appendix A2 for details). Finding the optimal smoothing function therefore requires tuning two parameters: the bandwidth and the polynomial degree.

We propose an alternative regularization of the indicator function based on kernel density estimation (KDE) (Rosenblatt, 1956), which has only one tuning parameter (the bandwidth). To implement our KDE-based smoothing, we replace the indicator function $\mathcal{I}_n(Q)$, where $n = 0, \ldots, S$, with $\Phi[(q_n - Q)/\delta] \equiv g_K[(q_n - Q)/\delta]$, where Φ is the CDF of the standard normal distribution and δ is the bandwidth over which the jump discontinuity is smeared out. The resulting MLMC estimator with smoothing for $\tau_{n,M}$ is

$$\hat{\mathcal{F}}_{n,M}^{\text{MLsm}} = \sum_{l=0}^{L_{\max}} \hat{\mathcal{F}}_n^{\text{MCsm}}(Y_l), \tag{8a}$$

where $\hat{\mathcal{F}}_n^{\text{MCsm}}(Y_l) = N_l^{-1} \sum_{j=1}^{N_l} g_n(Y_l^{(j)})$ and

$$g_n(Y_l^{(j)}) = \begin{cases} g_K\left(\dfrac{q_n - Q_{M_l}^{(j)}}{\delta_{K,l}}\right) - g_K\left(\dfrac{q_n - Q_{M_{l-1}}^{(j)}}{\delta_{K,l}}\right) & 1 \leq l \leq L_{\max} \\ g_K\left(\dfrac{q_n - Q_{M_l}^{(j)}}{\delta_{K,l}}\right) & l = 0. \end{cases} \tag{8b}$$

The chosen bandwidth ($\delta_{K,l}$) is level dependent, similar to its counterpart for polynomial smoothing (Appendix A2). The superscript MLsm stands for "smoothed MLMC".

The MSE of the kernel-smoothed MLMC estimator $\hat{F}_{h,\delta_K,M}^{\text{MLsm}}$ for $F_{h,M}$ is bounded by

$$\underbrace{\mathbb{E}[||F_h - \hat{F}_{h,\delta_K,M}^{\text{MLsm}})||_\infty^2]}_{(\varepsilon_{\text{est}}^{\text{MLsm}})^2} \leq \underbrace{\mathbb{E}[||\hat{F}_{h,\delta_K,M}^{\text{MLsm}} - \mathbb{E}[\hat{F}_{h,\delta_K,M}^{\text{MLsm}}]||_\infty^2]}_{(\varepsilon_{\text{sam}}^{\text{MLsm}})^2} + \underbrace{||F_{h,M} - F_h||_\infty^2}_{(\varepsilon_{\text{dis}}^{\text{MLsm}})^2} + \underbrace{||\mathbb{E}[\hat{F}_{h,\delta_K,M}^{\text{MLsm}}] - F_{h,M}||_\infty^2}_{(\varepsilon_{\text{sm}}^{\text{MLsm}})^2}. \tag{9}$$

To satisfy a user-specified error tolerance ϵ, we follow the procedure described in Appendix A2.

Our method has several advantages over the polynomial-smoothed MLMC (D. Lu et al., 2016):

1. Only one tunable parameter (bandwidth) instead of two (bandwidth and polynomial degree) is required to define the regularization of the indicator function. This reduces algorithmic complexity by eliminating an extra loop over possible polynomial degrees to find the optimally efficient estimator.
2. The use of a polynomial of degree d requires the QoI's PDF to be at least d times continuously differentiable (Giles et al., 2015). This introduces an additional constraint that needs to be taken into account.
3. Rather than estimating the optimal number of samples at each level *after* the maximum level has been reached, our method optimizes the number of samples required to satisfy the sampling error tolerance *throughout the entire algorithm* and at each level also recomputes prior estimates at *all previous levels*. This improves the accuracy of the sample size estimation (Cliffe et al., 2011).
4. The introduction of a free parameter α (Appendix A2) allows for more flexibility in the division between sampling and discretization error, enabling additional tuning of the algorithm to minimize the computational cost.

An implementation of our kernel-smoothed MLMC estimator, including a computation of its cost, is provided in Appendix C1. Our algorithm allows for computing its fine-resolution MC counterpart and the latter's computational cost. That information is used only for comparison purposes, rather than for a "hybrid" scheme, which switches to MC if no speedup is achieved with MLMC (D. Lu et al., 2016). Numerical experiments reported in section 4 demonstrate that, for error tolerances typically encountered in subsurface flow applications, our kernel-based MLMC is more efficient than fine-resolution MC. Future refinements to the algorithm (see section 5) could result in further speedup, especially for rare-event estimation where only tails of a CDF need to be characterized.

2.2. HLSS-Enhanced MLMC

2.2.1. Hierarchical Latinized Stratified Sampling

The Latinized stratified sampling (LSS) method (Shields & Zhang, 2016) aims to combine the benefits of stratified sampling (SS) and LHS. SS is good at reducing the variance associated with interactions between input variables and works well in low stochastic dimensions p (Shields & Zhang, 2016; Shields et al., 2015), while LHS provides strong variance reduction for additive (main) effects (Stein, 1987). LHS is superior to SS in high stochastic dimensions, $p > \log_2(N)$ where N denotes the number of samples (Shields et al., 2015). LSS simultaneously defines an LHS design and a p-dimensional SS design on the unit hypercube $[0, 1]^p$, which requires enforcement of two compatibility conditions between both strategies: All SS strata must coincide with an LHS stratum boundary, and they must all be equally weighted hyperrectangles.

The hierarchical LHS method (Shields, 2016) extends the sample size of an LSS design by breaking up existing strata and adding the unallocated samples to the newly created empty stratum. This is more optimal than adding samples to the existing strata (Shields et al., 2015). The HLSS estimator for $\tau_{n,M}$ based on N_{HLSS} independent samples of Q_M is

$$\hat{\mathcal{F}}^{\text{HLSS}}_{n,M} = \frac{1}{N_{\text{HLSS}}} \sum_{j=1}^{N_{\text{HLSS}}} \mathcal{I}_n(Q_M^{(j)}). \tag{10}$$

Since HLSS can be regarded as a stratified, proportionally allocated sampling design with one sample per stratum, the variance of $\hat{\mathcal{F}}^{\text{HLSS}}_{n,M}$ is given by (B7),

$$\mathbb{V}[\hat{\mathcal{F}}^{\text{HLSS}}_{n,M}] = \mathbb{V}[\hat{\mathcal{F}}^{\text{MC}}_{n,M}] - \frac{1}{N^2_{\text{HLSS}}} \sum_{j=1}^{N_{\text{HLSS}}} (\mu_{j,n} - \tau_{n,M})^2, \tag{11}$$

where the first term on the right-hand side is defined in (A2) and $\mu_{j,n}$ is the mean of $\mathcal{I}_n(Q_M)$ over the jth stratum.

HLSS requires at least a doubling of the number of samples upon each sample size extension. The refined LSS method (Shields, 2016) resolves this drawback by allowing for unequal sample weights. However, the latter necessitates the computation of strata variances to calculate the estimator variance. This introduces additional complexity in the case of a one-sample-per-stratum design (Shields, 2018) and invalidates (11). Hence, we consider only the HLSS sampling design.

2.2.2. Integrating HLSS Into a Multilevel Estimator

Integration of the HLSS approach (section 2.2.1) into the multilevel framework yields additional variance reduction compared to the standard MLMC estimator with simple MC in each discretization level. We refer to the resulting algorithm as HLSS-MLMC. Its estimator for $\tau_{n,M}$ is

$$\hat{\mathscr{I}}_{n,M}^{\text{HLSS -ML}} = \sum_{l=0}^{L_{\max}} \hat{\mathscr{I}}_{n}^{\text{HLSS}}(Y_l) = \sum_{l=0}^{L_{\max}} \frac{1}{N_l} \sum_{j=1}^{N_l} \mathscr{I}_n(Y_l^{(j)}), \tag{12}$$

where $\mathscr{I}_n(Y_l)$ with $l = 0, \ldots, L_{\max}$ are defined in (A3b). Its variance is given by (McKay et al., 2000)

$$\mathbb{V}[\hat{\mathscr{I}}_{n,M}^{\text{HLSS -ML}}] = \sum_{l=0}^{L_{\max}} \left[\mathbb{V}[\hat{\mathscr{I}}_{n}^{\text{MC}}(Y_l)] - \frac{1}{N_l^2} \sum_{j=1}^{N_l} (\mu_{j,n,l} - \tau_{n,l})^2 \right], \tag{13}$$

where $\mathbb{V}[\hat{\mathscr{I}}_{n}^{\text{MC}}(Y_l)]$ is defined analogously to (A2), $\mu_{j,n,l}$ is the jth stratum mean of $\mathscr{I}_n(Y_l)$, and $\tau_{n,l}$ is the mean of $\mathscr{I}_n(Y_l)$ over the entire sample space. The MSE of the HLSS-MLMC estimator for $F_{h,M}$, $\hat{F}_{h,M}^{\text{HLSS -ML}}$, is bounded in a similar fashion to its counterpart for $\hat{F}_{h,M}^{\text{ML}}$. We use the algorithm from Appendix C2 to compute $\hat{F}_{h,M}^{\text{HLSS -ML}}$ and measure its computational cost.

One can smooth the indicator function $\mathscr{I}_n(Y_l)$ by replacing it with $g_n(Y_l)$ defined in (8b) to obtain additional variance reduction. The smoothing will result in an additional term in the estimator's MSE.

2.3. Costs of MLMC and Fine-Resolution MC

To estimate the cost $\mathscr{C}$ of computing the non-smoothed MLMC estimator, $F_{h,M}^{\text{ML}}$, for an error tolerance ϵ, we consider an average over N_{real} independent realizations of the algorithm,

$$\mathscr{C}(\hat{F}_{h,M}^{\text{ML}}) = \frac{1}{N_{\text{real}}} \sum_{p=1}^{N_{\text{real}}} \sum_{l=0}^{L_{\max,p}^{\epsilon}} \bar{w}_l^{(p)} N_l^{(p)}. \tag{14}$$

Here $\overline{w}_l(p)$ is the average cost of computing a sample of Q_{M_l} on level l for the pth realization, and $L_{\max,p}^{\epsilon}$ is the finest discretization level at which sampling is performed for this realization at tolerance ϵ.

To compare the cost of $\hat{F}_{h,M}^{\text{ML}}$ with that of MC, we perform the latter on the finest level, $L_{\max,p}^{\epsilon}$, to ensure that both estimators satisfy the discretization part of the tolerance ϵ. The cost of the fine-resolution MC estimator, $\hat{F}_{h,M}^{\text{MC}}$, is

$$\mathscr{C}(\hat{F}_{h,M}^{\text{MC}}) = \frac{1}{N_{\text{real}}} \sum_{p=1}^{N_{\text{real}}} \bar{w}_{L_{\max,p}^{\epsilon}}^{(p)} N_{\text{MC}}^{(p)}, \tag{15}$$

where $N_{\text{MC}}^{(p)}$ is the number of samples computed in the pth realization of the algorithm.

We fix the maximum level for all multilevel variants, other than $\hat{F}_{h,M}^{\text{ML}}$, to the most frequently observed maximum level across all N_{real} realizations of the latter; this value is denoted by $L_{\max}^{\epsilon}$. We do so because the smaller number of samples at finer levels for those estimators may not yield sufficiently accurate estimates of the discretization error; these are based on a sample estimate of $\max_{0 \le n \le S} |\mathbb{E}[\mathscr{I}_n(Y_l)]|$ in accordance with (A7). Since the discretization error is dictated by the maximum discretization level, and since the discretization tolerance is taken to be identical for all estimators, it is reasonable to assume that if $\hat{F}_{h,M}^{\text{ML}}$ satisfies the required discretization error tolerance for a certain maximum level $L_{\max}^{\epsilon}$, then the other estimators also satisfy this tolerance when having $L_{\max}^{\epsilon}$ as their maximum level.

3. Numerical Experiments

To demonstrate the performance of our accelerated MLMC algorithms, we investigate one- and two-phase flows in heterogeneous porous media. In the following sections, we describe the governing equations of

these test beds and their discretization and explain the upscaling of material properties from finer to coarser levels.

3.1. One-Dimensional Single-Phase Flow

Spatiotemporal evolution of hydraulic head $h(x,t)$ in a one-dimensional heterogeneous porous medium D is described by

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left[k(x)\frac{\partial h}{\partial x}\right], \qquad x \in D, \quad t \in (0,\, T], \tag{16a}$$

where $k(x) = K(x)/S$ denotes the spatially varying hydraulic conductivity $K(x)$ normalized with a constant specific storage S. This equation is subject to an initial condition

$$h(x,\, 0) = h_{\text{in}}(x), \qquad x \in D \tag{16b}$$

and boundary conditions

$$\mathscr{B}(h,\, x,\, t) = b(x,\, t), \quad x \in \partial D, \quad t \in (0,\, T], \tag{16c}$$

where the boundary operator $\mathscr{B}$ represents Dirichlet and/or Neumann boundary conditions. As a practical matter, it is impossible to know $k(x)$ exactly at each point $x \in D$, rendering its spatial distribution uncertain. Consequently, (16) is recast in the probabilistic framework, which treats the input $k(x)$ and output $h(x,t)$ as random fields. Let $(\Omega,\, \mathscr{F},\, \mathscr{P})$ be a complete probability triple, where Ω is the sample space, $\mathscr{F} \subseteq 2^{\Omega}$ is the σ-algebra of events, and $\mathscr{P}: \mathscr{F} \to [0,\, 1]$ is the probability measure. We extend the domain of definition of $k(x)$ to the sample space Ω, so that $k = k(x,\, \omega): D \times \Omega \to \mathbb{R}$. Following the standard practice in stochastic hydrogeology, we assume that the random field $k(x,\omega)$ is lognormally distributed and that its natural logarithm $Y(x,\, \omega) = \ln k(x,\, \omega)$ has a continuous autocovariance function $C_Y(x,\, y) = \mathbb{E}\{[Y(x,\, \omega) - \mu_Y(x)][Y(y,\, \omega) - \mu_Y(y)]\}$ where $\mu_Y = \mathbb{E}(Y)$. For the sake of simplicity, we assume the initial and boundary conditions to be deterministic.

We represent the Gaussian field $Y(x,\omega)$ via a truncated KL expansion

$$\hat{Y}(x,\, \xi_p(\omega)) = \mu_Y(x) + \sum_{l=1}^{p} \sqrt{\gamma_l}\phi_l(x)\xi_l(\omega), \tag{17}$$

where the number of terms, p, retained in the otherwise infinite series is referred to as the *stochastic dimension*; γ_l and $\phi_l(x)$ are, respectively, the eigenvalues and eigenfunctions of the autocovariance function $C_Y(x,y)$; and $\boldsymbol{\xi}_p(\omega) = (\xi_1, \ldots, \xi_p)^{\top}$ is a set of i.i.d. standard Gaussian random variables; that is, it is characterized by a standard multivariate Gaussian PDF

$$\rho(\mathbf{s}) = (2\pi)^{-p/2}\exp\left(-\frac{1}{2}\mathbf{s}^{\top}\,\mathbf{s}\right) \tag{18}$$

with support $\mathbb{R}^p$. For a given variance of $Y(x,\omega)$, the value of p required to approximate the full KL expansion with a given accuracy depends on the rate of decay of the eigenvalues γ_l. This decay rate is given by the regularity of the autocovariance kernel $C_Y(x,y)$; however, regardless of the degree of regularity of C_Y, the value of p increases as the autocorrelation length of $Y(x,\omega)$ decreases. We choose $p = 17$ such that the truncated KL expansion (17) captures 95% of the energy of the field Y as determined by the square roots of the eigenvalues γ_l.

A random solution $h(x,t,\omega)$ of (16) is approximated by a random solution $\hat{h}(x,\, t,\, \omega)$ of (16) with $k(x,\omega)$ replaced by $\hat{k}(x,\, \xi_p) = \exp[\hat{Y}(x,\, \xi_p)]$. According to the Doob-Dynkin lemma, the latter solution is a function of $\boldsymbol{\xi}_p$. The solution $h_p(x,t,\boldsymbol{\xi}_p)$ is referred to as a stochastic response surface.

In the single-phase flow simulations reported below, we assume all quantities have been nondimensionalized, set $D = (0,2)$, $T = 0.2$, and $h_{\text{in}}(x) = 200 + 200\tanh[2(x-1)]$, and choose Dirichlet boundary conditions with $b(0,\, t) = 200 + 200\tanh(-2)$ and $b(2,\, t) = 200 + 200\tanh(2)$. The Gaussian field Y has zero mean,

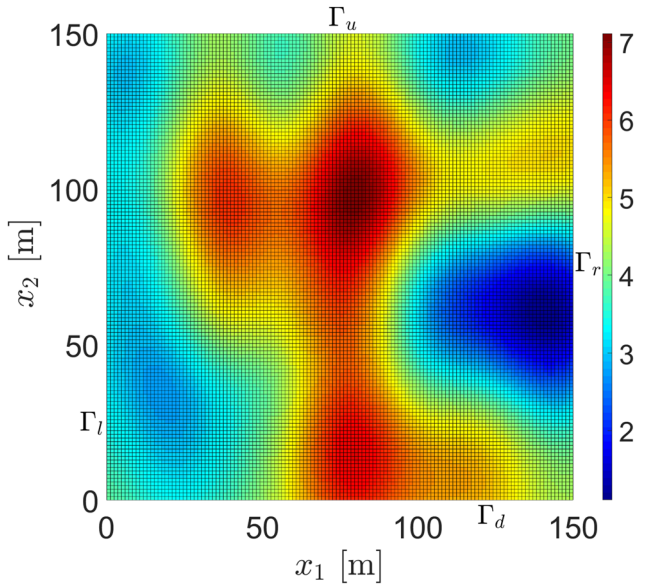

Figure 1. Domain setup for the two-phase flow problem and visualization of one realization of the log permeability field $Y = \ln k$ (with k in mDarcy) simulated at a resolution of 128×128 cells.

$\mu_Y(x) = 0$, and exponential covariance $C_Y(x, y) = \sigma_Y^2 \exp[-|x-y|/\lambda_Y]$ with the variance $\sigma_Y^2 = 0.8$ and the correlation length $\lambda_Y = 0.2$. This translates into the coefficient of variation (CV) of $k = \exp(Y)$,

$$\mathrm{CV}(k) \equiv \frac{\sigma_k}{\langle k \rangle} = \frac{\sqrt{[\exp(\sigma_Y^2)-1]\exp(2\mu_Y + \sigma_Y^2)}}{\exp[\mu_Y + (\sigma_Y^2/2)]} = 1.1.$$

The stochastic counterpart of (16) is discretized in space using a central finite difference scheme; the resulting system of initial-value problems is solved with the implicit Euler method. The matrix associated with the resulting linear system is tridiagonal. Hence, we apply the Thomas algorithm to solve it at reduced computational complexity. This numerical scheme is second-order accurate in space and first-order accurate in time.

3.2. Two-Dimensional Two-Phase Flow

Our second test deals with horizontal two-phase flow of incompressible and immiscible fluids in a random heterogeneous porous medium D that is both incompressible and isotropic. Propagation of the saturation $S_\ell(\mathbf{x},t)$ of the ℓth phase ($\ell = 1,2$) is described by the mass conservation equation

$$\phi \frac{\partial S_\ell}{\partial t} + \nabla \cdot \mathbf{u}_\ell + q_\ell = 0, \qquad \mathbf{x} \equiv (x_1, x_2)^\top \in D, \quad t \in [0, T], \tag{19a}$$

and continuity (pressure) equation

$$\nabla \cdot \mathbf{u}_{\mathrm{tot}} = 0. \tag{19b}$$

Here $\mathbf{u}_{\mathrm{tot}} = \sum_{\ell=1}^{2} \mathbf{u}_\ell$ with $\mathbf{u}_\ell$ the Darcy velocity (flux) of the ℓth phase given by

$$\mathbf{u}_\ell(\mathbf{x}) = -\mathbf{k}(\mathbf{x}) \frac{k_{r\ell}}{\mu_\ell} \cdot \nabla P_\ell(\mathbf{x}, t), \qquad \ell = 1, 2. \tag{19c}$$

In (19a), ϕ is the porosity; $S_\ell(\mathbf{x},t)$ satisfies the compatibility condition $S_1 + S_2 = 1$; and q_ℓ is a source/sink term that, in our numerical experiments, is taken to be zero but may represent, for example, one or more pumping wells. In (19c), $\mathbf{k}(\mathbf{x})$ is the intrinsic permeability tensor; since the medium is isotropic, $\mathbf{k}(\mathbf{x})$ becomes a scalar and will be denoted by k from now on. The quantities $k_{r\ell}(S_\ell)$ and μ_ℓ are the relative permeability and viscosity of the ℓth phase, respectively. We ignore capillary forces, that is, assume the equality of fluid pressure in the two phases, $P_1 = P_2 \equiv P(\mathbf{x},t)$, and capture multiphase effects through the relative permeability relationships. The latter are described by the Corey (1954) constitutive model. For the sake of concreteness, we take the subscripts $\ell = 1$ and 2 to stand for water and oil, respectively. Yet this formulation broadly applies to other multiphase flow processes such as contaminant migration, carbon sequestration, and geothermal flow.

We consider a square simulation domain D of size 150×150 m^2 shown in Figure 1. The Dirichlet conditions for both pressure p and water saturation S_1 are enforced along the vertical boundaries Γ_l and Γ_r: $P = 10.2$ MPa and $S_1 = 1.0$ on Γ_l and $P = 10.0$ MPa and $S_1 = 0.0$ on Γ_r. The remaining two boundaries, Γ_u and Γ_d, are impermeable to flow; that is, the homogeneous Neumann conditions are imposed on them. The simulations use a dummy third dimension to run in a general code. The domain size in this third dimension does not influence the solution for this test case as the problem is incompressible and the Dirichlet boundary conditions at Γ_l and Γ_r naturally scale with the volume of the cells. The initial conditions are $P = 10.1$ MPa and $S_1 = 0.0$.

All input parameters except for the intrinsic permeability field $k(\mathbf{x})$ are assumed to be known with certainty. As before, we consider $Y = \ln k$ to be Gaussian, with mean $\mu_Y(\mathbf{x}) = 0$ and exponential covariance

$C_Y(\mathbf{x}, \mathbf{y}) = \sigma_Y^2 \exp(-|\mathbf{x}-\mathbf{y}|/\lambda_Y)$ with variance $\sigma_Y^2 = 2.0$ and correlation length $\lambda_Y = 19.0$ m. The resulting CV for $k = \exp(Y)$ is

$$\mathrm{CV}(k) \equiv \frac{\sigma_k}{\langle k \rangle} = \frac{\sqrt{[\exp(\sigma_Y^2)-1]\exp(2\mu_Y + \sigma_Y^2)}}{\exp[\mu_Y + (\sigma_Y^2/2)]} = 2.53.$$

One realization of the log permeability field $Y(\mathbf{x},\omega)$ is shown in Figure 1. As in section 3.1, $Y(\mathbf{x},\omega)$ is approximated via a truncated KL expansion (17), with the number of terms in the expansion, $p = 31$, chosen to capture 95% of the energy of the field Y as determined by the square roots of the eigenvalues of its autocovariance.

The transport equation (19a) and pressure equation (19b) are discretized using a finite volume scheme in space and an implicit Euler scheme in time (Aziz & Settari, 1979). As this system of equations is highly nonlinear, at each time step we obtain the solution iteratively using the Newton-Raphson method, applying modified Appleyard saturation update damping (Appleyard et al., 1981) to improve convergence. That is, large updates in saturation values are chopped to a preset limit, $|S_{\ell,i}^{(\nu+1)} - S_{\ell,i}^{(\nu)}| \leq 0.3$ for each cell i and phase ℓ, where ν is the iteration number and i is the control volume index. To ensure convergence of both the flow (pressure) and transport (saturation) solutions, three convergence criteria are specified: normalized residual norm, maximum saturation update, and maximum relative pressure update:

$$\max_i \left| \Delta t \left(\frac{r_{\ell,i}}{\phi V_i} \right) \right| < \epsilon_1, \quad \max_i |S_{\ell,i}^{(\nu+1)} - S_{\ell,i}^{(\nu)}| < \epsilon_2, \quad \max_i \left| \frac{P_i^{(\nu+1)} - P_i^{(\nu)}}{P_i^{(\nu)}} \right| < \epsilon_3. \tag{20}$$

The tolerances are set to $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 10^{-2}$, and $\epsilon_3 = 10^{-3}$. Here V_i is the volume of cell i, and $r_{\ell,i}$ is the residual of the mass balance equation of phase ℓ for cell i. Note that the densities cancel and hence are not present in the normalization.

3.3. Upscaling of Material Properties

As MLMC relies on multiple grid resolutions to compute the CDF of a QoI, the medium's properties need to be consistent across levels. We achieve this with local single-phase upscaling, which is illustrated using a 2×2 block of the fine-scale (isotropic) permeability field as a concrete example (see Figure 2):

1. Denote the fine-scale permeability tensor by $\hat{\mathbf{k}}^{\mathrm{f}}$, and consider one realization of this random field.
2. Obtain the corresponding realization of its coarse-scale counterpart, $\hat{\mathbf{k}}^{\mathrm{c}}$, by clustering cells along the flow direction via length-weighted harmonic averaging and clustering cells perpendicular to the flow direction through area-weighted arithmetic averaging. The resulting tensor $\hat{\mathbf{k}}^{\mathrm{c}}$ is still diagonal but anisotropic.
3. Repeat this procedure as many times as needed to complete the sampling of the QoI at the coarser level.

The local single-phase upscaling strategy described above is cheap and effective. It can be replaced with more accurate yet more expensive techniques (Boso & Tartakovsky, 2018; Durlofsky, 2005). Regional or global multiphase upscaling can lead to a notable reduction in the total discretization error, that is, a smaller term $(\varepsilon_{\mathrm{dis}}^{\mathrm{MLsm}})^2$ in (9). As a result, advanced upscaling methods might improve convergence rates of MLMC at an additional computational cost.

Finally, the above upscaling of the permeability field defines the KL expansion (17) only on the finest discretization level. Alternatives to this method include formulating KL expansions at each level with a level-dependent number of terms (Gittelson et al., 2013; Teckentrup et al., 2013).

4. Simulation Results

We compare the performance of KDE-smoothed MLMC and HLSS-enhanced MLMC with and without KDE smoothing to that of fine-resolution MC and MLMC with and without polynomial smoothing on the single-phase (section 3.1) and two-phase (section 3.2) flow problems. We label these estimators below as MLMCsm (KDE), HLSS-MLMCsm, HLSS-MLMC, MC, MLMCsm (poly), and MLMC, respectively. We report the comparison in terms of their computational cost $\mathscr{C}$, averaged over $N_{\mathrm{real}} = 5$ independent runs, for error tolerances $\epsilon = 0.004$ and 0.0015 (single-phase flow) and $\epsilon = 0.06$, 0.02, and 0.04 (two-phase flow).

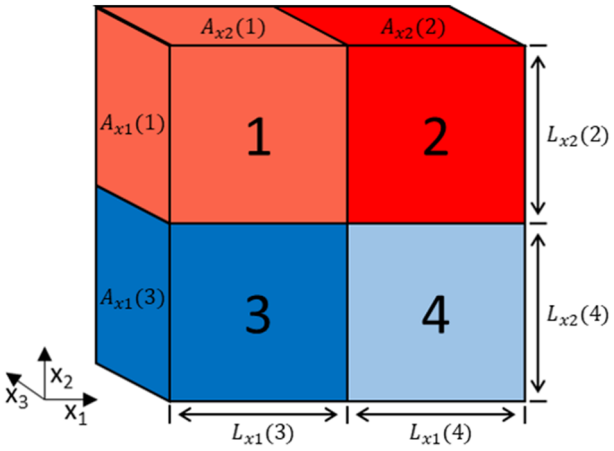

Figure 2. Coarsening procedure for a 2×2 block of the fine-scale permeability field $\hat{\mathbf{k}}^{\mathrm{f}}$. The red and blue colors correspond to regions of high and low permeability, respectively, in line with the color scheme used in Figure 1. When upscaling this block to a single cell, the resulting coarse-scale permeability tensor $\hat{\mathbf{k}}^{\mathrm{c}}$ is diagonal but anisotropic; its diagonal elements are $\hat{k}^{\mathrm{c}=}_{x_1x_1} AA\{HA[\hat{k}^f_{x_1x_1}(1), \hat{k}^f_{x_1x_1}(2)], HA[\hat{k}^f_{x_1x_1}(3), \hat{k}^f_{x_1x_1}(4)]\}$ and $\hat{k}^{\mathrm{c}=}_{x_2x_2} AA\{HA[\hat{k}^f_{x_2x_2}(1), \hat{k}^f_{x_2x_2}(3)], HA[\hat{k}^f_{x_2x_2}(2), \hat{k}^f_{x_2x_2}(4)]\}$. Here the notation (i), with $i = 1, \ldots, 4$, refers to the fine-scale cell with index i; and $HA[a,b]$ and $AA\{a,b\}$ represent a functional form of length-weighted harmonic averaging and area-weighted arithmetic averaging, respectively. With the present color scheme, we expect $\hat{k}^{\mathrm{c}}_{x_2x_2}$ to be low (closer to blue) and $\hat{k}^{\mathrm{c}}_{x_1x_1}$ to be of medium magnitude (between blue and red).

These values enable us to test our estimators with different numbers of discretization levels: $\epsilon = 0.004$ and 0.0015 typically yield $L_{\max} = 4$ and $L_{\max} = 5$, respectively, for single-phase flow, while $\epsilon = 0.06$, 0.04, and 0.02 result in $L_{\max} = 3$, 4, and 5, respectively, for two-phase flow. These tolerances lie in the *pre-asymptotic* regime typical for real-world subsurface flow simulations (Mukherjee, 2013), which is defined as $\epsilon > 10^{-4}$(D. Lu et al., 2016). The choice of tolerance $\mathcal{O}(10^{-2})$ rather than $\mathcal{O}(10^{-3})$ in the two-phase case is driven by both computational requirements and our aim to demonstrate that even at these higher tolerances our MLMC-based estimators outperform fine-resolution MC. All numerical experiments were performed on an Ubuntu system with 10 cores (20 hyperthreads) running at 4.20 GHz and having 64 GB of RAM.

The upscaling procedure in section 3.3 utilizes $N_{\max}$ or $N_{\max,\mathrm{strat}}$ samples of $\hat{k}$ on the finest level for the standard or HLSS-enhanced multilevel approaches, respectively. These numbers are chosen to be higher than the maximum number of samples of Q required at the coarsest ($l = 0$) level. For $N_{\max,\mathrm{strat}}$, we choose the multiple of 2 closest to $N_{\max}$; that is because HLSS-MLMC is initiated from a single sample at each level l and then doubles the number of samples at each sample size extension. The fine-scale permeability realizations are upscaled to their coarser-scale counterparts to compute the corresponding samples of Q on those coarser levels.

For MLMC, MLMCsm (KDE), and MLMCsm (poly), we first compute N_l^0 ($l = 0, \ldots, L_{\max}$) warm-up or *pilot* samples of Q to produce an initial estimate of the indicator functions variance; additional samples are then added as required to satisfy the sampling error tolerance. This warm-up procedure is an integral part of the overall sampling design and does not yield any overhead cost provided that oversampling is minimized. This is done by determining N_l^0 for each tolerance and each (standard) multilevel estimator separately through some initial trial runs; the resulting values are then employed for all N_{real} independent algorithm runs. The HLSS-MLMC and HLSS-MLMCsm algorithms are initiated with a single sample at each level l; then the sample size is extended through successive refinements of the univariate strata. For each value of ϵ, we

1. perform N_{real} runs of MLMC, yielding maximum levels $L^{\epsilon}_{\max,\,p}$ ($p = 1, \ldots, N_{\mathrm{real}}$), and denote the most frequently observed maximum level by $L^{\epsilon}_{\max}$;
2. at the end of the pth run, perform MC at level $L^{\epsilon}_{\max,\,p}$, reusing already computed samples from the MLMC run;
3. perform N_{real} runs of MLMCsm (KDE) with a computed smoothing parameter $\delta_{\mathrm{K},l}$ at each level l, fixing the maximum level for each run at $L^{\epsilon}_{\max}$;
4. perform N_{real} runs of MLMCsm (poly) with a computed smoothing parameter $\delta_{\mathrm{G},l}$ at each level l, fixing the maximum level for each run at $L^{\epsilon}_{\max}$;
5. perform N_{real} runs of HLSS-MLMC, with maximum level $L^{\epsilon}_{\max}$; and
6. perform N_{real} runs of HLSS-MLMCsm, with maximum level $L^{\epsilon}_{\max}$.

The reason for fixing the maximum level for MLMCsm (KDE and poly), HLSS-MLMC, and HLSS-MLMCsm to $L^{\epsilon}_{\max}$ was given in section 2.3.

For the single-phase flow problem, we set the parameter α defining the relative magnitudes of the different error sources (sampling error, bias, and, if applicable, smoothing error) to 0.5 (D. Lu et al., 2016). For the two-phase flow problem, we use $\alpha = 0.23$. Increasing the discretization tolerance enables us to satisfy the overall error tolerance with fewer levels. This significantly reduces the overall computational cost, since the two-phase simulations are very expensive at the finest levels.

Finally, we assume the PDFs of the QoIs to be at least 3 times continuously differentiable so that cubic splines can be used to interpolate the estimated CDF point values $\hat{F}_{h,\,M}(q)$ with $q \in \mathcal{S}_h$.

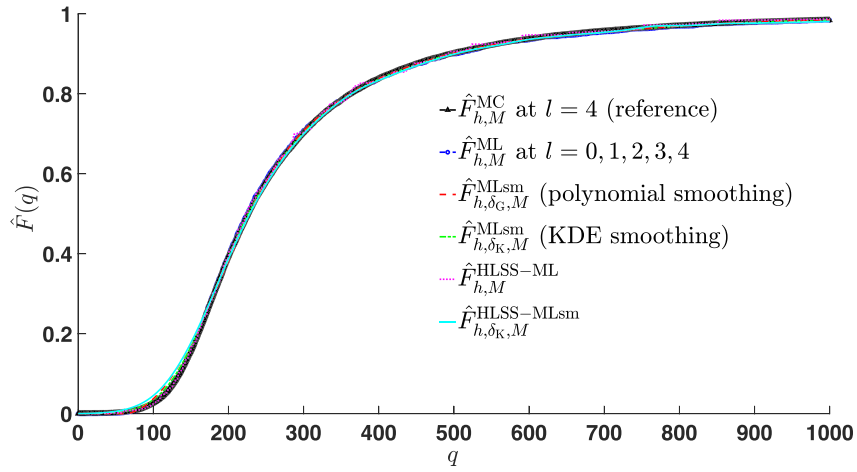

Figure 3. Estimated CDF of the QoI Q in the single-phase flow problem obtained via MLMC, MLMCsm (KDE and poly), HLSS-MLMC, and HLSS-MLMCsm, for tolerance $\epsilon = 0.004$. The MC estimator computed on the finest level is shown for reference.

4.1. Single-Phase Flow

Our QoI in this example is the average pressure (hydraulic head) in a sample,

$$Q = \frac{1}{2}\int_0^2 h(x,\, t = 0.2)dx. \tag{21}$$

The goal is to estimate the CDF of Q, $F(q)$, on the interval $0 \le q \le 1{,}000$ using $S + 1 = 1{,}001$ interpolation points. The flow domain $D = [0,2]$ is discretized with a hierarchy of spatial grids $\mathscr{T}_{M_l}$ indexed by $l = 0, \ldots, L^{\epsilon}_{\max}$, where $M_l = 2M_{l-1}$, $M_0 = 100$, and $L^{\epsilon}_{\max} = 4$ for tolerance $\epsilon = 0.004$ and 5 for $\epsilon = 0.0015$. For MLMC, MLMCsm (KDE), and MLMCsm (poly), we generate $N_{\max} = 5 \cdot 10^5$ samples of $\hat{k}$ at $l = 5$ (the finest level considered); for HLSS-MLMC and HLSS-MLMCsm, we choose $N_{\max,\mathrm{strat}} = 2^{19} = 524{,}288$.

Figure 3 shows the CDF approximations computed using a single run of the various multilevel estimators for $\epsilon = 0.004$, along with a fine-grid MC estimator for reference. The largest discrepancy with fine-grid MC can be seen near the distribution's left tail for MLMCsm and HLSS-MLMCsm. This illustrates the limitations of the currently used L^{∞} norm for expressing an estimator's MSE, which does not allow tight control over the error in specific regions of the CDF such as its tails. In future iterations of our algorithms, we may therefore consider switching to the L^1 or L^2 norm.

Figure 4 collates the computational costs $\mathscr{C}$ of all the estimators at the two tolerance levels. For $\epsilon = 0.004$, $\mathscr{C}(\hat{F}^{\mathrm{ML}}_{h,M})$ is less than half of $\mathscr{C}(\hat{F}^{\mathrm{MC}}_{h,M})$, the cost of MC performed at the finest level $L^{\epsilon}_{\max} = 4$, where $M \equiv M_{L^{\epsilon}_{\max}}$. The difference in cost increases for the lower tolerance of 0.0015 since $L^{\epsilon}_{\max}$ increases from 4 to 5 and hence the fine-scale MC simulations become more expensive. Applying KDE-based smoothing to the indicator function $\mathscr{I}_n(Y_l)$ at each level $l = 0, \ldots, L^{\epsilon}_{\max}$, and using a bandwidth $\delta_{\mathrm{K},l}$ computed via the procedure in Appendix A2, yields about a factor of 3 savings for $\epsilon = 0.004$ and nearly half an order-of-magnitude speedup for $\epsilon = 0.0015$. Smoothing based on a third-degree polynomial consistently performs more poorly than its KDE-based counterpart, and increasing the polynomial order reduces the efficiency even further. For example, at $\epsilon = 0.004$, the N_{real}-averaged cost using a ninth-degree polynomial is around 93 s, while that using a third-degree polynomial is only about 72 s. We conclude that kernel-based smoothing outperforms the polynomial-based techniques.

The different degrees of computational savings obtained by MLMC and MLMCsm compared to MC can be explained by comparing the evolution of $\tilde{V}[\mathscr{I}_n(Q_{M_l})]$ with level to that of $\tilde{V}[\mathscr{I}_n(Y_l)]$ and $\tilde{V}[g_n(Y_l)]$, where $\tilde{V}$ denotes a sample estimate of $\mathbb{V}$ and g_n is a smooth approximation of $\mathscr{I}_n$. For a single run with $\epsilon = 0.004$, Figure 5a demonstrates that while $\tilde{V}[\mathscr{I}_n(Q_{M_l})]$ remains approximately constant as the spatial resolution increases, $\tilde{V}[\mathscr{I}_n(Y_l)]$ and $\tilde{V}[g_n(Y_l)]$ decay as the spatial mesh is refined. That results in fewer required

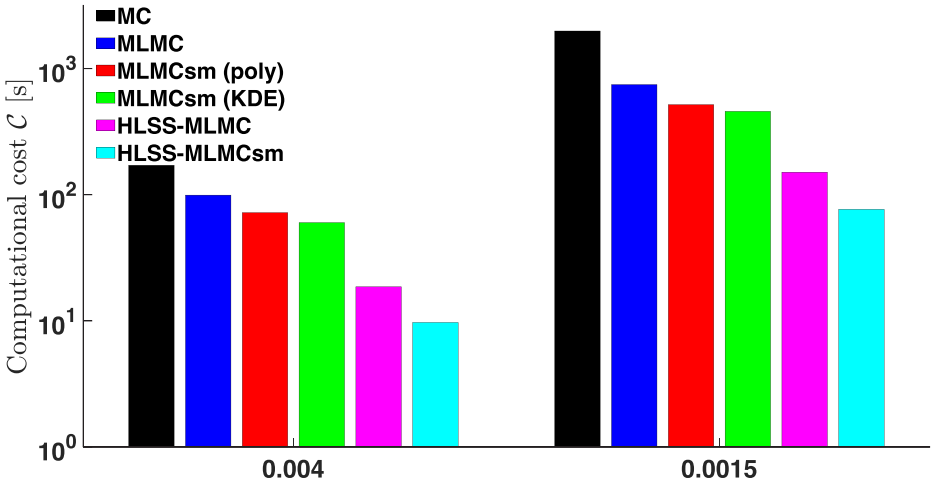

Figure 4. Computational cost (in seconds) of the standard and HLSS-enhanced multilevel estimators and their fine-resolution MC counterpart for the single-phase flow test bed at tolerances $\epsilon = 0.004$ (left) and 0.0015 (right).

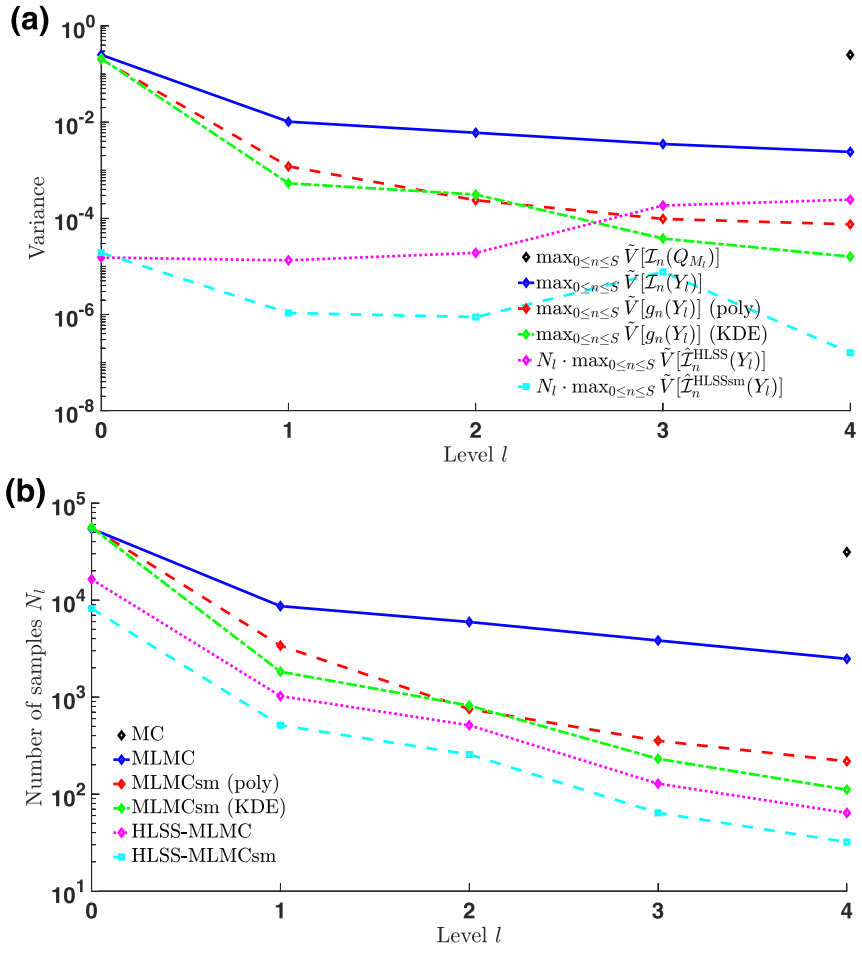

Figure 5. Evolution of the variance (a) and number of samples (b) with level for a single run of the single-phase flow problem and $\epsilon = 0.004$.

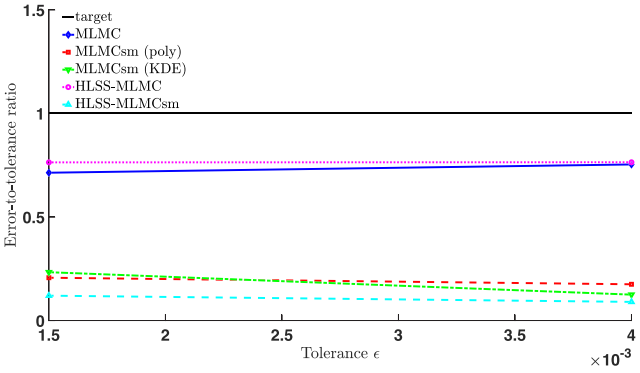

Figure 6. Values of the ratio $\varepsilon_{\text{est}}/\epsilon$ for MLMC, the ratio $\varepsilon_{\text{sam}}/\epsilon_{\text{sam}}$ for HLSS-MLMC, and the ratio $(\varepsilon_{\text{sam}} + \epsilon_{\text{smooth}})/(\epsilon_{\text{sam}} + \epsilon_{\text{smooth}})$ for MLMCsm (poly and KDE) and HLSS-MLMCsm, at all considered tolerances ϵ for the single-phase flow problem.

samples at higher levels of discretization (see Figure 5b). The faster decay of $\tilde{V}[g_n(Y_l)]$ compared to $\tilde{V}[\mathscr{I}_n(Y_l)]$ makes MLMCsm more efficient than its non-smoothed counterpart, with the largest effects seen for KDE-based smoothing.

Next, we investigate the speedups achieved by HLSS-MLMC and HLSS-MLMCsm and compare them to the gains from MLMCsm (KDE). Given the different sampling architecture of HLSS, where at each level we start from a single sample and then extend the sample size repeatedly by a constant factor (in our case, 2), the procedure described in Appendix C2 does not allow for a straightforward comparison. Instead, we modify it as follows.

1. For each independent realization $p = 1, \dots, N_{\text{real}}$ of HLSS-MLMC/HLSS-MLMCsm, pick N_l for each level l to be the multiple of 2 closest to its counterpart in MLMCsm (KDE).
2. Tweak this initial sampling design until the total estimator variance across all levels is as close as possible, but still below, the mean square sampling error tolerance, ensuring a monotonic decay in N_l with increasing l in the process.
3. Compare the resulting average cost, $\mathscr{C}(\hat{F}_{h,M}^{\text{HLSS -ML}})$ or $\mathscr{C}(\hat{F}_{h,M}^{\text{HLSS -MLsm}})$, to that of MLMCsm (KDE).

Figure 4 shows that Latinized stratification produces even higher efficiency gains than KDE-based smoothing: The speedup is nearly an order of magnitude for $\epsilon = 0.004$ and exceeds an order of magnitude for $\epsilon = 0.0015$. Combining HLSS with KDE-based smoothing further increases the efficiency by about a factor of 2, as the additional variance reduction from the smoothing reduces the number of samples in some of the levels and still allows the estimator to satisfy the sampling error tolerance.

To define an equivalent to $\max_{0 \le n \le S} \tilde{V}[\mathscr{I}_n(Y_l)]$ for HLSS-MLMC (a similar reasoning applies to HLSS-MLMCsm), we consider the variance contribution at each level l for MLMC, $N_l^{-1} \max_{0 \le n \le S} \mathbb{V}[\mathscr{I}_n(Y_l)]$, that is, consider the quantity $N_l \cdot \max_{0 \le n \le S} \mathbb{V}[\hat{\mathscr{I}}_n^{\text{HLSS}}(Y_l)]$. The use of HLSS yields a higher variance reduction compared to MLMC and MLMCsm at the coarser levels (Figure 5a). Thus, it is responsible for the higher computational efficiency of the HLSS-enhanced multilevel estimators, with MLMCsm displaying the largest variance reduction. The rise in variance toward the finer grids for HLSS-MLMC should be interpreted with caution because the numbers of samples computed on those fine-resolution levels are small and, hence, the corresponding sample estimates of $\mathbb{V}[\hat{\mathscr{I}}_n^{\text{HLSS}}(Y_l)]$ become less reliable. Moreover, the equivalency defined above is only an approximation.

Figure 5b shows the breakdown of the numbers of samples on the various levels, for a single run and $\epsilon = 0.004$, for the HLSS-MLMC and HLSS-MLMCsm algorithms. The lower numbers of samples for these estimators compared to their standard multilevel counterparts, particularly on the coarser levels, reflect the significant variance reduction achieved through the Latinized stratification of the input sample space.

Finally, to demonstrate that our multilevel estimators satisfy the required error tolerance, one could compute the ratio $\varepsilon_{\text{est}}/\epsilon$, averaged over N_{real} runs. However, at the finest levels, the low number of samples makes the sample estimate of the root mean square discretization error, ε_{dis}, less reliable (see section 2.3). Figure 6 displays the ratio of *total* root MSE to *total* tolerance only for MLMC; for the other multilevel variants, we remove the discretization portion from the total error and compare the resulting error to the corresponding fraction of the total tolerance. We find this ratio to be less than one for all our multilevel estimators, both at $\epsilon = 0.004$ and 0.0015.

We conclude that for our 1-D problem, both KDE-based smoothing of the indicator function and Latinized stratification at each level improve upon the efficiency of a non-smoothed multilevel estimator, with the latter of the two yielding the highest cost reduction, that is, about an order of magnitude compared to fine-resolution MC for $\mathcal{O}(10^{-3})$ error tolerances.

4.2. Two-Phase Flow

For the problem described in section 3.2, Figure 7 shows the water saturation, S_1, and pressure, P, at breakthrough time ($q = 2{,}856$ days) for one realization of the permeability field k at a spatial resolution of 128 ×

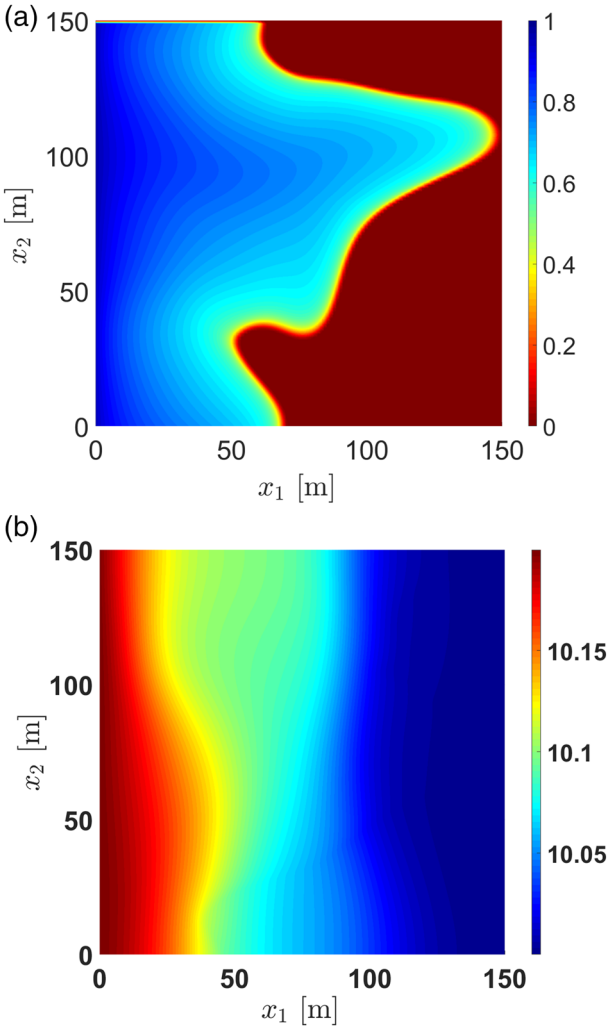

Figure 7. Water saturation S_1 (a) and pressure P in MPa (b) at breakthrough time $q = 2{,}856$ days for the permeability field k shown in Figure 1.

128 cells (i.e., at level $l = 4$). The QoI Q is the time of water breakthrough at the right boundary, Γ_r. We estimate its CDF $F(q)$ on the interval $0 \le q \le$ 10,950 days (~30 years) using $S + 1 = 10{,}951$ interpolation points. This interval was chosen based on our observation that over 99% of sampled breakthrough times fell below 10,950 days; we set $Q = 10{,}950$ days for runs where breakthrough did not occur within that time frame.

As the early time process is highly nonlinear, we choose the first five time steps to be less than or equal to 50 days to ensure convergence of the Newton-Raphson iterations. The time step is then fixed to 50 days for the remainder of the simulation. If a certain time step does not converge,the time step is cut in half and taken twice. By doing so, differences in temporal discretization are minimized. The domain D is discretized using a hierarchy of spatial grids $\mathcal{T}_{M_l}$ with $l = 0, \ldots, L^{\epsilon}_{\max}$, where $M_l = 4M_{l-1}$, $M_0 = 64$ (an 8×8 grid), and $L^{\epsilon}_{\max} = 3$, 4, and 5 for tolerances $\epsilon = 0.06$, 0.04, and 0.02, respectively. Following the reasoning of section 4.1, we set $N_{\max} = 50{,}000$ and $N_{\max,\text{strat}} = 2^{15} = 32{,}768$.

Figure 8 displays the CDF approximations obtained via the different MLMC estimators at $\epsilon = 0.04$, along with a fine-grid reference MC estimator. The biggest difference between MLMC and fine-grid MC occurs for HLSS-MLMC, which yields a more noisy approximation. Recall that the tolerance here is an order of magnitude higher than that used in Figure 3 (i.e., 0.04 vs. 0.004). Lowering ϵ to 0.02 reduces the discrepancy with the reference MC estimate, as expected. Moreover, KDE-based smoothing of the indicator function dampens this noise. The use of the L^1 or L^2 norm instead of the L^∞ norm might improve error control and, hence, reduce noise in the CDF approximation.

Figure 9 shows that (non-smoothed) MLMC leads to a modest reduction in computational cost compared to fine-resolution MC, amounting to nearly half an order of magnitude at the lowest tolerance, $\epsilon = 0.02$, with the added discretization levels again increasing the discrepancy between both estimators' performances. KDE-based smoothing of the indicator function $\mathcal{I}_n(Y_l)$ at each level yields significantly higher cost savings. For $\epsilon = 0.06$, the speedup is about an order of magnitude and remains approximately constant when the tolerance is decreased to 0.04. Further reduction in the tolerance (to 0.02) increases the speedup to almost 2 orders of magnitude. On the other hand, smoothing with a third-degree polynomial yields almost no additional cost savings compared to its non-smoothed counterpart. This result is in line with previous findings (D. Lu et al., 2016), according to which fine-resolution MC of a related two-phase flow problem could be faster than MLMCsm (poly) at tolerances of $\mathcal{O}(10^{-2})$. Increasing the polynomial degree to 9 again decreases performance. We conclude that KDE-based smoothing offers a major advantage over polynomial techniques in realistic multiphase flow setups at tolerances relevant to engineering applications.

Next, we consider HLSS-MLMC and HLSS-MLMCsm and again follow the procedure described in section 4.1 to determine the appropriate numbers of samples N_l in each level l for these estimators. Figure 9 demonstrates that HLSS-MLMC is more efficient than MLMCsm (KDE) for $\epsilon = 0.06$ but less efficient than the latter for $\epsilon = 0.04$ and 0.02. However, the cost of both methods is on the same order of magnitude for all tolerances; the exact values of $\mathscr{C}$ should be interpreted with caution because they are an average over a finite number of independent runs ($N_{\text{real}} = 5$) and hence will vary with N_{real} to some extent. Adding KDE-based smoothing of the indicator function improves the performance of the HLSS-enhanced estimator, making it more efficient than MLMCsm (KDE) across all tolerances.

Figure 10a illustrates the variance reduction from KDE smoothing and Latinized stratification responsible for the large cost savings achieved with MLMCsm (KDE) and HLSS-MLMC, respectively. As in the

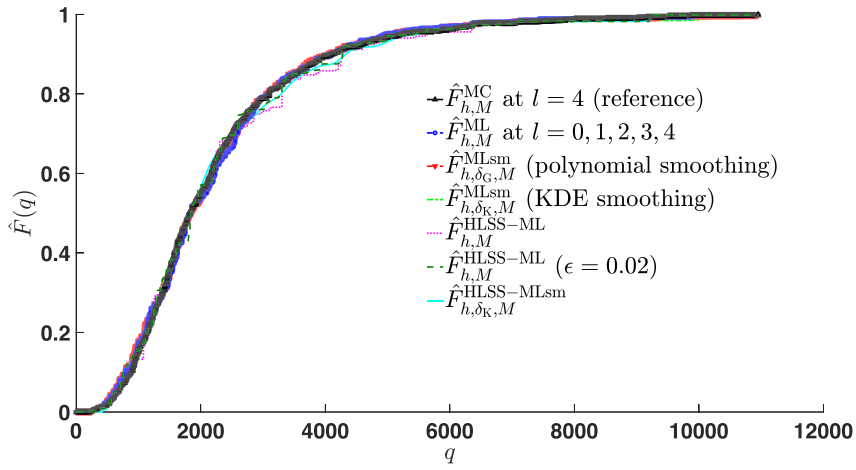

Figure 8. Estimated CDF of the breakthrough time Q (in days) in the two-phase flow problem obtained via MLMC, MLMCsm (KDE and polynomial), HLSS-MLMC, and HLSS-MLMCsm, for tolerance $\epsilon = 0.04$. The MC estimator computed on the finest level is shown for reference.

single-phase flow problem, Latinized stratification provides more variance reduction at the coarser levels, reducing the numbers of samples at those levels (Figure 10b). Polynomial smoothing follows the trend of its non-smoothed counterpart and is only slightly more efficient than the latter (it satisfies the discretization error tolerance with only four levels). KDE-based smoothing achieves a much larger variance reduction and associated decrease in the number of samples with level. This causes the run shown in Figure 10 to carry a lower computational cost than its polynomial-smoothed counterpart despite employing one extra level. Similar behavior was observed for the other runs, leading to a lower average cost for MLMCsm (KDE) compared to MLMCsm (poly).

The above results suggest that, for two-phase flow, our implementation of MLMC provides significant computational savings even at relatively high tolerances. Specifically, it yields up to nearly 2 orders of magnitude speedup compared to MC when KDE-based smoothing is applied to the indicator function or LSS is employed at each level. At tolerances of $\mathcal{O}(10^{-3})$ and $\mathcal{O}(10^{-4})$, which fall within the pre-asymptotic regime (Mukherjee, 2013), 3 or more orders of magnitude in cost savings could be achieved by these methods.

While our numerical tests are done with efficient MATLAB® codes, lower speedups may be observed when performed with highly efficient commercial software that scales better with the number of grid cells in the

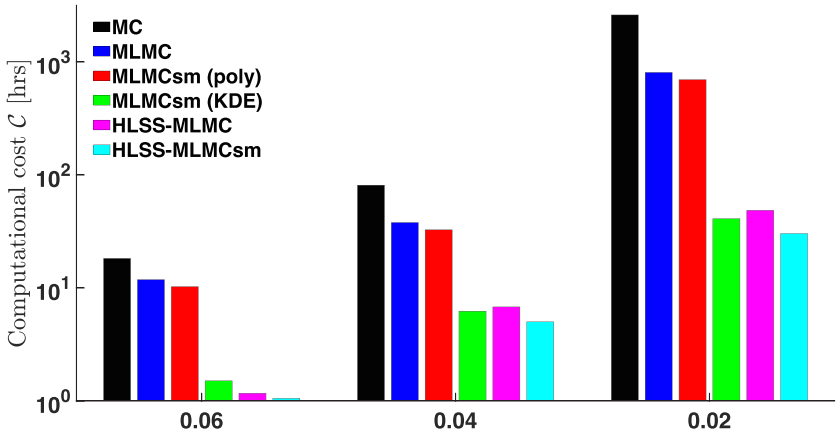

Figure 9. Computational cost (in hours) of the standard and HLSS-enhanced multilevel estimators and their fine-resolution MC counterpart for the two-phase flow test bed at tolerances $\epsilon = 0.06$, 0.04, and 0.02 (from left to right).

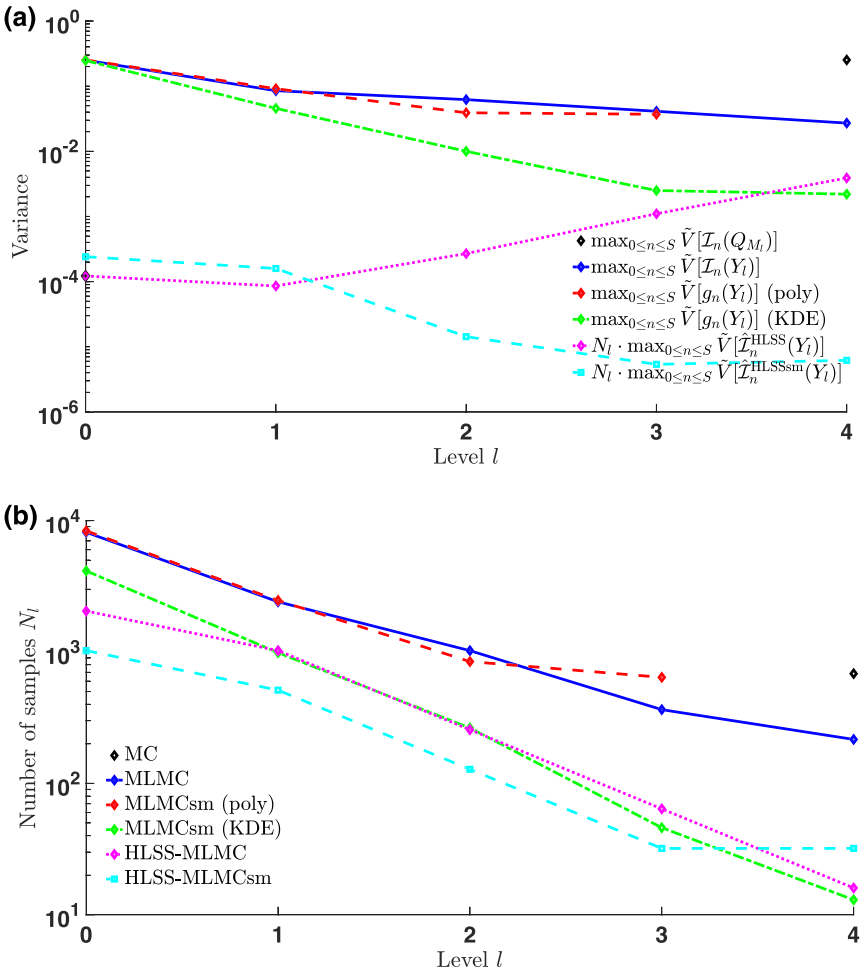

Figure 10. Evolution of the variance (a) and the number of samples (b) with level for a single run of the two-phase flow problem and $\varepsilon = 0.04$.

domain. Nevertheless, we still expect our smoothed standard MLMC and HLSS-MLMC algorithms to achieve order-of-magnitude computational savings at tolerances of 0.02 or lower on commercial platforms.

5. Conclusions

We proposed novel MLMC algorithms to efficiently estimate cumulative probability distributions (exceedance probabilities) of QoIs. The methods either employ kernel-based smoothing of the indicator function within a standard multilevel approach or replace standard MC at each level of discretization with a sampling design that combines LHS with stratification known as HLSS. We assess the performance of the new estimators, respectively, referred to as MLMCsm (KDE) and HLSS-MLMC, on single- and two-phase flow problems. In both cases, the source of parametric uncertainty is a spatially varying permeability that has a lognormal distribution and an exponential autocovariance for its logarithm.

Our study yields the following major conclusions:

1. For 1-D single-phase flow, MLMCsm (KDE) and HLSS-MLMC yield computational cost savings of, respectively, about a half and a full order of magnitude compared to MC applied at the finest MLMC level for error tolerances of $\mathcal{O}(10^{-3})$.
2. For 2-D two-phase flow, MLMCsm (KDE) and HLSS-MLMC yield an even larger speedup compared to MC applied at the finest MLMC level for error tolerances of $\mathcal{O}(10^{-2})$. Specifically, we find computational time savings of up to nearly 2 orders of magnitude with our MATLAB® simulator.

3. KDE-based smoothing consistently outperforms the polynomial-based techniques regardless of polynomial degree for both 1-D single-phase and 2-D two-phase flows, with the biggest discrepancy occurring for the latter problem where polynomial smoothing barely yields additional cost savings compared to its non-smoothed counterpart.
4. Latinized stratification produces a larger variance reduction at the coarser levels compared to KDE-based smoothing of the indicator function.
5. Combining KDE-based indicator function smoothing with Latinized stratification at each level yields the most efficient estimator.

For non-smoothed MLMC, polynomial-smoothed MLMC, KDE-smoothed MLMC, and our HLSS-MLMC algorithm, the construction of an approximate CDF $\hat{F}_{h,M}$ via (8) can lead to a decreasing sequence of values $\hat{F}_{h,M}(q_n)$ with $q_0 < \ldots < q_S$ (Giles et al., 2017). Even when this sequence is non-decreasing, the resulting piecewise polynomial interpolant $\hat{F}_{h,M}(q)$ is not necessarily non-decreasing. One could perform a two-stage post-processing of the $\hat{F}_{h,M}(q_n)$ to ensure the resulting point values are non-decreasing, so that the piecewise polynomial interpolation of these modified values yields a monotonic CDF (Giles et al., 2017). We applied this post-processing step only to produce the CDF figures in section 4 because it does not affect the performance of the MLMC estimators or any subsequent uncertainty quantification analysis performed with the resulting CDF approximations. In future work, we will embed such procedures in the algorithm itself rather than applying them as a post-processing step.

The current strategy to compare the performance of MLMCsm (KDE) and HLSS-MLMC is rather ad hoc, and we plan to replace it by a more automated approach to make this comparison both easier and more rigorous. This may involve additional tweaks to the HLSS algorithm, which was originally designed to run on a single discretization level and could be optimized further within the multilevel context. Another direction for future research concerns a more thorough characterization of the distribution's tails. The current L^∞ norm-based approach is not optimized for this task, and the L^1 or L^2 norm may be a more suitable choice.

Multilevel methods belong to the wider class of "multifidelity" approaches which involve a combination of models with varying degrees of fidelity (Müller et al., 2014; O'Malley et al., 2018; Peherstorfer et al., 2016). The maximum variance reduction (and, hence, speedup) MLMC can achieve depends on the degree of correlation between the levels (Gorodetsky et al., 2020). For complex physics, where refining the grid actually resolves more features, this correlation will be lower, and so will be the variance reduction achieved by going from coarser to finer grids. In the problems discussed here, the correlation between levels is sufficiently high for MLMC to achieve a notable variance reduction.

Appendix A: Standard MLMC

A1. Standard MLMC Without Smoothing

The MC estimator for $\tau_{n,M} \equiv \mathbb{E}[\mathcal{I}_n(Q_M)]$ based on N_{MC} independent samples of Q_M is defined by

$$\hat{\mathcal{I}}^{\mathrm{MC}}_{n,M} = \frac{1}{N_{\mathrm{MC}}} \sum_{j=1}^{N_{\mathrm{MC}}} \mathcal{I}_n(Q_M^{(j)}), \tag{A1}$$

where $Q_M^{(j)}$ is the jth sample of Q_M and M is the number of grid cells in the spatial mesh $\mathcal{T}_M$. Its variance is

$$\mathbb{V}[\hat{\mathcal{I}}^{\mathrm{MC}}_{n,M}] = \frac{1}{N_{\mathrm{MC}}} \mathbb{V}[\mathcal{I}_n(Q_M)]. \tag{A2}$$

Rather than considering a single resolution (i.e., value of M), we can look at a sequence of approximations Q_{M_l} ($l = 0, \ldots, L_{\max}$) of Q associated with corresponding discrete meshes $\mathcal{T}_{M_l}$ (Heinrich, 1998, 2001). Here M_l denotes the number of grid cells in mesh $\mathcal{T}_{M_l}$, such that $M_{l-1} = 2^{-d} M_l$ where d is the spatial dimension, and $M_{L_{\max}} \equiv M$. This last condition enables a performance comparison with the MC estimator in (A1). The use of multiple spatial resolutions allows one to generate cheap-to-compute samples on a coarse mesh and then gradually correct the resulting estimate of $F_{h,M}$ by sampling on finer grids, where generating a realization is more computationally expensive. Then $\tau_{n,M}$ can be rewritten as a telescopic sum

$$\tau_{n,M} = \mathbb{E}[\mathcal{I}_n(Q_{M_0})] + \sum_{l=1}^{L_{\max}} \mathbb{E}[\mathcal{I}_n(Q_{M_l}) - \mathcal{I}_n(Q_{M_{l-1}})] \equiv \sum_{l=0}^{L_{\max}} \mathbb{E}[\mathcal{I}_n(Y_l)], \tag{A3a}$$

where $\mathcal{I}_n(Y_l)$ with $l = 0, \ldots, L_{\max}$ has the form

$$\mathcal{I}_n(Y_l) = \begin{cases} \mathcal{I}_n(Q_{M_l}) - \mathcal{I}_n(Q_{M_{l-1}}) & 1 \le l \le L_{\max} \\ \mathcal{I}_n(Q_{M_l}) & l = 0. \end{cases} \tag{A3b}$$

This procedure yields the following MLMC estimator for $\tau_{n,M}$

$$\hat{\mathcal{I}}^{\mathrm{ML}}_{n,M} = \sum_{l=0}^{L_{\max}} \hat{\mathcal{I}}^{\mathrm{MC}}_{n}(Y_l) = \sum_{l=0}^{L_{\max}} \frac{1}{N_l} \sum_{j=1}^{N_l} \mathcal{I}_n(Y_l^{(j)}), \tag{A4}$$

which has a variance

$$\mathbb{V}[\hat{\mathcal{I}}^{\mathrm{ML}}_{n,M}] = \sum_{l=0}^{L_{\max}} \frac{1}{N_l} \mathbb{V}[\mathcal{I}_n(Y_l)]. \tag{A5}$$

(We employ the shorthand notation ML to refer to MLMC in estimator expressions.) MLMC achieves variance reduction through the fact that $\mathbb{V}[\mathcal{I}_n(Y_l)]$ decreases with l. That is in contrast to $\mathbb{V}[\mathcal{I}_n(Q_{M_l})]$, which remains approximately constant for different values of l. This means that MLMC can achieve the same sampling error as MC performed at its finest level by computing fewer samples N_l at higher l, where sampling is more costly. If $\mathbb{V}[\mathcal{I}_n(Y_l)]$ decreases fast enough with l, this can make the overall computational cost of the estimator $\hat{\mathcal{I}}^{\mathrm{ML}}_{n,M}$ lower than that of its MC counterpart, $\hat{\mathcal{I}}^{\mathrm{MC}}_{n,M}$.

It follows from (7) and (A5) that the MSE of the non-smoothed MLMC estimator $\hat{F}^{\mathrm{ML}}_{h,M}$ for F_h satisfies the inequality

$$\begin{aligned} \underbrace{\mathbb{E}[||F_h - \hat{F}^{\mathrm{ML}}_{h,M})||^2_\infty]}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{est}})^2} &\le \mathbb{E}[||\hat{F}^{\mathrm{ML}}_{h,M} - \mathbb{E}[\hat{F}^{\mathrm{ML}}_{h,M}]||^2_\infty] + ||F_{h,M} - F_h||^2_\infty \\ &\le \underbrace{\max_{0 \le n \le S} \sum_{l=0}^{L_{\max}} N_l^{-1} \mathbb{V}[\mathcal{I}_n(Y_l)]}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{sam}})^2} + \underbrace{\max_{0 \le n \le S} |\mathbb{E}[\mathcal{I}_n(Q_{M_{L_{\max}}}) - \mathcal{I}_n(Q)]|^2}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{dis}})^2}. \end{aligned} \tag{A6}$$

To achieve a root mean square error (RMSE) of at most ϵ, we introduce a tunable parameter $\alpha \in (0,1)$ and choose $\varepsilon^{\mathrm{ML}}_{\mathrm{sam}} \le \sqrt{\alpha}\,\epsilon$ and $\varepsilon^{\mathrm{ML}}_{\mathrm{dis}} \le \sqrt{1-\alpha}\,\epsilon$. The value of α determines the relative magnitudes of the allowable sampling error and discretization error (bias), which, along with an optimal choice of the number of samples at each level (see Appendix C1), aids in minimizing the total computational cost for a given tolerance ϵ. We estimate the bias via the triangle inequality,

$$\max_{0 \le n \le S} |\mathbb{E}[\mathcal{I}_n(Y_{L_{\max}})| \approx \max_{0 \le n \le S} |\mathbb{E}[\mathcal{I}_n(Q_{M_{L_{\max}}}) - \mathcal{I}_n(Q)]|. \tag{A7}$$

Hence, the maximum level $L_{\max}$ of an MLMC simulation, for a given tolerance ϵ, is determined by verifying that the condition $\max_{0 \le n \le S} |\mathbb{E}[\mathcal{I}_n(Y_L)| \le \sqrt{1-\alpha}\,\epsilon$ is satisfied for the current level L. If it is not, a new level is added; otherwise, $L_{\max} = L$.

A2. Standard MLMC With Polynomial-Based Smoothing

The jump discontinuity in the indicator function may lead to a slow decay of $\mathbb{V}[\mathcal{I}_n(Y_l)]$, causing MLMC to become slower than MC for sufficiently large values of the error tolerance ϵ (D. Lu et al., 2016). To accelerate the variance decay and thereby improve the computational efficiency of MLMC, a sigmoid-type smoothing function can be used to remove the singularity in $\mathcal{I}_n(Y_l)$. For example, polynomial-based smoothing (Giles et al., 2015) has been used to accelerate MLMC simulations in reservoir engineering (D. Lu et al., 2016). Polynomial smoothing requires the user to specify both an appropriate smoothing parameter or

"bandwidth" $\delta_{G,l}$ at each level l, which defines the distance over which the discontinuity in $\mathcal{I}_n(Q_{M_l})$ is smeared out, and the polynomial degree p of, at most, $r+1$ with r the number of times the (unknown) PDF $f(q)$ is continuously differentiable. Then $\mathcal{I}_n(Y_l)$ is replaced by $g_n(Y_l)$ defined by

$$g_n(Y_l) = \begin{cases} g_G^p\left(\frac{Q_{M_l} - q_n}{\delta_{G,l}}\right) - g_G^p\left(\frac{Q_{M_{l-1}} - q_n}{\delta_{G,l}}\right) & 1 \le l \le L_{\max} \\ g_G^p\left(\frac{Q_{M_l} - q_n}{\delta_{G,l}}\right) & l = 0, \end{cases} \tag{A8}$$

where g_G^p is a smoothing polynomial of degree p computed through the procedure described in Giles et al. (2015).

The MSE of the polynomial-smoothed MLMC estimator $\hat{F}^{\mathrm{MLsm}}_{h,\delta_G,M}$ for F_h is bounded by

$$\underbrace{\mathbb{E}[||F_h - \hat{F}^{\mathrm{MLsm}}_{h,\delta_G,M})||^2_\infty]}_{(\varepsilon^{\mathrm{MLsm}}_{\mathrm{est}})^2} \le \underbrace{\mathbb{E}[||\hat{F}^{\mathrm{smML}}_{h,\delta_G,M} - \mathbb{E}[\hat{F}^{\mathrm{ML}}_{h,\delta_G,M}]||^2_\infty]}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{sam}})^2} + \underbrace{||F_{h,M} - F_h||^2_\infty}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{dis}})^2} + \underbrace{\mathbb{E}[||\hat{F}^{\mathrm{ML}}_{h,\delta_G,M}] - F_{h,M}||^2_\infty}_{(\varepsilon^{\mathrm{ML}}_{\mathrm{sm}})^2}. \tag{A9}$$

Compared to (A6), (A9) contains an additional term $(\varepsilon^{\mathrm{ML}}_{\mathrm{sm}})^2$, which is the (mean square) smoothing error. To achieve an RMSE of at most ϵ, we may choose $\varepsilon^{\mathrm{ML}}_{\mathrm{dis}} \le \sqrt{1-\alpha}\,\epsilon$, the same as for the non-smoothed MLMC estimator, such that $\hat{F}^{\mathrm{ML}}_{h,\delta_G,M}$ satisfies the discretization error tolerance for the same number of levels as its non-smoothed counterpart. Choosing $\varepsilon^{\mathrm{ML}}_{\mathrm{sam}} \le \sqrt{\alpha/2}\,\epsilon$ and $\varepsilon^{\mathrm{ML}}_{\mathrm{sm}} \le \sqrt{\alpha/2}\,\epsilon$ then enables us to satisfy the total error tolerance.

Given the above constraint on $\varepsilon^{\mathrm{ML}}_{\mathrm{sm}}$, the optimal value for the bandwidth $\delta_{G,l}$ at each level l is such that $\varepsilon^{\mathrm{ML}}_{\mathrm{sm}}$ is as close as possible, but still smaller than, $\sqrt{\alpha/2}\,\epsilon$. Choosing a larger value for $\delta_{G,l}$ yields a bigger value for $\varepsilon^{\mathrm{ML}}_{\mathrm{sm}}$, which does not allow us to satisfy the smoothing error tolerance. Choosing a smaller value for $\delta_{G,l}$ yields a lower reduction in $\mathbb{V}[g_n(Y_l)]$ and therefore a less efficient algorithm. A possible strategy to find the optimal $\delta_{G,l}$ consists of the following steps (D. Lu et al., 2016).

1. Start with level $l = 0$.
2. Estimate $\delta_{G,l,n}$ for each interpolation point q_n in $S_h = \{q_n, n = 0, \ldots, S\}$ by solving

$$\frac{1}{N_l^0}\left|\sum_{j=1}^{N_l^0}\left[g_G\left(\frac{Q^{(j)}_{M_l} - q_n}{\delta_{G,l,n}}\right) - \mathcal{I}_n\left(Q^{(j)}_{M_l}\right)\right]\right| = \sqrt{\frac{\alpha}{2}}\,\epsilon \tag{A10}$$

based on a set of initial samples $\{Q^{(j)}_{M_l}\}_{j=1}^{N_l^0}$.
3. Define the smoothing parameter $\delta_{G,l}$ as

$$\delta_{G,l} = \max_{0 \le n \le S} \delta_{G,l,n}. \tag{A11}$$

4. Repeat Steps 2 and 3 for each new level l.

Appendix B: Stratified Sampling and Latin Hypercube Sampling

B1. Stratified Sampling

In stratified sampling (SS), the sample space $\mathbf{\Omega}$ of the random inputs $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^T$ is divided into r mutually exclusive and exhaustive subsets or *strata* $\mathcal{D}_k$ ($k = 1, \ldots, r$). All N_{SS} samples, $\mathbf{s}_j = [s_{j\,1}, \ldots, s_{jp}]$ with

$j = 1, \ldots, N_{SS}$, are generated by randomly drawing N_k samples within the strata $\mathscr{D}_k$, with $\sum_{k=1}^{r} N_k = N_{SS}$, according to

$$s_{ji}^{(k)} = F_{\xi_i}^{-1}(U_{ik}^{j}), \qquad i = 1, \ldots, p. \tag{B1}$$

Here F_{ξ_i} is the CDF of ξ_i and U_{ik}^{j} are independent, uniformly distributed samples on $[\zeta_{ik}^{low} = F_{\xi_i}(\xi_{ik}^{low}), \zeta_{ik}^{upp} = F_{\xi_i}(\xi_{ik}^{upp})]$, where $[\xi_{ik}^{low}, \xi_{ik}^{upp}]$ are the one-dimensional strata $\mathscr{D}_{ki}$ with $\mathscr{D}_k \equiv \mathscr{D}_{k1} \times \ldots \times \mathscr{D}_{kp}$.

Let p_k denote the probability of stratum $\mathscr{D}_k$, that is, $p_k = \mathbb{P}(\xi \in \mathscr{D}_k)$. Then the SS estimator for $\tau_{n,M}$ based on N_{SS} independent samples of Q_M is defined by

$$\hat{\mathscr{I}}_{n,M}^{SS} = \sum_{k=1}^{r} \frac{p_k}{N_k} \sum_{m=1}^{N_k} \mathscr{I}_n(Q_M^{(m,k)}), \tag{B2}$$

where $Q_M^{(m,k)}$ is the mth sample of Q_M that has a corresponding input vector ($\boldsymbol{\xi}$) in $\mathscr{D}_k$. The variance of $\hat{\mathscr{I}}_{n,M}^{SS}$ is

$$\mathbb{V}[\hat{\mathscr{I}}_{n,M}^{SS}] = \sum_{k=1}^{r} \frac{\sigma_{k,n}^2 p_k^2}{N_k}, \tag{B3}$$

where

$$\sigma_{k,n}^2 = \frac{1}{p_k} \int_{\mathscr{D}_k} (\mathscr{I}_n(Q_M(\mathbf{s})) - \mu_{k,n})^2 dF_{\xi}(\mathbf{s}) \tag{B4}$$

with

$$\mu_{k,n} = \frac{1}{p_k} \int_{\mathscr{D}_k} \mathscr{I}_n(Q_M(\mathbf{s}))\, dF_{\xi}(\mathbf{s}). \tag{B5}$$

Here $\mu_{k,n}$ and $\sigma_{k,n}^2$ represent, respectively, the mean and variance of $\mathscr{I}_n(Q_M(\xi))$ with $\xi \in \mathscr{D}_k$. We will refer to these quantities as the "strata means" and "strata variances", respectively, from now on, with the understanding that they apply to the output space (of Q) rather than the input space (of $\boldsymbol{\xi}$).

A common choice for N_k is proportional allocation (Fishman, 1996), according to which $N_k = p_k N_{SS}$ and (B2) becomes

$$\hat{\mathscr{I}}_{n,M}^{SS} = \frac{1}{N_{SS}} \sum_{k=1}^{r} \sum_{m=1}^{N_k} \mathscr{I}_n(Q_M^{(m,k)}). \tag{B6}$$

In the limit of $N_k = 1$ (i.e., one sample per stratum) for all $k = 1, \ldots, r$, the variance is (McKay et al., 2000)

$$\mathbb{V}[\hat{\mathscr{I}}_{n,M}^{SS}] = \mathbb{V}(\hat{\mathscr{I}}_{n,M}^{MC}) - \frac{1}{N_{SS}^2} \sum_{j=1}^{N_{SS}} (\mu_{j,n} - \tau_{n,M})^2. \tag{B7}$$

This result demonstrates the variance reduction achieved through stratification.

B2. Latin Hypercube Sampling

In Latin hypercube sampling (LHS), the range of the CDFs F_{ξ_i} ($i = 1, \ldots, p$) is subdivided into N_{LHS} strata $\mathscr{D}_{ik}$ ($k = 1, \ldots, N_{LHS}$) of equal probability $1/N_{LHS}$; that is, stratification occurs in *probability* space. Only one sample is drawn from each stratum. The Cartesian product of these strata across the stochastic input dimensions yields N_{LHS}^{p} cells $\mathbf{m}_j = (m_{j\,1}, m_{j\,2}, \ldots, m_{jp})$ with equal probability N_{LHS}^{-p}, where m_{ji} is the interval number of component ξ_i represented in cell j ($j = 1, \ldots, N_{LHS}^{p}$). A Latin hypercube sample of size N_{LHS} is then obtained by randomly selecting N_{LHS} cells $\mathbf{m}_1, \ldots, \mathbf{m}_{N_{LHS}}$, with the condition that for each i the set $\{m_{ji}\}_{j=1}^{N_{LHS}}$ is a random permutation of the integers $1, \ldots, N_{LHS}$. This yields samples $\mathbf{s}_j = [s_{j\,1}, \ldots, s_{jp}]$ ($j = 1, \ldots, N_{LHS}$) with

$$s_{ji}^{(k)} = F_{\xi_i}^{-1}(U_{ik}^{(j)}), \qquad i = 1, ..., p, \tag{B8}$$

where the $U_{ik}^{(j)}$ are independent, uniformly distributed samples on $[\zeta_{ik}^{\text{low}}, \zeta_{ik}^{\text{upp}}]$ with $\zeta_{ik}^{\text{low}} = (k-1)/N_{\text{LHS}}$ and $\zeta_{ik}^{\text{upp}} = k/N_{\text{LHS}}$.

The LHS estimator for $\tau_{n,M}$ based on N_{LHS} independent samples of Q_M is written as

$$\hat{\mathcal{F}}_{n,M}^{\text{LHS}} = \frac{1}{N_{\text{LHS}}} \sum_{j=1}^{N_{\text{LHS}}^p} w_j \mathcal{F}_n(Q_M^{(j)}), \tag{B9}$$

where w_j is an indicator variable defined as

$$w_j = \begin{cases} 1 & \text{if cell } j \text{ is in the sample} \\ 0 & \text{otherwise.} \end{cases} \tag{B10}$$

Appendix C: Computation of $\hat{F}^{\mathrm{MLsm}}_{h,\delta_K,M}$ and $\hat{F}^{\mathrm{HLSS\text{-}ML}}_{h,M}$

C1. Standard Kernel-Smoothed MLMC

Algorithm 1: Standard multilevel Monte Carlo with KDE-based smoothing

Input : RMSE tolerance ϵ; set of $S+1$ interpolation points S_h; sequence of discrete meshes $\{\mathcal{T}_{M_l}, l=0,\dots,L_{\max}\}$; initial number of samples N_l^0 at each level l; tuning parameter α;
Output : An estimate of the CDF $F(q)$;
Procedure: :
Generate $N_{\max}$ samples of the random input field $(\star)$;
Initialize $L=-1$;
while $L < L_{\max}$ **do**
  Set $L = L+1$;
  if $L=0$ **then**
    Compute N_0^0 samples of Q_{M_0} using upscaled values of $(\star)$;
  else
    Compute N_L^0 samples of Q_{M_L} and $Q_{M_{L-1}}$ using upscaled values of $(\star)$;
  end
  Compute $\delta_{K,L}$;
  for $n=0,\dots,S$ **do**
    for $j=1,\dots,N_L^0$ **do**
      Compute $g_n(Y_L^{(j)})$;
    end
  end
  Compute the computational cost at level L, $\bar{w}_L$;
  for $n=0,\dots,S$ **do**
    Compute $\hat{I}_n^{\mathrm{MC}}(Y_L)$ and $\hat{I}_n^{\mathrm{MCsm}}(Y_L)$;
    Compute $\tilde{V}[g_n(Y_L)] = \sum_{j=1}^{N_L^0}(g_n(Y_L^{(j)}) - \hat{I}_n^{\mathrm{MCsm}}(Y_L))^2/\max(N_L^0-1,1)$;
  end
  Set $N_L = \mathrm{ceil}\left(\max_{0\le n\le S}\frac{2}{\alpha\epsilon^2}\sqrt{\tilde{V}[g_n(Y_L)]/\bar{w}_L}\left(\sum_{z=0}^{L}\sqrt{\tilde{V}[g_n(Y_z)]\bar{w}_z}\right)\right)$;
  if $L=0$ **then**
    Compute $\max(N_0 - N_0^0, 0)$ samples of Q_{M_0} using upscaled values of $(\star)$;
  else
    Compute $\max(N_L - N_L^0, 0)$ samples of Q_{M_L} and $Q_{M_{L-1}}$ using upscaled values of $(\star)$;
  end
  for $n=0,\dots,S$ **do**
    for $j=N_L^0+1,\dots,N_L$ **do**
      Compute $g_n(Y_L^{(j)})$;
    end
  end
  Compute the computational cost at level L, $\bar{w}_L$;
  for $n=0,\dots,S$ **do**
    Compute $\hat{I}_n^{\mathrm{MC}}(Y_L)$ and $\hat{I}_n^{\mathrm{MCsm}}(Y_L)$;
    Compute $\tilde{V}[g_n(Y_L)] = \sum_{j=1}^{N_L}(g_n(Y_L^{(j)}) - \hat{I}_n^{\mathrm{MCsm}}(Y_L))^2/\max(N_L-1,1)$;
  end
  Set $N_L^\star = N_L$;
  for $l=0,\dots,L-1$ **do**
    Set $N_l = \mathrm{ceil}\left(\max_{0\le n\le S}\frac{2}{\alpha\epsilon^2}\sqrt{\tilde{V}[g_n(Y_l)]/\bar{w}_l}\left(\sum_{z=0}^{L}\sqrt{\tilde{V}[g_n(Y_z)]\bar{w}_z}\right)\right)$;
    if $l=0$ **then**
      Compute $\max(N_0 - N_0^\star, 0)$ samples of Q_{M_0} using upscaled values of $(\star)$;
    else
      Compute $\max(N_l - N_l^\star, 0)$ samples of Q_{M_l} and $Q_{M_{l-1}}$ using upscaled values of $(\star)$;
    end
    for $n=0,\dots,S$ **do**
      for $j=N_l^\star+1,\dots,N_l$ **do**
        Compute $g_n(Y_l^{(j)})$;
      end
    end
    Compute the computational cost at level l, $\bar{w}_l$;
    for $n=0,\dots,S$ **do**
      Compute $\hat{I}_n^{\mathrm{MC}}(Y_l)$ and $\hat{I}_n^{\mathrm{MCsm}}(Y_l)$;
      Compute $\tilde{V}[g_n(Y_l)] = \sum_{j=1}^{N_l}(g_n(Y_l^{(j)}) - \hat{I}_n^{\mathrm{MCsm}}(Y_l))^2/\max(N_l-1,1)$;
    end
    Set $N_l^\star = N_l$;
  end
  $\star$
end

⋆ **if** $(L \geq 1$ and $\max_{0\leq n\leq S} |\hat{I}_n^{MC}(Y_L)| \leq \sqrt{1-\alpha}\,\epsilon)$ or $(L = L_{\max})$ **then**
  Set $M = M_L$;
  Compute the kernel-smoothed MLMC estimator of $F(q)$, $\hat{F}^{\text{MLsm}}_{h,\delta_K,M}(q)$;
  Compute the cost of kernel-based MLMCsm, $C(\hat{F}^{\text{MLsm}}_{h,\delta_K,M})$;
  Set $N_{\text{MC}} = (\alpha\epsilon^2)^{-1} \max_{0\leq n\leq S} \tilde{V}[\mathcal{I}_n(Q_M)]$;
  Compute the cost of MC, $C(\hat{F}^{\text{MC}}_{h,M})$;
  Compute $\max(N_{\text{MC}} - N_L, 0)$ samples of Q_M;
  Compute the MC estimator of $F(q)$, $\hat{F}^{\text{MC}}_{h,M}(q)$;
  Finish;
end

C2. HLSS-MLMC

Algorithm 2: Multilevel Monte Carlo with hierarchical Latinized stratified sampling

Input : RMSE tolerance ϵ, $S+1$ interpolation points $\mathcal{S}_h$; sequence of discrete meshes $\{\mathcal{T}_{M_l}, l = 0, \ldots, L_{\max}\}$; tuning parameter α;
Output : An estimate of the CDF $F(q)$;
Procedure:
Generate $N_{\max,\text{strat}}$ samples of the random input field $(\star)$;
Initialize $L = -1$;
while $L < L_{\max}$ **do**
  Set $L = L + 1$;
  if $L = 0$ **then**
    Compute 1 sample of Q_{M_0} using upscaled values of $(\star)$;
    Extend the sample size of Q_{M_0} by a factor of 2;
    for $n = 0, \ldots, S$ **do**
      for $j = 1, \ldots, N_0$ **do**
        Compute $\mathcal{I}_n(Y_0^{(j)})$;
      end
      Compute $\hat{I}_n^{\text{HLSS}}(Y_0)$ and $\tilde{V}[\hat{I}_n^{\text{HLSS}}(Y_0)]$;
    end
    while $\max_{0\leq n\leq S} \tilde{V}[\hat{I}_n^{HLSS}(Y_0)] > \alpha\epsilon^2$ **do**
      Extend the sample size of Q_{M_0} by a factor of 2;
      for $n = 0, \ldots, S$ **do**
        for $j = 1, \ldots, N_0$ **do**
          Compute $\mathcal{I}_n(Y_0^{(j)})$;
        end
        Compute $\hat{I}_n^{\text{HLSS}}(Y_0)$ and $\tilde{V}[\hat{I}_n^{\text{HLSS}}(Y_0)]$;
      end
    end
    Set N_0 = total number of samples of Q_{M_0};
  else
    Compute 1 sample of Q_{M_L} and $Q_{M_{L-1}}$ using upscaled values of $(\star)$;
    Extend the sample size of Q_{M_L} and $Q_{M_{L-1}}$ by a factor of 2;
    for $n = 0, \ldots, S$ **do**
      for $j = 1, \ldots, N_L$ **do**
        Compute $\mathcal{I}_n(Y_L^{(j)})$;
      end
      Compute $\hat{I}_n^{\text{HLSS}}(Y_L)$ and $\tilde{V}[\hat{I}_n^{\text{HLSS}}(Y_L)]$;
    end
    while $\sum_{l=0}^{L} \max_{0\leq n\leq S} \tilde{V}[\hat{I}_n^{\text{HLSS}}(Y_l)] > \alpha\epsilon^2$ **do**
      Extend the sample size of Q_{M_L} and $Q_{M_{L-1}}$ by a factor of 2;
      for $n = 0, \ldots, S$ **do**
        for $j = 1, \ldots, N_L$ **do**
          Compute $\mathcal{I}_n(Y_L^{(j)})$;
        end
        Compute $\hat{I}_n^{\text{HLSS}}(Y_L)$ and $\tilde{V}[\hat{I}_n^{\text{HLSS}}(Y_L)]$;
      end
    end
    Set N_L = total number of samples of Q_{M_L};
  end
  Compute the computational cost at level L, $\bar{w}_L$;
  if $(L \geq 1$ and $\max_{0\leq n\leq S} |\hat{I}_n^{HLSS}(Y_L)| \leq \sqrt{1-\alpha}\,\epsilon)$ *or* $(L = L_{\max})$ **then**
    Set $M = M_L$;
    Compute the HLSS-MLMC estimator of $F(q)$, $\hat{F}^{\text{HLSS-ML}}_{h,M}(q)$;
    Compute the cost of HLSS-enhanced MLMC, $C(\hat{F}^{\text{HLSS-ML}}_{h,M})$;
    Finish;
  end
end

Data Availability Statement

There are no data sharing issues since all of the numerical information is provided in the figures produced by solving the equations in the paper.

Acknowledgments
The authors thank Hamdi Tchelepi for fruitful discussions. They are also grateful to Michael D. Shields for generously sharing his HLSS MATLAB® code and providing them with insightful comments. This work was supported in part by Air Force Office of Scientific Research under Award Number FA9550-17-1-0417 and by a gift from TOTAL. S. B. is supported by a named Stanford Graduate Fellowship in Science and Engineering (SGF).

References

Appleyard, J., Cheshire, I., & Pollard, R. (1981). Special techniques for fully implicit simulators. *Proceedings of the European Symposium on Enhanced Oil Recovery*, 395–408.

Aziz, K., & Settari, A. (1979). *Petroleum reservoir simulation*. London: Elsevier Applied Science Publishers.

Barajas-Solano, D. A., & Tartakovsky, D. M. (2016). Stochastic collocation methods for nonlinear parabolic equations with random coefficients. *SIAM/ASA Journal on Uncertainty Quantification*, *4*, 475–494.

Bierig, C., & Chernov, A. (2016). Approximation of probability density functions by the Multilevel Monte Carlo Maximum Entropy method. *Journal of Computational Physics*, *314*, 661–681.

Boso, F., & Tartakovsky, D. M. (2018). Information-theoretic approach to bidirectional scaling. *Water Resources Research*, *54*, 4916–4928. https://doi.org/10.1029/2017WR021993

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

Corey, A. T. (1954). The interrelation between gas and oil relative permeabilities. *Monthly Production*, *19*(1), 38–41.

Crevillén-García, D., & Power, H. (2017). Multilevel and quasi-Monte Carlo methods for uncertainty quantification in particle travel times through random heterogeneous porous media. *Royal Society Open Science*, *4*, 170203.

Durlofsky, L. J. (2005). Upscaling and gridding of fine scale geological models for flow simulation. In *Paper presented at the 8th International Forum on Reservoir Simulation, Îles Borromées, Stresa*.

Fishman, G. S. (1996). *Monte Carlo: Concepts, algorithms and applications*. New York: Springer Series in Operations Research.

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

Giles, M. B., Nagapetyan, T., & Ritter, K. (2015). Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA Journal for Uncertainty Quantification*, *3*, 267–295.

Giles, M. B., Nagapetyan, T., & Ritter, K. (2017). Adaptive Multilevel Monte Carlo approximation of distribution functions. arXiv:1706.06869.

Gittelson, C. J., Könnö, J., Schwab, C., & Stenberg, R. (2013). The multi-level Monte Carlo finite element method for a stochastic Brinkman problem. *Numerische Mathematik*, *125*(2), 347–386.

Gorodetsky, A. A., Geraci, G., Eldred, M. S., & Jakeman, J. D. (2020). A generalized approximate control variate framework for multifidelity uncertainty quantification. *Journal of Computational Physics*, *408*, 109,257.

Heinrich, S. (1998). Monte Carlo complexity of global solution of integral equations. *Journal of Complexity*, *14*, 151–175.

Heinrich, S. (2001). Multilevel Monte Carlo methods. *Springer Lecture Notes in Computer Science*, *2179*, 3624–3651.

Kebaier, A., & Lelong, J. (2018). Coupling importance sampling and Multilevel Monte Carlo using sample average approximation. *Methodology and Computing in Applied Probability*, *20*, 611–641.

Krumscheid, S., & Nobile, F. (2018). Multilevel Monte Carlo approximation of functions. *SIAM/ASA Journal for Uncertainty Quantification*, *6*, 1256–1293.

Kumar, P., Rodrigo, C., Gaspar, F. J., & Oosterlee, C. W. (2019). A parametric acceleration of multilevel Monte Carlo convergence for nonlinear variably saturated flow. *Computers & Geosciences*, *24*, 311–331.

Kuo, F. Y., Scheichl, R., Schwab, C., Sloan, I. H., & Ullmann, E. (2017). Multilevel Quasi-Monte Carlo methods for lognormal diffusion problems. *Mathematics of Computation*, *86*, 2827–2860.

Linde, N., Ginsbourger, D., Irving, J., Nobile, F., & Doucet, A. (2017). On uncertainty quantification in hydrogeology and hydrogeophysics. *Advances in Water Resources*, *17*, 166–181.

Lu, Z., Neuman, S. P., Guadagnini, A., & Tartakovsky, D. M. (2002). Conditional moment analysis of steady state unsaturated flow in bounded, randomly heterogeneous soils. *Water Resources Research*, *38*(4), 1038, 1–15. https://doi.org/10.1029/2001WR000278

Lu, D., Zhang, G., Webster, C., & Barbier, C. (2016). An improved multilevel Monte Carlo method for estimating probability distribution functions in stochastic oil reservoir simulations. *Water Resources Research*, *52*, 9642–9660. https://doi.org/10.1002/2016WR019475

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55–61.

Moslehi, M., Rajagopal, R., & de Barros, F. P. J. (2015). Optimal allocation of computational resources in hydrogeological models under uncertainty. *Advances in Water Resources*, *83*, 299–309.

Mukherjee, M. (2013). Instrumented permeable blankets for estimating subsurface hydraulic conductivity and confirming numerical models used for subsurface liquid injection (PhD thesis), Mich. State Univ., East Lansing. Phd thesis, mich. state univ., east lansing.

Müller, F., Jenny, P., & Meyer, D. W. (2013). Multilevel Monte Carlo for two phase flow and Buckley–Leverett transport in random heterogeneous porous media. *Journal of Computational Physics*, *250*, 685–702.

Müller, F., Jenny, P., & Meyer, D. W. (2016). Parallel multilevel Monte Carlo for two-phase flow and transport in random heterogeneous porous media with sampling-error and discretization-error balancing. *Society of Petroleum Engineers Journal*, *21*(6), 2027–2037.

Müller, F., Meyer, D. W., & Jenny, P. (2014). Solver-based vs. grid-based multilevel Monte Carlo for two phase flow and transport in random heterogeneous porous media. *Journal of Computational Physics*, *268*, 39–50.

Neuman, S. P., Tartakovsky, D. M., Wallstrom, T. C., & Winter, C. L. (1996). Correction to the Neuman and Orr "nonlocal theory of steady state flow in randomly heterogeneous media". *Water Resources Research*, *32*(5), 1479–1480. https://doi.org/10.1029/96WR00489

O'Malley, D., Karra, S., Hyman, J. D., Viswanathan, H. S., & Srinivasan, G. (2018). Efficient Monte Carlo with graph-based subsurface flow and transport models. *Water Resources Research*, *54*, 3758–3766. https://doi.org/10.1029/2017WR022073

Peherstorfer, B., Willcox, K., & Gunzburger, M. (2016). Optimal model management for multifidelity Monte Carlo estimation. *SIAM Journal of Scientific Computing*, *38*, A3163–A3194.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, *27*(3), 832–837.

Shields, M. D (2016). Refined Latinized stratified sampling: A robust sequential sample size extension methodology for high-dimensional Latin hypercube and stratified designs. *International Journal for Uncertainty Quantification*, *6*(1), 79–97.

Shields, M. D. (2018). Adaptive Monte Carlo analysis for strongly nonlinear stochastic systems. *Reliability Engineering and System Safety*, *175*, 207–224.
Shields, M. D., Teferra, K., Hapij, A., & Daddazio, R. (2015). Refined stratified sampling for efficient Monte Carlo based uncertainty quantification. *Reliability Engineering and System Safety*, *142*, 310–325.
Shields, M. D., & Zhang, J. (2016). The generalization of Latin hypercube sampling. *Reliability Engineering and System Safety*, *148*, 96–108.
Sinsbeck, M., & Tartakovsky, D. M. (2015). Impact of data assimilation on cost-accuracy tradeoff in multifidelity models. *SIAM/ASA Journal for Uncertainty Quantification*, *3*(1), 954–968. https://doi.org/10.1137/141001743
Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, *29*(2), 143–151.
Tartakovsky, D. M. (2017). Uncertainty quantification in subsurface modeling, (3rd). In J. H. Cushman, & D. M. Tartakovsky (Eds.), *The handbook of groundwater engineering* (pp. 625–640). Boca Raton, FL: CRC Press.
Tartakovsky, D. M., Dentz, M., & Lichtner, P. C. (2009). Probability density functions for advective-reactive transport in porous media with uncertain reaction rates. *Water Resources Research*, *45*, W07414. https://doi.org/10.1029/2008WR007383
Tartakovsky, D. M., & Winter, C. L. (2008). Uncertain future of hydrogeology. *ASCE Journal of Hydrologic Engineering*, *13*(1), 37–39.
Taverniers, S., & Tartakovsky, D. M. (2017). Impact of parametric uncertainty on estimation of the energy deposition into an irradiated brain tumor. *Journal of Computational Physics*, *348*, 139–150.
Teckentrup, A., Scheichl, R., Giles, M. B., & Ullman, E. (2013). Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numerische Mathematik*, *125*, 569–600.
Ullmann, E., & Papaioannou, I. (2015). Multilevel estimation of rare events. *SIAM/ASA Journal for Uncertainty Quantification*, *3*, 922–953.
Venturi, D., Tartakovsky, D. M., Tartakovsky, A. M., & Karniadakis, G. E. (2013). Exact PDF equations and closure approximations for advective-reactive transport. *Journal of Computational Physics*, *243*, 323–343. https://doi.org/10.1016/j.jcp.2013.03.001
Winter, C. L., & Tartakovsky, D. M. (2002). Groundwater flow in heterogeneous composite aquifers. *Water Resources Research*, *38*(8), 1148. https://doi.org/10.1029/2001WR000450
Winter, C. L., Tartakovsky, D. M., & Guadagnini, A. (2003). Moment equations for flow in highly heterogeneous porous media. *Surveys in Geophysics*, *24*, 81–106.
Ye, M., Neuman, S. P., Guadagnini, A., & Tartakovsky, D. M. (2004). Nonlocal and localized analyses of conditional mean transient flow in bounded, randomly heterogeneous porous media. *Water Resources Research*, *40*, W05104. https://doi.org/10.1029/2003WR002099