Journal of Computational Physics 419 (2020) 109572

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Estimation of distributions via multilevel Monte Carlo with stratified sampling

Søren Taverniers, Daniel M. Tartakovsky [*]

Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 12 June 2019
Received in revised form 7 March 2020
Accepted 15 May 2020
Available online 9 July 2020

Keywords:
Multilevel Monte Carlo
Stratified sampling
Kernel density estimation
Cumulative distribution function

ABSTRACT

We design and implement a novel algorithm for computing a multilevel Monte Carlo (MLMC) estimator of the joint cumulative distribution function (CDF) of a vector-valued quantity of interest in problems with random input parameters and initial conditions. Our approach combines MLMC with stratified sampling of the input sample space by replacing standard Monte Carlo at each level with stratified Monte Carlo initialized with proportionally allocated samples. We show that the resulting stratified MLMC (sMLMC) algorithm is more efficient than its standard MLMC counterpart due to the additional variance reduction provided by the stratification of the random parameter's domain, especially at the coarsest levels. Additional computational cost savings are obtained by smoothing the indicator function with a Gaussian kernel, which proves to be an efficient and robust alternative to recently developed polynomial-based techniques.

1. Introduction and motivation

Simulation of many complex systems, such as subsurface flows in porous media [1,2] or reaction initiation in heterogeneous explosives [3], is complicated by a lack of information about key properties such as permeability or initial porosity. Uncertainty in a medium's properties or initial state propagates into uncertainty in predicted quantities of interest (QoIs), such as mass flow rate or the material's temperature.

Probabilistic methods, which treat an uncertain input or initial state of the system as a random variable, provide a natural venue to quantify predictive uncertainty in a QoI. These techniques render the QoI random as well, i.e., it takes on values that are distributed according to some probability density function (PDF). Such approaches include stochastic finite element methods (FEMs), which characterize the random parameter fields in terms of a finite set of random variables, e.g., via a spectral representation or a Karhunen-Loève expansion. This finite set of random variables defines a finite-dimensional outcome space on which the solution to the resulting stochastic partial differential equation (PDE) is defined. Examples of stochastic FEMs include stochastic Galerkin, which expands the solution of a stochastic PDE in terms of orthogonal basis functions, and stochastic collocation, which samples the random parameters at predetermined values or "nodes" [4]. While such methods perform well when the number of stochastic parameters (aka "stochastic dimension") is low and these parameters exhibit a long correlation length, for many stochastic degrees of freedom and short correlation lengths their performance decreases dramatically [5]. In addition, since nonlinearity degrades the solution's regularity in the outcome space, stochastic FEMs also struggle with solving highly nonlinear problems [6]. Another class of probabilistic techniques

[*] Corresponding author.
E-mail address: tartakovsky@stanford.edu (D.M. Tartakovsky).

https://doi.org/10.1016/j.jcp.2020.109572
0021-9991/© 2020 Elsevier Inc. All rights reserved.

involves the derivation of deterministic equations for the statistical moments [7,8] or PDF [9,10] of the QoI. While these methods do not suffer from the "curse of dimensionality", they require a closure approximation for the derived moment or PDF equations. Such closures often involve perturbation expansions of relevant quantities into series in the powers of the random parameters' variances, which limits their applicability to parameters with low coefficients of variation.

Monte Carlo simulations (MC) [11] remain the most robust and straightforward way to solve PDEs with random parameters/initial conditions. The method samples the random variables from their distribution, solves the deterministic PDE for each realization, and computes the resulting statistics of the QoI. While combining a nonintrusive character with a convergence that is independent of the stochastic dimension, MC converges very slowly: the standard deviation of the MC estimator for the QoI's expectation value is inversely proportional to $\sqrt{N}$ where N is the number of realizations [11]. While this drawback spurred the development of alternative probabilistic methods such as those listed above, efforts to combine MC with the multigrid concept, by Heinrich [12,13] and later by Giles [14], have sparked renewed interest in MC under the form of the multilevel Monte Carlo (MLMC) method. MLMC aims to achieve the same solution error as MC but at a lower computational cost by correcting realizations on a coarse spatial grid with sampling at finer levels of discretization. By sampling predominantly at coarsest levels, where samples are cheaper to compute, it aims to outperform fine-resolution MC (for a fixed discretization error). The related technique of *multifidelity* MC (also referred to as *solver*-based MLMC as a opposed to traditional *grid*-based MLMC) generalizes this approach by using models of varying fidelities and speeds on the different levels [15–17].

Most work on MLMC deals with estimators of expected values and variances of QoIs (see, e.g., [18–21]). The information contained in these first two moments is insufficient to estimate, e.g., quantiles or the probability of rare events. This task requires the construction of the QoI's PDF or, equivalently, its cumulative distribution function (CDF) [22–24]. Consider a QoI $\mathbf{Q} \equiv (Q_1, \dots, Q_n)^\top$ with $n \in \mathbb{N}_0$ that is a function of a random input ξ, i.e., $\mathbf{Q} = \mathbf{Q}(\xi)$; the continuous random variable $\xi : \Omega \to \mathbb{R}$ is a measurable function with a sample space Ω. The joint CDF F of $\mathbf{Q}$ at a point $\mathbf{q} = (q_1, \dots, q_n)^\top \in \mathbb{R}^n$ is given by the expected value $\mathbb{E}[\mathcal{I}_{\mathbf{A_q}}(\mathbf{Q})]$ of the indicator function

$$\mathcal{I}_{\mathbf{A_q}}(\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{s} = (s_1, \dots, s_n)^\top \in \mathbf{A_q} \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $\mathbf{A_q} \equiv (-\infty,\ q_1] \times \cdots \times (-\infty,\ q_n]$. To fix the key ideas of the proposed approach, we restrict our presentation to a single random input variable ξ; generalization to high-dimensional, correlated random input fields $\boldsymbol{\xi}$ is left for a follow-up study.

Application of MLMC to the estimation of distributions started but a few years ago. Giles et al. [25] developed an algorithm to estimate PDFs and CDFs using the indicator function approach, while Bierig et al. [26] approximated PDFs via a truncated moment sequence and the method of maximum entropy. Both approaches were used to approximate CDFs of molecular species in biochemical reaction networks [27]. Elfverson et al. [28] used MLMC to estimate failure probabilities, which are single-point evaluations of the CDF. Lu et al. [29] calibrated the polynomial smoothing of the indicator function proposed in [25] to optimize the smoothing bandwidth for a given value of the error tolerance, and enabled an a posteriori switch to MC if the latter turned out to have a lower computational cost. Krumscheid et al. [30] developed an algorithm for approximating general parametric expectations, including CDFs and characteristic functions, simultaneously deriving robustness indicators such as quantiles and conditional values-at-risk.

To reduce the computational cost of MLMC further, the standard MC approach at each discretization level can be replaced with a more efficient sampling strategy. For example, quasi-Monte Carlo uses quasi-random, rather than random or pseudo-random, sequences to achieve faster convergence than MC, and has been used to speed up the MLMC computation of the mean system state [31]. Furthermore, a number of "variance reduction" techniques have been developed to obtain estimators with a lower variance than MC for the same number of realizations. These include stratification, antithetic sampling, importance sampling, and control variates [11]. Recently, importance sampling was incorporated into MLMC for a more efficient computation of expected values of QoIs [32]. Ullman et al. [33] estimated failure probabilities for rare events by combining subset simulation using Markov chain Monte Carlo [34] with multilevel failure domains defined on a hierarchy of discrete spatial grids with decreasing mesh sizes.

We propose a novel indicator-function based MLMC algorithm for the estimation of CDFs. It replaces simple MC sampling of the QoI at each discretization level with its stratified counterpart by dividing the domain of a random input parameter or initial condition into a number of regions or *strata*, and samples the random variable from each of these strata. We use the resulting "stratified" MLMC (sMLMC) approach to estimate the joint CDF of the vector-valued QoI $\mathbf{Q}$. Furthermore, we develop a smoothing technique for the indicator function based on Kernel Density Estimation (KDE), and combine this with the sMLMC algorithm to magnify the variance reduction (and hence cost savings).

A mathematical description of CDF estimators of vector-valued QoIs in general, and smoothed MLMC and sMLMC estimators and their associated cost and error in particular, is provided in Section 2. In Section 3, we formulate two testbed problems, one for a scalar QoI and one for a vector-valued QoI, which are used to compare the performance of the various MLMC and sMLMC algorithms. Section 4 contains results of the numerical experiments we performed on each of these problems. Conclusions and future research directions are reserved for Section 5.

2. Multilevel Monte Carlo for CDFs

Let the QoI $\mathbf{Q} \in \mathbb{R}^n$ be an output derived from a numerical solution of a PDE obtained, e.g., via finite volume or finite difference. Its computation requires a discretization of the simulation domain with a spatial grid $\mathcal{T}_M$ and introduces a sequence of random variables $\mathbf{Q}_M$, where M is the number of cells in $\mathcal{T}_M$, which converges to $\mathbf{Q}$ as M increases. Our goal is to approximate the distribution of $\mathbf{Q}_M$, rather than of $\mathbf{Q}$, i.e., to estimate the joint CDF F_M of $\mathbf{Q}_M$ on a compact set $\mathbf{A} \equiv [a_1, b_1] \times \cdots \times [a_n, b_n] \subset \mathbb{R}^n$. At each point $\mathbf{q} \in \mathbf{A}$, $F_M(\mathbf{q})$ is given by

$$F_M(\mathbf{q}) = \mathbb{E}[\mathcal{I}_{\mathbf{A_q}}(\mathbf{Q}_M)], \tag{2}$$

where $\mathbf{A_q} \equiv (-\infty,\ q_1] \times \cdots \times (-\infty,\ q_n]$ and $\mathbf{q} = (q_1, \ldots, q_n)^\top$. This approach is justified if $\mathbf{Q}_M$ converges to $\mathbf{Q}$ as $M \to \infty$ both in the mean and in the sense of distributions, i.e.,

$$\mathbb{E}[\mathbf{Q}_M - \mathbf{Q}] = \mathcal{O}(M^{-\alpha_1}), \qquad \mathbb{E}[\mathcal{I}_{\mathbf{A_q}}(\mathbf{Q}_M) - \mathcal{I}_{\mathbf{A_q}}(\mathbf{Q})] = \mathcal{O}(M^{-\alpha_2}), \tag{3}$$

as $M \to \infty$ for $\alpha_1, \alpha_2 \in \mathbb{R}$ independent of M and $\mathbf{q}$.

Consider a set of points $\mathcal{S}_\mathbf{h} = \{a_1 = q_{1,0} < q_{1,1} < \ldots < q_{1,S_1} = b_1\} \times \ldots \times \{a_n = q_{n,0} < q_{n,1} < \ldots < q_{n,S_n} = b_n\}$, where q_{i,v_i} $(v_i = 0, \ldots, S_i)$ are equidistant nodes on $[a_i, b_i]$ with separation distance h_i, and $\mathbf{h} = (h_1, \ldots, h_n)^\top$. At each of these points, F_M is computed by defining an unbiased estimator $\hat{\mathcal{I}}_{\mathbf{v},M}$ of $\mathbb{E}[\mathcal{I}_\mathbf{v}(\mathbf{Q}_M)]$ where $\mathcal{I}_\mathbf{v}(\mathbf{Q}_M) \equiv \mathcal{I}_{\mathbf{A}_{\mathbf{q_v}}}(\mathbf{Q}_M)$, with $\mathbf{A}_{\mathbf{q_v}} \equiv (-\infty,\ q_{1,v_1}] \times \ldots \times (-\infty,\ q_{n,v_n}]$ and $\mathbf{v} = (v_1, \ldots, v_n)^\top$. At any other $\mathbf{q} \in \mathbf{A}$, $F_M(\mathbf{q})$ is evaluated via an n-dimensional piecewise polynomial interpolation. The latter's degree, $\max(d, 1)$, is determined by the smoothness of the PDF f_M of $\mathbf{Q}_M$: f_M needs to be d-times continuously differentiable on $[a_1 - \zeta_1, b_1 + \zeta_1] \times \ldots \times [a_n - \zeta_n, b_n + \zeta_n]$ for some $d \in \mathbb{N}_0$ and $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)^\top \in \mathbb{R}^n_{>0}$. We employ a cubic spline interpolation, i.e., assume that the PDF f_M is at least three times continuously differentiable on its support. The resulting estimator (i.e., after interpolation) is referred to as $\hat{F}_{\mathbf{h},M}(\mathbf{q})$.

The discrepancy between the estimator $\hat{F}_{\mathbf{h},M}(\mathbf{q})$ and the true CDF of $\mathbf{Q}$, $F_\mathbf{h}(\mathbf{q})$, is due to both the discretization error from approximating $F_\mathbf{h}$ by $F_{\mathbf{h},M}$ and the sampling error from approximating $F_{\mathbf{h},M}$ by $\hat{F}_{\mathbf{h},M}$.[1] This yields a bound for ϵ^2_{est}, the mean squared error (MSE),

$$\begin{aligned}\underbrace{\mathbb{E}[\|F_\mathbf{h} - \hat{F}_{\mathbf{h},M}\|^2_\infty]}_{\varepsilon^2_{\text{est}}} &\le \mathbb{E}[\|\hat{F}_{\mathbf{h},M} - \mathbb{E}[\hat{F}_{\mathbf{h},M}]\|^2_\infty] + \|F_{\mathbf{h},M} - F_\mathbf{h}\|^2_\infty \\ &\le \underbrace{\max_{\mathbf{v}\in\mathcal{S}} \mathbb{V}[\hat{\mathcal{I}}_{\mathbf{v},M}]}_{\varepsilon^2_{\text{sam}}} + \underbrace{\max_{\mathbf{v}\in\mathcal{S}} |\mathbb{E}[\mathcal{I}_\mathbf{v}(\mathbf{Q}_M) - \mathcal{I}_\mathbf{v}(\mathbf{Q})]|^2}_{\varepsilon^2_{\text{dis}}}\end{aligned} \tag{4}$$

Here $\mathcal{S} = \{0, \ldots, S_1\} \times \cdots \times \{0, \ldots, S_n\}$; $\|\cdot\|_\infty$ refers to the L^∞ norm; $\mathbb{V}[\cdot]$ denotes the variance operator; and ε_{sam} and ε_{dis} are, respectively, the sampling and discretization errors, in the root mean squared sense.

2.1. Standard MLMC with smoothing based on kernel density estimation (KDE)

The first estimator $\hat{F}$ we consider is grounded in the standard MLMC approach for estimating distributions [25]. While the original algorithm discussed in Appendix A.1 uses the indicator function directly, the slow decay of its variance with discretization level due to its jump discontinuity suggests regularization via a sigmoid, e.g., a polynomial g_G [25]. Polynomial-based smoothing replaces the indicator function $\mathcal{I}_\mathbf{v}(\mathbf{Q})$ with a product of polynomials, $\prod_{i=1}^n g_\text{G}[(Q_i - q_{i,v_i})/\delta_{\text{G},i}]$, where $\delta_{\text{G},i}$ $(i = 1, \ldots, n)$ is the smoothing bandwidth, i.e., the distance over which the discontinuity is smeared out, along the ith dimension of $\mathbf{Q}$. As demonstrated in [29], this enhances variance decay with increasing l and hence speeds up the algorithm. A brief overview of the polynomial-smoothed MLMC estimator is provided in Appendix A.2.

We propose an alternative approach inspired by KDE [35]. Specifically, we replace $\mathcal{I}_\mathbf{v}(\mathbf{Q})$ with the product of Gaussian kernels, g_K,

$$\mathcal{I}_\mathbf{v}(\mathbf{Q}) \approx \prod_{i=1}^n \Phi[(q_{i,v_i} - Q_i)/\delta_{\text{K},i}] \equiv \prod_{i=1}^n g_\text{K}[(q_{i,v_i} - Q_i)/\delta_{\text{K},i}],$$

[1] We assume the interpolation error is negligible for the number of interpolation points $\prod_{i=1}^n (S_i + 1)$ used.

where Φ is the standard Gaussian CDF, and $\delta_{\mathrm{K},i}$ is the smoothing bandwidth along the ith dimension of $\mathbf{Q}$. This approximation transforms (A.7) into

$$g_{\mathbf{v}}(\mathbf{Y}_l^{(j)}) = \begin{cases} \prod_{i=1}^n g_{\mathrm{K}}\left(\dfrac{q_{i,v_i} - Q_{i,M_l}^{(j)}}{\delta_{\mathrm{K},l,i}}\right) - g_{\mathrm{K}}\left(\dfrac{q_{i,v_i} - Q_{i,M_{l-1}}^{(j)}}{\delta_{\mathrm{K},l,i}}\right) & 1 \le l \le L_{\max} \\ \prod_{i=1}^n g_{\mathrm{K}}\left(\dfrac{q_{i,v_i} - Q_{i,M_l}^{(j)}}{\delta_{\mathrm{K},l,i}}\right) & l = 0, \end{cases} \tag{5}$$

where the subscript l of the bandwidth indicates its level-dependence.

Finding an optimal value of the bandwidths $\delta_{l,i}$ (given by $\delta_{\mathrm{G},l,i}$ for polynomial smoothing or $\delta_{\mathrm{K},l,i}$ for KDE-based smoothing) at a given error tolerance ϵ is crucial to the performance of MLMC. The mean squared smoothing error, $(\varepsilon_{\text{smooth}}^{\text{MLsm}})^2$, defined in (A.9) should be as close as possible to its maximum allowable value, $(\alpha/2)\epsilon^2$, with $0 < \alpha < 1$, in order to maximize the smoothing and, hence, the variance decay. We pursue the following strategy.

1. Given the error tolerance ϵ, at level $l = 0$ estimate the bandwidths $\delta_{l,\mathbf{v},i}$ $(i = 1, \dots, n)$ for each interpolation point $\mathbf{q}_{\mathbf{v}}$ in $\mathcal{S}_h$ by solving

$$\frac{1}{N_l^0}\left|\sum_{j=1}^{N_l^0}\left[\prod_{i=1}^n g\left(\pm\frac{Q_{i,M_l}^{(j)} - q_{i,v_i}}{\delta_{l,\mathbf{v},i}}\right) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_l}^{(j)})\right]\right| = \epsilon\sqrt{\frac{\alpha}{2}} \tag{6}$$

   based on a set of initial samples $\{\mathbf{Q}_{M_l}^{(j)}\}_{j=1}^{N_l^0}$, where "+" is used for polynomial smoothing and "-" for KDE-based smoothing.
2. Define the smoothing parameter for level $l = 0$ along the ith dimension, $\delta_{0,i}$, as

$$\delta_{0,i} = \max_{\mathbf{v}\in\mathcal{S}} \delta_{0,\mathbf{v},i}. \tag{7}$$

3. Repeat steps 1 and 2 for each new level.

We follow the numerical algorithm in Appendix A.3 to compute $\hat{F}_{\mathbf{h},\delta,M}^{\text{MLsm}}$ and to measure the associated computational cost (see Section 2.3). This algorithm is inspired [18,29].

2.2. Stratified MLMC (sMLMC)

Let $\xi = \xi(\omega) : \Omega \to \mathcal{D}$ be a random input parameter with the sample space Ω, the input domain $\mathcal{D}$, and the output QoI $\mathbf{Q}(\xi)$. In stratified Monte Carlo (sMC), one divides $\mathcal{D}$ into r mutually exclusive and exhaustive regions $\mathcal{D}_k$ $(k = 1, \dots, r)$ called strata.

Let F_ξ denote the CDF of ξ, and $p_k = \mathbb{P}(\xi \in \mathcal{D}_k)$ be the probability of ξ being in stratum $\mathcal{D}_k$. Then, the expected value of $\mathbf{Q}_M(\xi)$ is given by

$$\mathbb{E}[\mathbf{Q}_M(\xi)] = \sum_{k=1}^r p_k \boldsymbol{\zeta}_k. \tag{8}$$

Here $\boldsymbol{\zeta}_k$ $(k = 1, \dots, r)$ is the expected value of $\mathbf{Q}_M(\xi)$ for $\xi \in \mathcal{D}_k$. Its ith component $(i = 1, \dots, n)$ is given by

$$\zeta_{k,i} = \frac{1}{p_k}\int_{\mathcal{D}_k} Q_{i,M}(w)\,\mathrm{d}F_\xi(w). \tag{9}$$

The sMC estimator for $\mathbb{E}[\mathbf{Q}_M(\xi)]$ is

$$\hat{\mathbf{Q}}_M^{\text{sMC}} = \sum_{k=1}^r \frac{p_k}{n_k}\sum_{j=1}^{n_k} \mathbf{Q}_M^{(j,k)}, \tag{10}$$

where n_k is the number of independent samples of $\mathbf{Q}_M(\xi)$ with $\xi \in \mathcal{D}_k$ for each $k = 1, \dots, r$ with $\sum_{k=1}^r n_k \equiv N$; and $\mathbf{Q}_M^{(j,k)}$ is the jth sample of $\mathbf{Q}_M$ that has a corresponding input parameter (ξ) in $\mathcal{D}_k$. The variance of the ith component $(i = 1, \dots, n)$ of $\hat{\mathbf{Q}}_M^{\text{sMC}}$ is

$$\mathbb{V}[\hat{Q}^{\mathrm{sMC}}_{i,M}] = \sum_{k=1}^{r} \frac{\sigma^2_{k,i}\, p^2_k}{n_k}, \qquad \sigma^2_{k,i} = \frac{1}{p_k} \int_{\mathcal{D}_k} (Q_{i,M}(w) - \zeta_{k,i})^2 \mathrm{d}F_\xi(w), \tag{11}$$

where $\sigma^2_{k,i}$ is the variance of $Q_{i,M}(\xi)$ with $\xi \in \mathcal{D}_k$. We refer to these variances as "strata variances", with the understanding that they apply to the output space (of $\mathbf{Q}$) rather than the input space (of ξ).

Two common choices for n_k are proportional and optimal allocations [11]. In the former, $n_k = p_k N$ and

$$\hat{\mathbf{Q}}^{\mathrm{sMC}}_M = \frac{1}{N} \sum_{k=1}^{r} \sum_{j=1}^{n_k} \mathbf{Q}^{(j,k)}_M, \tag{12}$$

$$\mathbb{V}[\hat{Q}^{\mathrm{sMC}}_{i,M}] = \frac{1}{N} \sum_{k=1}^{r} \sigma^2_{k,i}\, p_k = \mathbb{V}[\hat{Q}^{\mathrm{MC}}_{i,M}] - \frac{1}{N} \sum_{k=1}^{r} p_k (\zeta_{k,i} - \mathbb{E}[Q_{i,M}])^2, \tag{13}$$

which shows that stratification produces an estimator with a lower variance than its MC counterpart. We use proportional allocation to initialize the values of n_k on each level of the multilevel algorithm.

When estimating $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$, we replace (10) with

$$\hat{\mathcal{I}}^{\mathrm{sMC}}_{\mathbf{v},M}(\mathbf{Q}_M) = \sum_{k=1}^{r} \frac{p_k}{n_k} \sum_{j=1}^{n_k} \mathcal{I}_{\mathbf{v}}(\mathbf{Q}^{(j,k)}_M). \tag{14}$$

We incorporate (14) into the multilevel framework and obtain the following "stratified" MLMC (sMLMC) estimator for $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ (cf. (A.3))

$$\begin{aligned} \hat{\mathcal{I}}^{\mathrm{sML}}_{\mathbf{v},M} &= \sum_{l=0}^{L_{\max}} \hat{\mathcal{I}}^{\mathrm{sMC}}_{\mathbf{v}}(\mathbf{Y}_l) \\ &= \sum_{l=0}^{L_{\max}} \sum_{k=1}^{r} \frac{p_k}{n_{k,l}} \sum_{j=1}^{n_k} \mathcal{I}_{\mathbf{v}}(\mathbf{Y}^{(j,k)}_l), \end{aligned} \tag{15}$$

where $\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)$ with $l = 0, \ldots, L_{\max}$ are defined in (A.2b). The MSE of the non-smoothed sMLMC estimator $\hat{F}^{\mathrm{sML}}_{\mathbf{h},M}$ for $F_{\mathbf{h},M}$ is bounded by

$$\underbrace{\mathbb{E}[\|F_{\mathbf{h}} - \hat{F}^{\mathrm{sML}}_{\mathbf{h},M}\|^2_\infty]}_{\varepsilon^2_{\mathrm{est}}} \le \underbrace{\max_{\mathbf{v}\in\mathcal{S}} \sum_{l=0}^{L_{\max}} \mathbb{V}[\hat{\mathcal{I}}^{\mathrm{sMC}}_{\mathbf{v}}(\mathbf{Y}_l)]}_{(\varepsilon^{\mathrm{sML}}_{\mathrm{sam}})^2} + \underbrace{\max_{0\le n\le S} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{L_{\max}}}) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q})]|^2}_{(\varepsilon^{\mathrm{sML}}_{\mathrm{dis}})^2}, \tag{16}$$

with $M_{L_{\max}} = M$.

To reduce the computational cost further, we smooth the indicator function. This introduces an additional error term in (16) similar to $(\varepsilon^{\mathrm{MLsm}}_{\mathrm{smooth}})^2$ in (A.9). The resulting sMLMC estimator with smoothing for $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ is defined as

$$\hat{\mathcal{I}}^{\mathrm{sMLsm}}_{\mathbf{v},M} = \sum_{l=0}^{L_{\max}} \hat{\mathcal{I}}^{\mathrm{sMCsm}}_{\mathbf{v}}(\mathbf{Y}_l), \tag{17}$$

where $\hat{\mathcal{I}}^{\mathrm{sMCsm}}_{\mathbf{v}}(\mathbf{Y}_l) = \sum_{k=1}^{r} (p_k/n_{k,l}) \sum_{j=1}^{n_{k,l}} g_{\mathbf{v}}(\mathbf{Y}^{(j,k)}_l)$ and

$$g_{\mathbf{v}}(\mathbf{Y}^{(j,k)}_l) = \begin{cases} \prod_{i=1}^{n} g_{\mathrm{K}}\left(\frac{q_{i,v_i} - Q^{(j,k)}_{i,M_l}}{\delta_{\mathrm{K},l,i}}\right) - g_{\mathrm{K}}\left(\frac{q_{i,v_i} - Q^{(j,k)}_{i,M_{l-1}}}{\delta_{\mathrm{K},l,i}}\right) & 1 \le l \le L_{\max} \\ \prod_{i=1}^{n} g_{\mathrm{K}}\left(\frac{q_{i,v_i} - Q^{(j,k)}_{i,M_l}}{\delta_{\mathrm{K},l,i}}\right) & l = 0 \end{cases} \tag{18}$$

for KDE-based smoothing, and similarly for polynomial smoothing. The sMLMC estimator with smoothing for $F_{\mathbf{h},M}$ is then given by

$$\hat{F}^{\mathrm{sMLsm}}_{\mathbf{h},\delta,M}(\mathbf{q}) = \sum_{v_1=0}^{S_1} \cdots \sum_{v_n=0}^{S_n} \hat{\mathcal{I}}^{\mathrm{sMLsm}}_{\mathbf{v},M} \prod_{i=1}^{n} \phi_{v_i}(q_i), \tag{19}$$

where $\boldsymbol{\delta}$ refers to either $\boldsymbol{\delta}_{\mathrm{G}}$ or $\boldsymbol{\delta}_{\mathrm{K}}$ depending on whether we consider polynomial or KDE-based smoothing, respectively.

To compute $\hat{F}^{\mathrm{sMLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}$ and $\hat{F}^{\mathrm{sMLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{K}},M}$, and to measure the associated computational cost, we deploy the algorithm in Appendix A.4.

2.3. Relative cost of standard and stratified MLMC versus MC

For a given error tolerance ϵ, we estimate the total cost $\mathcal{C}$ of computing the MLMC estimator without smoothing of $F_{\mathbf{h},M}$ as an average over N_{real} independent realizations of the multilevel algorithm,

$$\mathcal{C}(\hat{F}^{\mathrm{ML}}_{\mathbf{h},M}) = \frac{1}{N_{\mathrm{real}}} \sum_{p=1}^{N_{\mathrm{real}}} \sum_{l=0}^{L^{\epsilon}_{\max,p}} \bar{w}^{(p)}_{l} N^{(p)}_{l}, \tag{20}$$

where $\bar{w}^{(p)}_{l}$ is the average cost of computing a sample of $\mathbf{Q}_{M_l}$ on level l for the pth realization, and $L^{\epsilon}_{\max,p}$ denotes the finest discretization level at which the sampling is performed for this realization at tolerance ϵ.

To compare the performance of $\hat{F}^{\mathrm{ML}}_{\mathbf{h},M}$ with that of MC, on the pth run we conduct the latter at the finest discretization level, $L^{\epsilon}_{\max,p}$, to ensure that both algorithms satisfy the given discretization error tolerance, ϵ_{dis}. This procedure is detailed in Appendix A.3 and leads to the cost for $\hat{F}^{\mathrm{MC}}_{\mathbf{h},M}$,

$$\mathcal{C}(\hat{F}^{\mathrm{MC}}_{\mathbf{h},M}) = \frac{1}{N_{\mathrm{real}}} \sum_{p=1}^{N_{\mathrm{real}}} \bar{w}^{(p)}_{L^{\epsilon}_{\max,p}} N^{(p)}_{\mathrm{MC}}, \tag{21}$$

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

For the multilevel variants other than non-smoothed MLMC, we fix the maximum discretization level at the most frequently observed value of $L^{\epsilon}_{\max,p}$ over all N_{real} runs, which we denote by $L^{\epsilon}_{\max}$ (see Section 4 for a detailed explanation). For sMLMC without smoothing, this yields

$$\mathcal{C}(\hat{F}^{\mathrm{sML}}_{\mathbf{h},M}) = \frac{1}{N_{\mathrm{real}}} \sum_{p=1}^{N_{\mathrm{real}}} \sum_{l=0}^{L^{\epsilon}_{\max}} \sum_{k=1}^{r} \bar{w}^{(p)}_{k,l} n^{(p)}_{k,l}, \tag{22}$$

where $\bar{w}^{(p)}_{k,l}$ is the average cost of computing a sample of $\mathbf{Q}_{M_l}$ with corresponding input (ξ) in stratum $\mathcal{D}_k$ on level l for the pth realization.

Expression (21) provides a single average cost against which to compare $\mathcal{C}(\hat{F}^{\mathrm{ML}}_{\mathbf{h},M})$, $\mathcal{C}(\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta},M})$, $\mathcal{C}(\hat{F}^{\mathrm{sML}}_{\mathbf{h},M})$ and $\mathcal{C}(\hat{F}^{\mathrm{sMLsm}}_{\mathbf{h},\boldsymbol{\delta},M})$.

3. Numerical experiments

We consider two testbed problems: fluid flow described by inviscid Burgers' equation with an uncertain initial condition, and ion diffusion in an electrically charged isotropic nanoporous material with an uncertain input parameter.

3.1. One-dimensional inviscid Burgers' equation

Consider an inviscid Burgers' equation

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}(u^2) = 0, \qquad x \in (-5, 5), \quad t > 0 \tag{23a}$$

subject to boundary and initial conditions

$$u(-5,t) = u_{\mathrm{L}}, \quad u(5,t) = u_{\mathrm{R}} \tag{23b}$$

$$u(x,0) = \begin{cases} U_1 & -5 < x \le 0 \\ u_{\mathrm{R}} & 0 < x < 5. \end{cases} \tag{23c}$$

The initial state U_1 is uncertain, while the boundary values u_{L} and u_{R} are known with certainty. Equation (23) represents a double Riemann problem at $x = -5$ and $x = 0$. The solution at time t consists of two right-moving shock waves corresponding to the solutions u_a and u_b given by

$$u_a(x,t) = \begin{cases} u_{\mathrm{L}} & x < s_a t \\ U_1 & x > s_a t \end{cases} \tag{24}$$

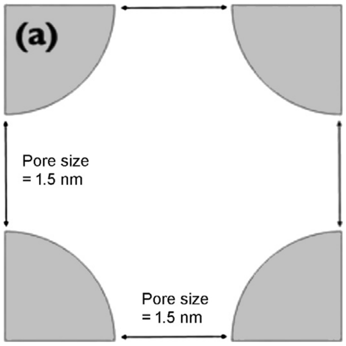


Fig. 1. A unit cell comprising an isotropic nanoporous material with $l_{\text{por}} = 0.75$ nm (half of the pore throat size in each direction). Adapted from [36].

and

$$u_b(x,t) = \begin{cases} U_1 & x < s_b t \\ u_{\text{R}} & x > s_b t, \end{cases} \tag{25}$$

where $s_a = (u_{\text{L}} + U_1)/2$ and $s_b = (U_1 + u_{\text{R}})/2$ are the speeds of the leftmost and rightmost shock, respectively.

The initial state U_1 is drawn from a truncated lognormal PDF with support $\Omega_{U_1} = [u_{1,\ell}, u_{1,r}]$

$$f_{U_1}(u_1; \mu, \sigma, u_{1,\ell}, u_{1,r}) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma u_1} \begin{cases} \dfrac{\exp(-\frac{(\ln u_1 - \mu)^2}{2\sigma^2})}{\text{erf}\left(\frac{\ln u_{1,r} - \mu}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{\ln u_{1,\ell} - \mu}{\sqrt{2}\sigma}\right)} & u_1 \in \Omega_{U_1} \\ 0 & \text{otherwise}, \end{cases} \tag{26}$$

where μ and σ are, respectively, the mean and standard deviation of the parent normal distribution. Our goal is to estimate the CDF F of the total energy in the system at time $t = 1$,

$$Q = \frac{1}{2} \int_{-5}^{5} u^2(x, 1)\, \mathrm{d}x, \tag{27}$$

which depends on U_1.

In the simulations reported below, we set $u_{\text{L}} = 10$, $u_{\text{R}} = 0$, $\mu = 1.5$, $\sigma = 1$, $u_{1,\ell} = 0$, and $u_{1,r} = 10$. We discretize (23) using the Godunov method, which is first-order accurate in both space and time. This is a conservative finite volume scheme, which solves a Riemann problem at each inter-cell boundary.

3.2. Two-dimensional ion diffusion in charged nanoporous materials

The second test deals with ion transport in an electrically charged isotropic nanoporous medium used as a battery electrode [36]. Formation of an electrical double layer (EDL) at solid/electrolyte interfaces affects energy storage capabilities. Optimal design of such nanoporous materials requires the knowledge of a map between microscopic parameters (such as pore geometry and ion concentration in the electrolyte), and macroscopic quantities (such as effective diffusion coefficients and electrolyte conductivities). The latter serve as inputs to macroscopic transport models which treat the nanoporous material as a continuum with no distinction between pores and the solid skeleton.

Following [36], we compute effective diffusion coefficients $\mathbf{D}_{\pm}^{\text{eff}}$ for cations (+) and anions (-) by solving the Poisson equation for the electrical double layer (EDL) potential φ_{EDL} on a unit cell $\mathcal{U}$ representative of the material's pore structure (Fig. 1). Expressed in dimensionless quantities, this is given by [36]

$$\hat{\nabla}^2 \hat{\varphi}_{\text{EDL}} = \frac{\ell^2 \hat{c}_b^\star}{\epsilon^2 \lambda_D^2} \sinh(z\hat{\varphi}_{\text{EDL}}), \quad \hat{\mathbf{y}} \in \hat{\mathcal{P}}_{\mathcal{U}}; \quad \hat{\varphi}_{\text{EDL}} = \hat{\varphi}_\Gamma, \quad \hat{\mathbf{x}} \in \hat{\Gamma}_{\mathcal{U}}. \tag{28}$$

Here $\hat{\varphi}_{\text{EDL}} = \mathcal{F}\varphi_{\text{EDL}}/(RT)$ with the Faraday constant $\mathcal{F} = 96485$ C/mol, the gas constant $R = 8.314$ J/(mol K), and temperature T; $\hat{c}_b^\star = c_b^\star / c_{\text{in}}$ with a characteristic ion concentration in the system, $c_b^\star$, e.g., its initial or average value, and c_{in} the initial ion concentration in the electrolyte; λ_D is the Debye length, a characteristic length of the EDL; ℓ is the characteristic pore size; $\epsilon \equiv \ell/L$ with the characteristic length of the porous material, L; z is the ion valence; $\hat{\varphi}_\Gamma = \mathcal{F}\varphi_\Gamma/(RT)$ is the (dimensionless) boundary potential; and $\hat{\mathcal{P}}_{\mathcal{U}}$ and $\hat{\Gamma}_{\mathcal{U}}$ are the (dimensionless) pore space and fluid-solid interface contained in the unit cell. The effective diffusion tensors $\mathbf{D}_{\pm}^{\text{eff}}$ are obtained via

$$\mathbf{D}_{\pm}^{\text{eff}} = \frac{\mathcal{D}\omega}{G_{\pm}} \int_{\hat{\mathcal{P}}_{\mathcal{U}}} \mathrm{e}^{\mp z \hat{\varphi}_{\text{EDL}}} (\mathbf{I} + \nabla_{\mathbf{y}} \boldsymbol{\chi}_{\pm}^{\top}) \mathrm{d}\mathbf{y}, \quad G_{\pm} = \int_{\hat{\mathcal{P}}_{\mathcal{U}}} \mathrm{e}^{\mp z \hat{\varphi}_{\text{EDL}}} \mathrm{d}\mathbf{y}, \tag{29}$$

where $\mathcal{D}$ is the molecular diffusion coefficient, ω is the porosity, and $\mathbf{I}$ is the identity matrix. The closure variables $\boldsymbol{\chi}_{\pm}(\mathbf{y})$ are $\mathcal{U}$-periodic vector functions computed as solutions of the boundary value problems

$$\nabla_{\mathbf{y}}[\mathrm{e}^{\mp z \hat{\varphi}_{\text{EDL}}} (\mathbf{I} + \nabla_{\mathbf{y}} \boldsymbol{\chi}_{\pm}^{\top})] = \mathbf{0}, \quad \mathbf{y} \in \hat{\mathcal{P}}_{\mathcal{U}}; \tag{30a}$$

$$\mathbf{n}(\mathbf{I} + \nabla_{\mathbf{y}} \boldsymbol{\chi}_{\pm}^{\top}) = \mathbf{0}, \quad \mathbf{y} \in \hat{\Gamma}_{\mathcal{U}}; \tag{30b}$$

$$\int_{\hat{\mathcal{P}}_{\mathcal{U}}} \boldsymbol{\chi}_{\pm} \mathrm{d}\mathbf{y} = \mathbf{0}. \tag{30c}$$

Given the symmetry of $\mathcal{U}$, the diffusion coefficients in (29) are scalars, denoted by D_{+}^{eff} and D_{-}^{eff}.

The effective conductivity of the electrolyte, κ^{eff}, is defined in terms of the cation and anion effective diffusion coefficients as [36]

$$\kappa^{\text{eff}} = \nu z^2 \frac{F^2 c_{\text{in}}}{RT} (D_{+}^{\text{eff}} + D_{-}^{\text{eff}}). \tag{31}$$

To quantify the impact of pore-scale parameters on the effective (macroscopic) QoIs above, we treat the former as random variables that vary over some specified range, and estimate the CDF of the marginal and/or joint distributions of the latter using MLMC. In particular, we compute the joint CDF of the normalized cation effective diffusion coefficient, $D_{+}^{\text{eff,norm}} \equiv D_{+}^{\text{eff}}/\mathcal{D}$, and the effective conductivity of the electrolyte, κ^{eff}. (Half of) the pore throat size, l_{por}, (Fig. 1) serves as the uncertain input parameter with uniform distribution on $[0.375, 1.125]$ nm, which constitutes an interval of ±50% around the base value of 0.75 nm taken from [36]. Equations (28)–(31) are solved using a co-simulation framework of COMSOL Multiphysics® and MATLAB® software packages.

4. Simulation results

We investigate the relative efficiency of sMLMC with and without KDE smoothing, KDE-smoothed MLMC, MLMC with and without polynomial smoothing, and fine-resolution MC. These estimators are labeled below by the abbreviations sMLMCsm, sMLMC, MLMCsm (KDE), MLMCsm (polynomial), MLMC, and MC, respectively. The comparison is reported in terms of their computational cost $\mathcal{C}$ averaged over $N_{\text{real}} = 10$ independent runs for error tolerances $\epsilon = 0.01$, 0.005 and 0.001 (inviscid Burgers'), and averaged over $N_{\text{real}} = 5$ independent runs for error tolerance $\epsilon = 0.03$ (ion diffusion in nanopores). At each level l we first compute N_l^0 ($l = 0, \dots, L_{\max}$) warm-up (aka "pilot") samples to obtain an initial estimate of the indicator function's variance; then, we supplement those with additional samples as needed to satisfy the required sampling error tolerance (Appendix A.3 and Appendix A.4).

The warm-up sampling constitutes an integral part of the overall sampling procedure and does not yield any overhead cost, as long as oversampling is minimized. We do that by tuning N_l^0 for each tolerance and each estimator separately through some initial trial runs; the resulting values are then used for all N_{real} independent runs of the algorithm. For sMLMC and sMLMCsm, we define the number of warm-up samples $n_{k,l}^0$ in each stratum $k = 1, \dots, r$, based on N_l^0 and the proportional allocation strategy; the values of $n_{k,l}^0$ are identical for all independent runs. Representative values of N_l^0 for the two test problems in Section 3 are provided below.

The results reported below are obtained with the following procedure, for each value of ϵ:

1. Do N_{real} runs of MLMC to obtain maximum levels $L_{\max,p}^{\epsilon}$ ($p = 1, \dots, N_{\text{real}}$). Denote the most frequently observed maximum level by $L_{\max}^{\epsilon}$.
2. At the end of the pth run ($p = 1, \dots, N_{\text{real}}$), perform MC at level $L_{\max,p}^{\epsilon}$, re-using already computed samples from the MLMC run.
3. Do N_{real} runs of MLMCsm (polynomial) with a computed smoothing parameter $\boldsymbol{\delta}_{\text{G},l}$ at each level l, fixing the maximum level for each run at $L_{\max}^{\epsilon}$.
4. Do N_{real} runs of MLMCsm (KDE) with a computed smoothing parameter $\boldsymbol{\delta}_{\text{K},l}$ at each level l, with maximum level $L_{\max}^{\epsilon}$.
5. Do N_{real} runs of sMLMC, with maximum level $L_{\max}^{\epsilon}$.
6. Do N_{real} runs of sMLMCsm with a computed smoothing parameter $\boldsymbol{\delta}_{\text{G},l}$ (which may be different from the value for its non-stratified counterpart) at each level l, with maximum level $L_{\max}^{\epsilon}$.

This procedure sets the maximum level for all multilevel variants, other than (non-smoothed) MLMC, to $L_{\max}^{\epsilon}$. That is because the lower number of samples at finer levels (especially, in each stratum on the finer levels of sMLMC/sMLMCsm) may not yield sufficiently accurate estimates of the discretization error; these are based on a sample estimate of $\max_{\mathbf{v} \in \mathcal{S}} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)]|$ in accordance with (A.5). The discretization error is dictated by the maximum discretization level, and

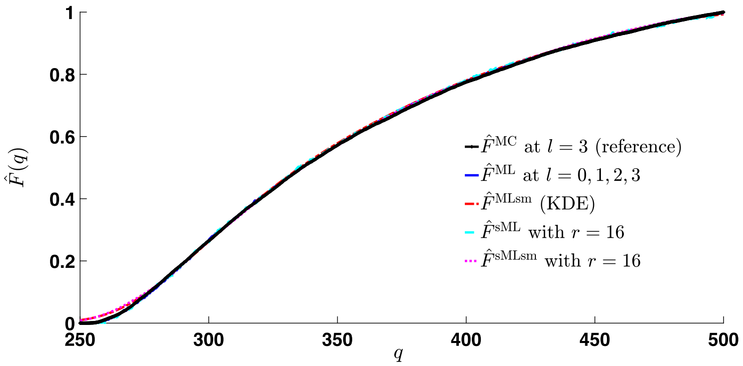


Fig. 2. CDF of Q for the inviscid Burgers' problem computed with a representative set of multilevel estimators for $\epsilon = 0.005$. The reference CDF computed with MC at the highest ($l = 3$) discretization level using 40000 samples is also shown for validation purposes.

Table 1
Number of warm-up samples as a function of level (from coarse to fine going to the right) used for each estimator in the inviscid Burgers' problem at tolerance $\epsilon = 0.005$.

Estimator	Number of warm-up samples
MLMC	20000 3000 2000 1400
MLMCsm (polynomial)	30000 100 40 15
MLMCsm (KDE)	30000 100 30 15
sMLMC 8 strata	2000 1000 800 500
sMLMC 16 strata	1500 800 700 500
sMLMCsm 8 strata	1500 20 8 8
sMLMCsm 16 strata	300 16 16 16

the corresponding discretization tolerance is taken to be identical for all estimators. Therefore, it is reasonable to assume that if the MLMC estimator satisfies the required discretization error tolerance for a certain maximum level $L^{\epsilon}_{\max}$, then the other estimators also satisfy this tolerance when having $L^{\epsilon}_{\max}$ as their maximum level.

For the inviscid Burgers' problem, we use $\alpha = 1/2$, while for the problem of ion diffusion in nanopores, we set $\alpha = 1/3$. This allows for a larger discretization error, and hence fewer levels, when modeling the 2D ion transport relative to the 1D Burgers' case. In both problems, we assume that the PDF of the QoI Q is at least 3 times continuously differentiable so that we can interpolate the point estimates of its CDF F via cubic splines.

The numerical experiments for inviscid Burgers' were performed on an Ubuntu system with 10 cores (20 hyperthreads) running at 4.20 GHz, while those for ion diffusion in nanopores were conducted on an Ubuntu system with 8 cores (16 hyperthreads) running at 2.60 GHz. Both workstations have 64 GB of RAM.

4.1. Inviscid Burgers' equation

The support of the PDF f of the total energy Q is defined by considering the minimum and maximum values of U_1: $Q = 250$ for $U_1 = 0$ and $Q = 500$ for $U_1 = 10$. Hence, the corresponding CDF F is estimated over the interval $[a, b] = [250, 500]$; we use $S + 1 = 501$ interpolation points, i.e., set $h = 0.5$, omitting the relevant subscripts in Section 2 since the QoI Q is a scalar. The computational domain $[-5, 5]$ is discretized via a (tolerance-dependent) hierarchy of spatial grids $\mathcal{T}_{M_l}$ ($l = 0, \dots, L^{\epsilon}_{\max}$), where $M_l = 2M_{l-1}$ and $M_0 = 200$.

To perform a visual quality check of the various multilevel estimators considered in this work, Fig. 2 exhibits, for $\epsilon = 0.005$, the estimated CDF obtained via MLMC, MLMCsm (KDE), sMLMC with 16 strata, and sMLMCsm with 16 strata, along with a reference CDF computed with MC at the highest ($l = 3$) level using 40000 samples (about twice the required number for converging the sampling error tolerance). The smoothed estimators, MLMCsm and sMLMCsm, display a higher discrepancy near the left tail. This illustrates the limitations of using the L^{∞} norm in the variance estimation, which does not allow sufficient control on the error in specific regions of the CDF. In problems where accurate estimation of tails is needed, such as rare event probability estimation, using the L^1 or L^2 norm may be more appropriate.

Table 1 shows the number of warm-up samples as a function of level specified for all CDF estimators at this tolerance. As alluded to above, the number of pilot samples is tuned to minimize oversampling, and is therefore reduced going from the most expensive estimator (MLMC) to the least expensive estimator (sMLMCsm with 16 strata). For the sMLMCsm estimators, we found that the least amount of oversampling occurred when using only one sample per stratum at the finest levels, despite the resulting breakdown of our procedure to estimate strata variances at those levels.

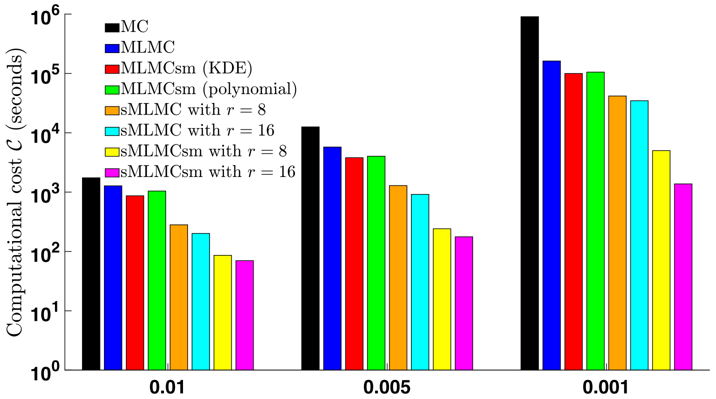


Fig. 3. Computational cost (in seconds) of MC, MLMC, MLMCsm (KDE and polynomial), and sMLMC and sMLMCsm with 8 or 16 strata for the inviscid Burgers' problem. The cost values are shown for tolerances $\epsilon = 0.1$, 0.005, and 0.001. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

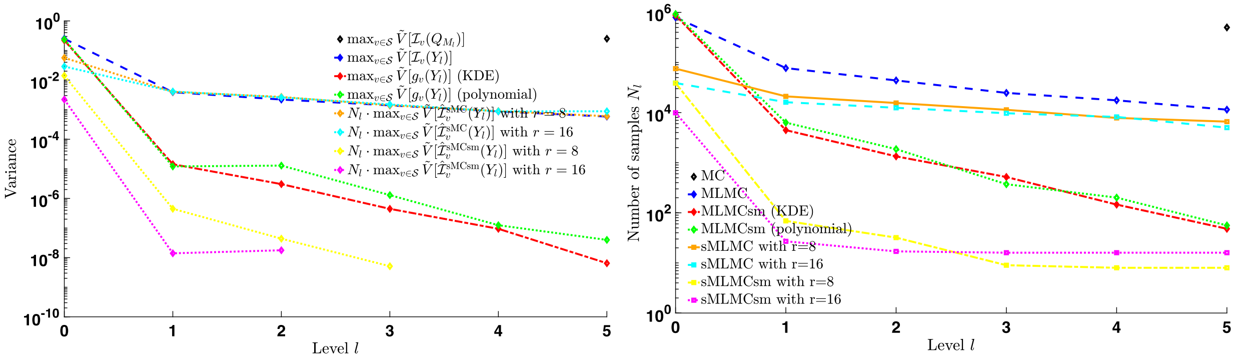


Fig. 4. Evolution of the variance (left) and number of samples (right) with level for a single run of the inviscid Burgers' problem and $\epsilon = 0.001$.

Fig. 3 collates the computational cost of the various multilevel estimators, and fine-resolution MC. For all tolerances considered, MLMCsm (KDE) is slightly more efficient than its counterpart smoothed with a third-degree polynomial. In [29], a higher polynomial degree was shown to increase the optimal smoothing bandwidth δ_{G} and to yield a more efficient multilevel estimator. However, our tests show that using a ninth-degree rather than a third-degree polynomial leads to a higher computational cost; e.g., for $\epsilon = 0.001$, smoothing with the former costs $\mathcal{C} = 1.1779 \cdot 10^5$ s while for the latter $\mathcal{C} = 1.0525 \cdot 10^5$ s (averaged over $N_{\mathrm{real}} = 10$ independent runs). These findings suggest that polynomial smoothing might necessitate the inclusion of an additional loop over different polynomial degrees in the multilevel algorithm to find the optimal polynomial order yielding the most efficient estimator. However, doing so would further increase computational complexity. KDE-based smoothing only requires optimization over the smoothing bandwidth δ_{K}. Moreover, polynomial smoothing puts more stringent requirements on the PDF f of Q: a polynomial of degree $d+1$ requires f to be d times continuously differentiable. We conclude that KDE-based smoothing constitutes a more efficient and robust approach. It is used in all subsequent simulations related to our stratified multilevel estimators.

The computational savings gained by smoothing the indicator function are rather modest in this test case (Fig. 3). Stratifying Ω_{U_1}, the sample space of U_1, into 8 or 16 strata of equal width leads to a substantially higher reduction in computational cost than smoothing does. When combined with KDE-based smoothing, the savings are further increased and amount to two orders of magnitude for the sMLMCsm estimator with 16 strata at the lowest tolerance of 0.001. Increasing the number of strata invariably reduces the computational cost, but the maximum number of strata may be limited by the (low) number of samples at the finer levels.

The computational savings discussed above can be attributed to the variance decay of the indicator function or its approximation by a smoothing function. Fig. 4 (left) illustrates the dependence on level l of $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Q_{M_l})]$, $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Y_l)]$ and $\max_{v\in\mathcal{S}} \tilde{V}[g_v(Y_l)]$ (for both polynomial and KDE-based smoothing), for a single run and $\epsilon = 0.001$. Here $\tilde{V}$ denotes a sample estimate of $\mathbb{V}$, and $\mathcal{S} = \{0, \ldots, S\}$. While $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Q_{M_l})]$ remains approximately constant as the spatial resolution increases, $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Y_l)]$ and $\max_{v\in\mathcal{S}} \tilde{V}[g_v(Y_l)]$ decay as the spatial mesh is refined, so that fewer samples are needed at higher levels of discretization (Fig. 4, right). The decay of $\max_{v\in\mathcal{S}} \tilde{V}[g_v(Y_l)]$ is faster than that of $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Y_l)]$, making the MLMCsm estimators more efficient than their non-smoothed counterpart.

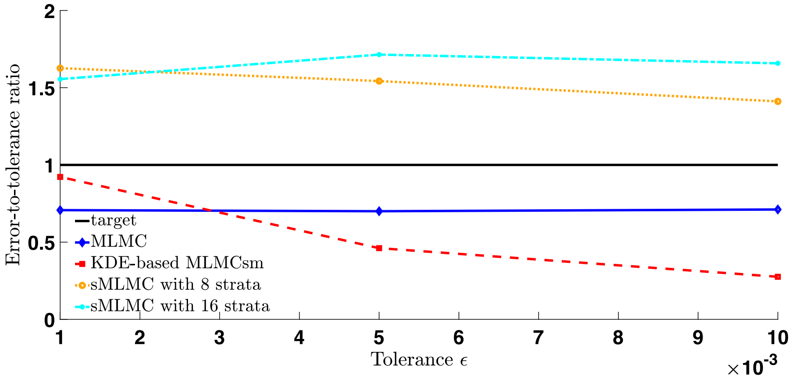


Fig. 5. Values of the ratio $\varepsilon_{\text{est}}/\epsilon$ for MLMC, the ratio $\varepsilon_{\text{sam}}/\epsilon_{\text{sam}}$ for sMLMC, and the ratio $(\varepsilon_{\text{sam}} + \varepsilon_{\text{smooth}})/(\epsilon_{\text{sam}} + \epsilon_{\text{smooth}})$ for MLMCsm (KDE), at all considered tolerances ϵ for the inviscid Burgers' problem.

To define an equivalent to $\max_{v\in\mathcal{S}} \tilde{V}[\mathcal{I}_v(Y_l)]$ for sMLMC, we compare the variance contribution at each level l for MLMC, $N_l^{-1}\max_{v\in\mathcal{S}} \mathbb{V}[\mathcal{I}_v(Y_l)]$, to (13) (replacing Q with Y), and hence consider $N_l \cdot \max_{v\in\mathcal{S}} \mathbb{V}[\hat{\mathcal{I}}_v^{\text{sMC}}(Y_l)]$. Proportional allocation only holds at the warm-up sampling stage and hence becomes an approximation after warm-up is completed. A similar equivalency is defined for the smoothed indicator function. Fig. 4 (left) demonstrates that, for sMLMC, the variance at the coarsest level is an order of magnitude lower than its counterpart for MLMCsm, but decays slower with increasing level. This translates into a much lower number of coarse samples but a higher number of finer samples for sMLMC compared to MLMCsm. Overall, this still lowers the computational cost of the former compared to the latter. Addition of KDE-based smoothing to sMLMC is the most optimal strategy with strong variance reduction (and hence a lower number of samples) at all levels, yielding a much cheaper estimator. The number of samples at the finest levels for both sMLMCsm estimators is so low that their variance estimates are omitted in Fig. 4 (left).

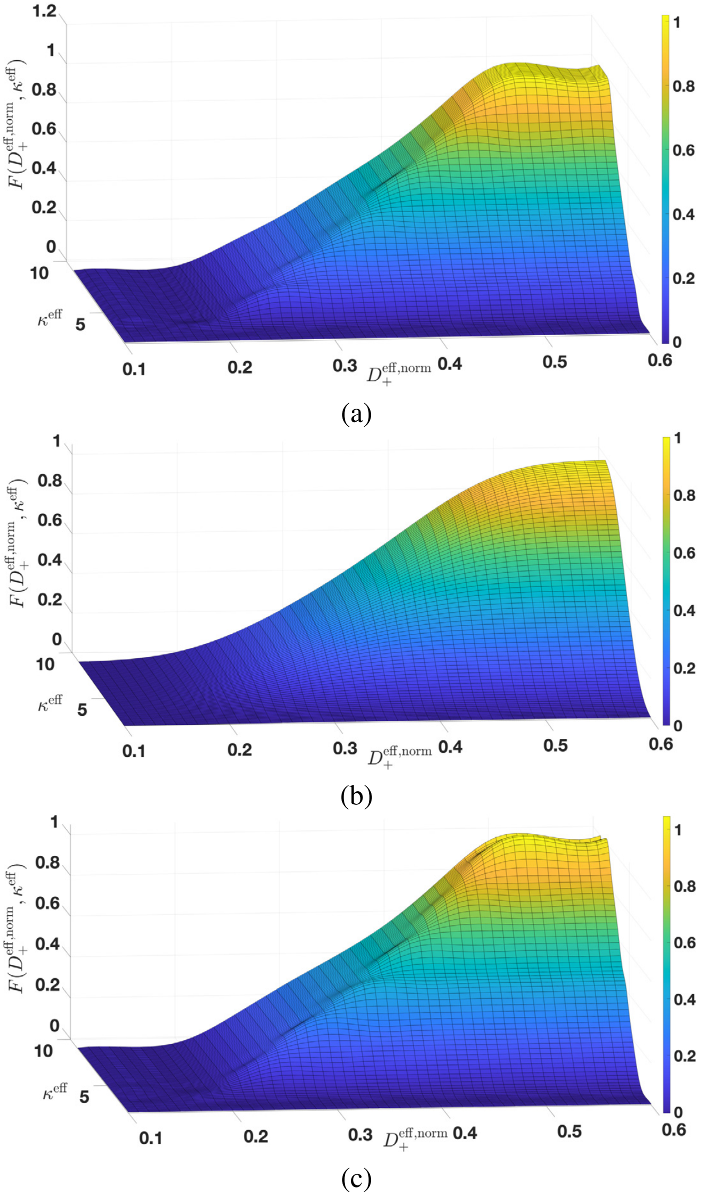
Finally, to illustrate the ability of our CDF estimators to satisfy the required error tolerance, one could compute the ratio $\varepsilon_{\text{est}}/\epsilon$, averaged over N_{real} runs of the algorithm. However, as discussed at the beginning of this Section, a low number of samples at the finest level might impact the accuracy of the sample estimate of the root mean square discretization error, ε_{dis}. Therefore, we use the maximum discretization level from the MLMC estimator for all other multilevel variants as well. Fig. 5 shows the ratio of *total* root mean square error to *total* tolerance only for MLMC: the value of $\varepsilon_{\text{est}}/\epsilon < 1$ for all tolerances ϵ considered. For the other multilevel estimators, we leave out the discretization portion of the total error and compare the resulting error to the corresponding fraction of the total tolerance. While applying the formulae for computing the optimum number of samples in each stratum (Appendix A.4) should theoretically allow the sMLMC estimator to satisfy the sampling error tolerance, Fig. 5 yields a ratio $\varepsilon_{\text{sam}}/\epsilon_{\text{sam}} \approx 1.5$. This result should be interpreted with caution as it is difficult to estimate strata variances with a low number of samples in each stratum. Smoothing exacerbates this problem, and hence we omitted the corresponding result for the sMLMCsm estimators.

4.2. Effective diffusion in nanoporous materials

We estimate the joint CDF F of $Q_1 = D_+^{\text{eff,norm}}$ and $Q_2 = \kappa^{\text{eff}}$ defined in Section 3.2; half of the pore throat size in each direction, l_{por}, serves as an uncertain input. We approximate F with tolerance $\epsilon = 0.03$ over the domain $[0.1, 0.6] \times [2, 9]$. Its size is determined by first doing an MLMC run using only the coarsest level, over $[0, 1] \times [0, 10]$; refining this initial estimate based on the values of Q_{1,M_0} and Q_{2,M_0} for which the CDF approached 0 and 1; and applying a safety factor to allow for samples at higher levels to lie slightly outside this area. The initial lower bounds are based on the physical constraint that Q_1 and Q_2 are positive. The initial upper bound for Q_1 follows from the definition of $Q_1 = D_+^{\text{eff}}/\mathcal{D}$ and the realization that the cation effective diffusion coefficient D_+^{eff} is smaller than the molecular diffusion coefficient $\mathcal{D}$. The initial upper bound for Q_2 is estimated from numerical experiments based on [36]. We use $(S_1 + 1) \times (S_2 + 1) = 11 \times 15$ interpolation points, i.e., set $h_1 = 0.05$ and $h_2 = 0.5$. The computational domain is discretized via a hierarchy of unstructured spatial grids $\mathcal{T}_{M_l}$ $(l = 0, \ldots, 3)$ obtained via Delaunay triangulation. In COMSOL, the user can specify the mesh resolution with a number ranging from 9 ("extremely coarse") to 1 ("extremely fine"). We set the level $l = 0$ to resolution "4" in COMSOL. Finally, given our findings for the inviscid Burgers equation, we use only KDE-based smoothing.

Fig. 6 depicts representative plots of the estimated CDFs obtained using MLMC, MLMCsm (KDE) and sMLMC with $r = 2$. As in the Burgers' case, the larger error is near the tail area for the smoothed estimator. The sMLMC estimator does not suffer from this issue, while offering comparable efficiency (Fig. 7). Details on the warm-up sampling, aimed at minimizing oversampling, are collated in Table 2.

Fig. 7 compares the computational cost of MLMC, MLMCsm, sMLMC and sMLMCsm with that of MC performed at the finest discretization level. While MLMC is less efficient than MC, applying KDE-based smoothing yields a multilevel estimator that is faster than its MC counterpart. This demonstrates that for non-scalar QoIs, in the absence of other variance reduction



(a)

(b)

(c)

Fig. 6. Representative plots of the estimated CDFs computed with MLMC (a), MLMCsm (KDE) (b), and sMLMC with 2 strata (c).

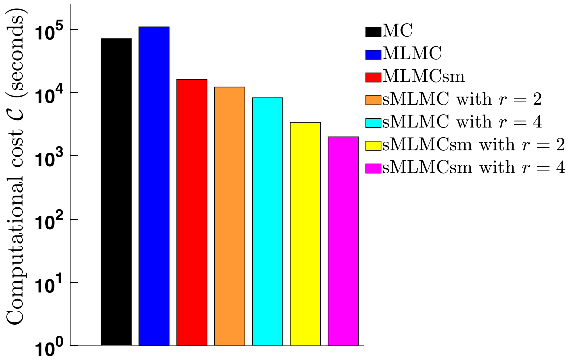


Fig. 7. Computational cost (in seconds) of MC, MLMC, MLMCsm, and sMLMC and sMLMCsm with 2 strata for the problem of ion diffusion in nanopores. The cost values are shown for tolerance $\epsilon = 0.03$.

Table 2
Number of warm-up samples as a function of level (from coarse to fine going to the right) employed for each estimator in the problem of ion diffusion in nanopores.

Estimator	Number of warm-up samples
MLMC	800 300 200 100
MLMCsm KDE	800 30 20 10
sMLMC 2 strata	400 190 140 2
sMLMC 4 strata	200 130 100 4
sMLMCsm 2 strata	250 20 10 2
sMLMCsm 4 strata	100 10 4 4

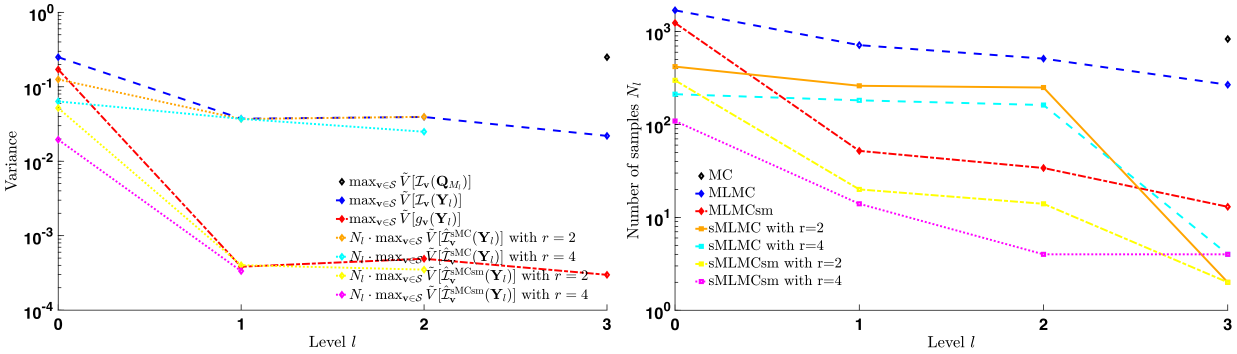


Fig. 8. Dependence of the variance (a) and the number of samples (b) on level l, for a single run of the ion diffusion problem.

techniques, smoothing the indicator function may be essential for a multilevel algorithm to outperform fine-resolution MC, especially at $\mathcal{O}(10^{-2})$ tolerances typical for real-world engineering applications.

Stratification of the sample space of l_{hor} with $r = 2$ yields an estimator with a slightly lower computational cost than its MLMCsm counterpart. Fig. 8 (left) shows that stratification reduces the variance mostly at the coarsest level, leading to a corresponding reduction in the number of samples at that level (Fig. 8, right). The increase in the number of strata from 2 to 4 magnifies this effect, reduces the computational cost by an order of magnitude, compared to fine-resolution MC. This finding demonstrates that the benefits of a stratified approach generalize to vector-valued QoIs. Combining stratification with KDE-based smoothing of the indicator function provides the strongest level of variance (and hence computational cost) reduction (Fig. 7 and Fig. 8, left).

5. Conclusions and future work

We constructed a stratified multilevel Monte Carlo (sMLMC) algorithm for estimating the cumulative distribution function (CDF) of a quantify of interest (QoI) in a problem with a random initial state or random input parameters. Our method combines the benefits of multigrid from standard multilevel Monte Carlo (MLMC) with variance reduction from stratified sampling in each of the discretization levels. We also explored the use of Gaussian Kernel Density Estimator (KDE) in lieu of the currently used polynomial-based smoothers.

Our study yields the following major conclusions:

1. For all test problems and error tolerances considered, the computational cost of non-smoothed sMLMC is smaller than that of non-smoothed MLMC. This is due to the steeper decay of the variance, predominantly at the coarsest levels, in sMLMC due to the stratification of the input sample space.
2. For the same smoothing error tolerance, KDE-based smoothing is more efficient than its polynomial-based counterpart, especially considering the fact only one parameter (bandwidth) instead of two (bandwidth and polynomial degree) needs to be tuned.
3. For estimating joint CDFs of non-scalar QoIs, smoothing of the indicator function is found to be of crucial importance to making MLMC more efficient than MC performed at the finest discretization level.
4. For both the scalar and non-scalar QoIs, stratification of the input sample space yields a higher reduction in computational cost than smoothing of the indicator function.
5. Combining indicator function smoothing and input stratification yields the most efficient estimator, providing order-of-magnitude computational cost savings.

This study dealt with a single random input variable. A follow-up study will extend our stratified MLMC algorithm to problems with high stochastic dimension, wherein a QoI depends on a correlated random input field. Further efforts are needed to characterize the tails of a distribution, which is important, e.g., when the probability of failure needs to be estimated. This may require a number of modifications to our existing algorithm to better control the error in specific regions of a CDF. The L^∞ norm may fail to detect errors in areas such as tails; the L^1 or L^2 norm might be more appropriate. Finally, since fewer samples are drawn at higher discretization levels, future iterations of our algorithm will allow for the number of discrete points (bins) at which the CDF is estimated to be level-dependent. This would avoid having too few samples in some of the CDF bins at the finest levels.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by Defense Advanced Research Projects Agency under award number 101513612, by Air Force Office of Scientific Research under award number FA9550-17-1-0417, by U.S. Department of Energy under award number DE-SC0019130, and by a gift from TOTAL.

Appendix A. Derivation and computation of estimators $\hat{F}^{\text{MLsm}}_{\boldsymbol{h},\delta,M}$ and $\hat{F}^{\text{sMLsm}}_{\boldsymbol{h},\delta,M}$

A.1. Standard multilevel Monte Carlo without smoothing

The MC estimator for $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ based on N_{MC} independent samples of $\mathbf{Q}_M$ is defined by

$$\hat{\mathcal{I}}^{\text{MC}}_{\mathbf{v},M} = \frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M^{(j)}), \tag{A.1}$$

where $\mathbf{Q}_M^{(j)}$ is the jth sample of $\mathbf{Q}_M$. Rather than sampling $\mathbf{Q}_M$ on a single spatial mesh, one may consider a sequence of approximations $\mathbf{Q}_{M_l}$ $(l = 0, \dots, L_{\max})$ of $\mathbf{Q}$ associated with discrete meshes $\{\mathcal{T}_{M_l}, l = 0, \dots, L_{\max}\}$. Here M_l is the number of cells in mesh $\mathcal{T}_{M_l}$, which increases with level l (a common rule for rectilinear grids is $M_{l-1} = 2^{-d} M_l$, where d is the spatial dimension). The idea behind this approach is to start by performing cheap-to-compute samples on a coarse mesh, and then gradually correct the resulting estimate of $F_{\mathbf{h},M}$ by sampling on finer grids, where generating a realization is more computationally expensive. One rewrites $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ as a telescopic sum

$$\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)] = \mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_0})] + \sum_{l=1}^{L_{\max}} \mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_l}) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{l-1}})] \equiv \sum_{l=0}^{L_{\max}} \mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)], \tag{A.2a}$$

where $\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)$ are given by

$$\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l) = \begin{cases} \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_l}) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{l-1}}) & 1 \le l \le L_{\max} \\ \mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_l}) & l = 0, \end{cases} \tag{A.2b}$$

and $M_{L_{\max}} \equiv M$. The non-smoothed multilevel Monte Carlo (MLMC) estimator for $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ is defined as

$$\hat{\mathcal{I}}^{\text{ML}}_{\mathbf{v},M} = \sum_{l=0}^{L_{\max}} \hat{\mathcal{I}}^{\text{MC}}_{\mathbf{v}}(\mathbf{Y}_l) = \sum_{l=0}^{L_{\max}} \frac{1}{N_l} \sum_{j=1}^{N_l} \mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l^{(j)}). \tag{A.3}$$

(We use the shorthand notation ML to refer to MLMC in estimator expressions.) While $\mathbb{V}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_l})]$ remains approximately constant with l, $\mathbb{V}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)]$ decreases with l, allowing the estimator $\hat{\mathcal{I}}^{\text{ML}}_{\mathbf{v},M}$ to have the same overall sampling error as its MC counterpart $\hat{\mathcal{I}}^{\text{MC}}_{\mathbf{v},M}$ using a decreasing number of samples N_l as l increases.

From (4) and (A.3), it follows that the MSE of the non-smoothed MLMC estimator $\hat{F}^{\text{ML}}_{\mathbf{h},M}$ for $F_{\mathbf{h},M}$ is bounded by

$$\begin{aligned} \underbrace{\mathbb{E}[\|F_{\mathbf{h}} - \hat{F}^{\text{ML}}_{\mathbf{h},M}\|_\infty^2]}_{\varepsilon^2_{\text{est}}} &\le \mathbb{E}[\|\hat{F}^{\text{ML}}_{\mathbf{h},M} - \mathbb{E}[\hat{F}^{\text{ML}}_{\mathbf{h},M}]\|_\infty^2] + \|F_{\mathbf{h},M} - F_{\mathbf{h}}\|_\infty^2 \\ &\le \underbrace{\max_{\mathbf{v}\in\mathcal{S}} \sum_{l=0}^{L_{\max}} N_l^{-1} \mathbb{V}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)]}_{(\varepsilon^{\text{ML}}_{\text{sam}})^2} + \underbrace{\max_{\mathbf{v}\in\mathcal{S}} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{L_{\max}}}) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q})]|^2}_{(\varepsilon^{\text{ML}}_{\text{dis}})^2}. \end{aligned} \tag{A.4}$$

To achieve a root mean squared error (RMSE) $\varepsilon_{\mathrm{est}}$ of at most ϵ, we may choose $(\varepsilon^{\mathrm{ML}}_{\mathrm{dis}})^2 \le (1-\alpha)\epsilon^2$ and $(\varepsilon^{\mathrm{ML}}_{\mathrm{sam}})^2 \le \alpha\epsilon^2$, where $0 < \alpha < 1$. From the triangle inequality, it follows that

$$\max_{\mathbf{v}\in\mathcal{S}} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_{L_{\max}})| \approx \max_{\mathbf{v}\in\mathcal{S}} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{L_{\max}}}) - \mathcal{I}_{\mathbf{v}}(\mathbf{Q})]|. \tag{A.5}$$

Hence, to determine the maximum level $L_{\max}$ of an MLMC simulation with given tolerance ϵ, we check if $\max_{\mathbf{v}\in\mathcal{S}} |\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_L)| \le \epsilon\sqrt{1-\alpha}$ is satisfied for the current level L. If this holds, then $L_{\max} = L$ and we can compare the performance of MLMC to MC by performing the latter on this finest level, re-using the samples already computed with MLMC at this level.

A.2. Standard multilevel Monte Carlo with polynomial smoothing

The jump discontinuity in the indicator function may lead to a slow decay of $\mathbb{V}[\mathcal{I}_{\mathbf{v}}(\mathbf{Y}_l)]$ and make MLMC slower than MC for sufficiently large values of the error tolerance ϵ [29]. To accelerate the variance decay and improve the computational efficiency of MLMC, a sigmoid-type smoothing function g can be used to remove the singularity in the indicator function.

For a scalar QoI Q (i.e., $n = 1$), Giles et al. [25] suggested replacing the indicator function $\mathcal{I}_{(-\infty,q]}(Q_{M_l})$ for $q \in \mathcal{S}_h$ (we leave out the subscripts for notational convenience) by a polynomial $g_{\mathrm{G}}((Q_{M_l} - q)/\delta_{\mathrm{G},l})$ at each level l ($l = 0, \ldots, L_{\max}$), where the *bandwidth* $\delta_{\mathrm{G},l}$ is a measure of the width over which the discontinuity in $\mathcal{I}_{(-\infty,q]}(Q_{M_l})$ is smeared out. Generalizing to the case of n dimensions, the smoothed MLMC estimator for $\mathbb{E}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_M)]$ is given by

$$\hat{\mathcal{I}}^{\mathrm{MLsm}}_{\mathbf{v},M} = \sum_{l=0}^{L_{\max}} \hat{\mathcal{I}}^{\mathrm{MCsm}}_{\mathbf{v}}(\mathbf{Y}_l), \tag{A.6}$$

where $\hat{\mathcal{I}}^{\mathrm{MCsm}}_{\mathbf{v}}(\mathbf{Y}_l) = N_l^{-1} \sum_{j=1}^{N_l} g_{\mathbf{v}}(\mathbf{Y}_l^{(j)})$ with

$$g_{\mathbf{v}}(\mathbf{Y}_l^{(j)}) = \begin{cases} \prod_{i=1}^{n} g_{\mathrm{G}}\left(\dfrac{q_{i,v_i} - Q^{(j)}_{i,M_l}}{\delta_{\mathrm{G},l,i}}\right) - g_{\mathrm{G}}\left(\dfrac{q_{i,v_i} - Q^{(j)}_{i,M_{l-1}}}{\delta_{\mathrm{G},l,i}}\right) & 1 \le l \le L_{\max} \\ \prod_{i=1}^{n} g_{\mathrm{G}}\left(\dfrac{q_{i,v_i} - Q^{(j)}_{i,M_l}}{\delta_{\mathrm{G},l,i}}\right) & l = 0 \end{cases} \tag{A.7}$$

and the superscript MLsm refers to "smoothed MLMC". As (A.7) indicates, we allow for the smoothing bandwidth at level l, $\delta_{\mathrm{G},l}$, to be different in each dimension.

Defining $\boldsymbol{\delta}_{\mathrm{G},l} \equiv (\delta_{\mathrm{G},l,1}, \ldots, \delta_{\mathrm{G},l,n})^\top$ for notational convenience, the polynomial-smoothed MLMC estimator $\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}$ for $F_{\mathbf{h},M}$ is then given by a multidimensional piecewise polynomial interpolation

$$\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}(\mathbf{q}) = \sum_{v_1=0}^{S_1} \cdots \sum_{v_n=0}^{S_n} \hat{\mathcal{I}}_{\mathbf{v},M} \prod_{i=1}^{n} \phi_{v_i}(q_i). \tag{A.8}$$

The MSE of $\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}$ is bounded by

$$\underbrace{\mathbb{E}[\|F_{\mathbf{h}} - \hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}\|^2_\infty]}_{\varepsilon^2_{\mathrm{est}}} \le \underbrace{\mathbb{E}[\|\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M} - \mathbb{E}[\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}]\|^2_\infty]}_{(\varepsilon^{\mathrm{MLsm}}_{\mathrm{sam}})^2} + \underbrace{\|F_{\mathbf{h},M} - F_{\mathbf{h}}\|^2_\infty}_{(\varepsilon^{\mathrm{MLsm}}_{\mathrm{dis}})^2} \tag{A.9}$$

$$+ \underbrace{\|\mathbb{E}[\hat{F}^{\mathrm{MLsm}}_{\mathbf{h},\boldsymbol{\delta}_{\mathrm{G}},M}] - F_{\mathbf{h},M}\|^2_\infty}_{(\varepsilon^{\mathrm{MLsm}}_{\mathrm{smooth}})^2}.$$

Compared to (A.4), (A.9) contains an additional term $(\varepsilon^{\mathrm{MLsm}}_{\mathrm{smooth}})^2$, which is the (mean squared) smoothing error. To achieve an RSME $\varepsilon_{\mathrm{est}}$ of at most ϵ, we may choose $(\varepsilon^{\mathrm{MLsm}}_{\mathrm{dis}})^2 \le (1-\alpha)\epsilon^2$, $(\varepsilon^{\mathrm{MLsm}}_{\mathrm{sam}})^2 \le (\alpha/2)\epsilon^2$ and $(\varepsilon^{\mathrm{MLsm}}_{\mathrm{smooth}})^2 \le (\alpha/2)\epsilon^2$, where $0 < \alpha < 1$. We choose the tolerance for $\varepsilon^{\mathrm{MLsm}}_{\mathrm{dis}}$ to be the same as that for $\varepsilon^{\mathrm{ML}}_{\mathrm{dis}}$ such that the smoothed MLMC estimator satisfies the discretization error tolerance for the same number of levels as its non-smoothed counterpart.

A.3. MLMC with smoothing

Algorithm 1: Standard multilevel Monte Carlo with smoothing.

```
Input      : RMSE tolerance $\epsilon$; set of $\prod_{i=1}^{n}(S_i+1)$ interpolation points $\mathcal{S}_{\mathbf{h}}$; sequence of discrete meshes $\{\mathcal{T}_{M_l}, l=0,\dots,L_{\max}\}$; initial number of
             samples $N_l^0$ at each level $l$; parameter $\alpha$;
Output     : An estimate of the CDF $F(\mathbf{q})$;
Procedure: :
Initialize $L=-1$;
while $L<L_{\max}$ do
    Set $L=L+1$;
    Draw $N_L^0$ samples of the random input parameter (IP)/initial condition (IC) ($\star$);
    if $L=0$ then
        Compute $N_0^0$ samples of $\mathbf{Q}_{M_0}$ based on ($\star$);
    else
        Compute $N_L^0$ samples of $\mathbf{Q}_{M_L}$ and $\mathbf{Q}_{M_{L-1}}$ based on ($\star$);
    end
    Compute $\boldsymbol{\delta}_L$;
    for $\mathbf{v}\in\mathcal{S}$ do
        for $j=1,\dots,N_L^0$ do
            Compute $g_{\mathbf{v}}(\mathbf{Y}_L^{(j)})$;
        end
    end
    Compute the computational cost at level $L$, $\bar{w}_L$;
    for $\mathbf{v}\in\mathcal{S}$ do
        Compute $\hat{\mathcal{I}}_{\mathbf{v}}^{\mathrm{MC}}(\mathbf{Y}_L)$ and $\hat{\mathcal{I}}_{\mathbf{v}}^{\mathrm{MCsm}}(\mathbf{Y}_L)$;
        Compute $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L)]=\sum_{j=1}^{N_L^0}(g_{\mathbf{v}}(\mathbf{Y}_L^{(j)})-\hat{\mathcal{I}}_{\mathbf{v}}^{\mathrm{MCsm}}(\mathbf{Y}_L))^2/N_L^0$
    end
    See next page
end
```

```
Set $N_L = \text{ceil}\left(\max_{\mathbf{v}\in\mathcal{S}} \frac{2}{\alpha\epsilon^2}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L)]/\bar{w}_L}\left(\sum_{z=0}^{L}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_z)]\bar{w}_z}\right)\right)$;
Draw $\max(N_L - N_L^0, 0)$ samples of the random IP/IC (†);
if $L = 0$ then
    Compute $\max(N_0 - N_0^0, 0)$ samples of $\mathbf{Q}_{M_0}$ based on (†);
else
    Compute $\max(N_L - N_L^0, 0)$ samples of $\mathbf{Q}_{M_L}$ and $\mathbf{Q}_{M_{L-1}}$ based on (†);
end
for $\mathbf{v} \in \mathcal{S}$ do
    for $j = N_L^0 + 1, \ldots, N_L$ do
        Compute $g_{\mathbf{v}}(\mathbf{Y}_L^{(j)})$;
    end
end
Compute the computational cost at level $L$, $\bar{w}_L$;
for $\mathbf{v} \in \mathcal{S}$ do
    Compute $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{MC}}(\mathbf{Y}_L)$ and $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{MCsm}}(\mathbf{Y}_L)$;
    Compute $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L)] = \sum_{j=1}^{N_L}(g_{\mathbf{v}}(\mathbf{Y}_L^{(j)}) - \hat{\mathcal{I}}_{\mathbf{v}}^{\text{MCsm}}(\mathbf{Y}_L))^2/N_L$;
end
Set $N_L^\star = N_L$;
for $l = 0, \ldots, L-1$ do
    Set $N_l = \text{ceil}\left(\max_{0\le n\le S} \frac{2}{\alpha\epsilon^2}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_l)]/\bar{w}_l}\left(\sum_{z=0}^{L}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_z)]\bar{w}_z}\right)\right)$;
    Draw $\max(N_0 - N_0^\star, 0)$ samples of the random IP/IC (‡);
    if $l = 0$ then
        Compute $\max(N_0 - N_0^\star, 0)$ samples of $\mathbf{Q}_{M_0}$ based on (‡);
    else
        Compute $\max(N_l - N_l^\star, 0)$ samples of $\mathbf{Q}_{M_l}$ and $\mathbf{Q}_{M_{l-1}}$ based on (‡);
    end
    for $\mathbf{v} \in \mathcal{S}$ do
        for $j = N_l^\star + 1, \ldots, N_l$ do
            Compute $g_{\mathbf{v}}(\mathbf{Y}_l^{(j)})$;
        end
    end
    Compute the computational cost at level $l$, $\bar{w}_l$;
    for $\mathbf{v} \in \mathcal{S}$ do
        Compute $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{MC}}(\mathbf{Y}_l)$ and $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{MCsm}}(\mathbf{Y}_l)$;
        Compute $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_l)] = \sum_{j=1}^{N_l}(g_{\mathbf{v}}(\mathbf{Y}_l^{(j)}) - \hat{\mathcal{I}}_{\mathbf{v}}^{\text{MCsm}}(\mathbf{Y}_l))^2/N_l$;
    end
    Set $N_l^\star = N_l$;
end
if $(L \ge 1$ and $\max_{\mathbf{v}\in\mathcal{S}} |\hat{\mathcal{I}}_{\mathbf{v}}^{\text{MC}}(\mathbf{Y}_L)| \le \sqrt{1-\alpha}\,\epsilon)$ or $(L = L_{\max})$ then
    Compute the cost of MLMC, $\mathcal{C}(\hat{F}_{\mathbf{h},\boldsymbol{\delta},M}^{\text{MLsm}})$;
    Compute the MLMC estimator of $F(\mathbf{q})$, $\hat{F}_{\mathbf{h},\boldsymbol{\delta},M}^{\text{MLsm}}(q)$;
    Set $N_{\text{MC}} = 2\epsilon^{-2}\max_{\mathbf{v}\in\mathcal{S}} \tilde{V}[\mathcal{I}_{\mathbf{v}}(\mathbf{Q}_{M_{L_{\max}}})]$;
    Compute the cost of MC, $\mathcal{C}(\hat{F}_{\mathbf{h},M}^{\text{MC}})$;
    Compute $\max(N_{\text{MC}} - N_{L_{\max}}, 0)$ samples of $\mathbf{Q}_{M_{L_{\max}}}$;
    Compute the MC estimator of $F(\mathbf{q})$, $\hat{F}_{\mathbf{h},M}^{\text{MC}}(\mathbf{q})$;
end
```

A.4. sMLMC with smoothing

Algorithm 2: Stratified MLMC with smoothing.

```
Input     : Input parameters of Algorithm in Section A.3; the strata $\mathcal{D}_k$ with $k = 1, \ldots, r$; and their probabilities $p_k$;
Output    : An estimate of the CDF $F(\mathbf{q})$;
Procedure:
Initialize $L = -1$;
while $L < L_{\max}$ do
    Set $L = L + 1$;
    For $k = 1, \ldots, r$, define initial number of samples in stratum $k$ at level $L$, $n^0_{k,L}$, based on $N^0_L$ and chosen allocation strategy, and draw $n^0_{k,L}$
     samples of random IP/IC from stratum $k$ ($\star$);
    if $L = 0$ then
        for $k = 1, \ldots, r$ do
            Compute $n^0_{k,0}$ samples of $\mathbf{Q}_{M_0}$ based on ($\star$);
        end
    else
        for $k = 1, \ldots, r$ do
            Compute $n^0_{k,L}$ samples of $\mathbf{Q}_{M_L}$ and $\mathbf{Q}_{M_{L-1}}$ based on ($\star$);
        end
    end
    Compute $\boldsymbol{\delta}_L$ with the combined samples from all strata;
    for $k = 1, \ldots, r$ do
        for $j = 1, \ldots, n^0_{k,L}$ do
            for $\mathbf{v} \in \mathcal{S}$ do
                Compute $g_{\mathbf{v}}(\mathbf{Y}_L^{(j,k)})$;
            end
        end
        Compute sample variance estimate $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L^{(k)})]$ of $\mathbb{V}[g_{\mathbf{v}}(\mathbf{Y}_L^{(k)})]$;
    end
    for $k = 1, \ldots, r$ do
        Compute the average computational cost per sample in stratum $k$ at level $L$, $\bar{w}_{k,L}$;
    end
    See next page
end
Set $n^\star_{k,L} = n_{k,L}$ for all $k = 1, \ldots, r$;
for $\mathbf{v} \in \mathcal{S}$ do
    for $k = 1, \ldots, r$ do
        Compute $n_{k,L,n} = \frac{2}{\alpha\epsilon^2} \sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L^{(k)})] p_k^2 / \bar{w}_{k,L}} \sum_{z=0}^{L} \sum_{k=1}^{r} \sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_z^{(k)})] p_k^2 \bar{w}_{k,z}}$;
    end
end
for $k = 1, \ldots, r$ do
    Set $n_{k,L} = \mathrm{ceil}\left(\max_{\mathbf{v} \in \mathcal{S}} n_{k,L,n}\right)$;
end
Set $N_L = \sum_{k=1}^{r} n_{k,L}$;
See next page
```

```
Draw $n_{k,L} - n^0_{k,L}$ samples of random IP/IC from stratum $k$;
if $L = 0$ then
    for $k = 1, \ldots, r$ do
        Draw $n_{k,0} - n^0_{k,0}$ samples of $\mathbf{Q}_{M_0}$ based on (†);
else
    for $k = 1, \ldots, r$ do
        Draw $n_{k,L} - n^0_{k,L}$ samples of $\mathbf{Q}_{M_L}$ and $\mathbf{Q}_{M_{L-1}}$ based on (†);
for $k = 1, \ldots, r$ do
    for $j = n^0_{k,L} + 1, \ldots, n_{k,L}$ do
        for $\mathbf{v} \in \mathcal{S}$ do
            Compute $g_{\mathbf{v}}(\mathbf{Y}^{(j,k)}_L)$;
    Compute improved estimate $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}^{(k)}_L)]$ of $\mathbb{V}[g_{\mathbf{v}}(\mathbf{Y}^{(k)}_L)]$;
for $k = 1, \ldots, r$ do
    Compute the average computational cost per sample in stratum $k$ at level $L$, $\bar{w}_{k,L}$;
end
for $\mathbf{v} \in \mathcal{S}$ do
    Compute $\hat{\mathcal{I}}^{\mathrm{sMC}}_{\mathbf{v}}(\mathbf{Y}_L)$ and $\hat{\mathcal{I}}^{\mathrm{sMCsm}}_{\mathbf{v}}(\mathbf{Y}_L)$;
end
Set $n^\star_{k,L} = n_{k,L}$ for all $k = 1, \ldots, r$;
for $l = 0, \ldots, L-1$ do
    for $\mathbf{v} \in \mathcal{S}$ do
        for $k = 1, \ldots, r$ do
            Compute $n_{k,l,n} = \frac{2}{\alpha\epsilon^2}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}^{(k)}_l)]p^2_k/\bar{w}_{k,l}}\sum_{z=0}^{L}\sum_{k=1}^{r}\sqrt{\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}^{(k)}_z)]p^2_k\bar{w}_{k,z}}$;
        end
    end
    for $k = 1, \ldots, r$ do
        Set $n_{k,l} = \mathrm{ceil}\left(\max_{\mathbf{v}\in\mathcal{S}} n_{k,l,n}\right)$;
    end
    Set $N_l = \sum_{k=1}^{r} n_{k,l}$;
    Draw $n_{k,L} - n^\star_{k,L}$ samples of random IP/IC from stratum $k$;
    if $l = 0$ then
        for $k = 1, \ldots, r$ do
            Compute $n_{k,0} - n^\star_{k,0}$ samples of $\mathbf{Q}_{M_0}$ based on (‡);
        end
    else
        for $k = 1, \ldots, r$ do
            Compute $n_{k,l} - n^\star_{k,l}$ samples of $\mathbf{Q}_{M_l}$ and $\mathbf{Q}_{M_{l-1}}$ based on (‡);
        end
    end
    See next page (part (a))
end
See next page (part (b))
```

```
(a)
for $k = 1, \ldots, r$ do
    for $j = n^\star_{k,l} + 1, \ldots, n_{k,l}$ do
        for $\mathbf{v} \in \mathcal{S}$ do
            Compute $g_{\mathbf{v}}(\mathbf{Y}_l^{(j,k)})$;
        end
    end
    Compute improved estimate $\tilde{V}[g_{\mathbf{v}}(\mathbf{Y}_L^{(k)})]$ of $\mathbb{V}[g_{\mathbf{v}}(\mathbf{Y}_L^{(k)})]$;
end
for $k = 1, \ldots, r$ do
    Compute the average computational cost per sample in stratum $k$ at level $l$, $\bar{w}_{k,l}$;
end
for $n = 0, \ldots, S$ do
    Compute $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{sMC}}(\mathbf{Y}_l)$ and $\hat{\mathcal{I}}_{\mathbf{v}}^{\text{sMCsm}}(\mathbf{Y}_l)$;
end
Set $n^\star_{k,l} = n_{k,l}$ for all $k = 1, \ldots, r$;
 (b)
if $(L \geq 1$ and $\max_{\mathbf{v} \in \mathcal{S}} |\hat{\mathcal{I}}_{\mathbf{v}}^{\text{sMC}}(\mathbf{Y}_L)| \leq \sqrt{1-\alpha}\,\epsilon)$ or $(L = L_{\max})$ then
    Compute the cost of sMLMC, $\mathcal{C}(\hat{F}_{\mathbf{h},\boldsymbol{\delta},M}^{\text{sMLsm}})$;
    Compute the sMLMC estimator of $F(\mathbf{q})$, $\hat{F}_{\mathbf{h},\boldsymbol{\delta},M}^{\text{sMLsm}}(\mathbf{q})$;
end
```

References

[1] M.C. Hill, C. Tiedeman, Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty, John Wiley, Hoboken, NJ, 2007.

[2] D.M. Tartakovsky, Assessment and management of risk in subsurface hydrology: a review and perspective, Adv. Water Resour. 51 (2013) 247–260.

[3] O. Sen, N.K. Rai, A.S. Diggs, D.B. Hardin, H.S. Udaykumar, Multi-scale shock-to-detonation simulation of pressed energetic material: a meso-informed ignition and growth model, J. Appl. Phys. 124 (2018) 085110.

[4] D. Xiu, Numerical Methods for Stochastic Computations: A Spectral Method Approach, Princeton University Press, Princeton, NJ, 2010.

[5] S. Taverniers, D.M. Tartakovsky, Impact of parametric uncertainty on estimation of the energy deposition into an irradiated brain tumor, J. Comput. Phys. 348 (2017) 139–150.

[6] D.A. Barajas-Solano, D.M. Tartakovsky, Stochastic collocation methods for nonlinear parabolic equations with random coefficients, SIAM/ASA J. Uncertain. Quantificat. 4 (2016) 475–494.

[7] K.D. Jarman, T.F. Russell, Eulerian moment equations for 2-D stochastic immiscible flow, Multiscale Model. Simul. 1 (2003) 598–608.

[8] C.L. Winter, D.M. Tartakovsky, A. Guadagnini, Moment equations for flow in highly heterogeneous porous media, Surv. Geophys. 24 (2003) 81–106.

[9] P.C. Lichtner, D.M. Tartakovsky, Upscaled effective rate constant for heterogeneous reactions, Stoch. Environ. Res. Risk Assess. 17 (2003) 419–429.

[10] D.M. Tartakovsky, M. Dentz, P.C. Lichtner, Probability density functions for advective-reactive transport in porous media with uncertain reaction rates, Water Resour. Res. 45 (2009) W07414.

[11] G.S. Fishman, Monte Carlo: Concepts, Algorithms and Applications, Springer Series in Operations Research, Springer-Verlag, New York, 1996.

[12] S. Heinrich, Monte Carlo complexity of global solution of integral equations, J. Complex. 14 (1998) 151–175.

[13] S. Heinrich, Multilevel Monte Carlo Methods, Springer Lecture Notes in Computer Science, vol. 2179, 2001, pp. 3624–3651.

[14] M.B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. 56 (3) (2008) 607–617.

[15] F. Müller, D.W. Meyer, P. Jenny, Solver-based vs. grid-based multilevel Monte Carlo for two phase flow and transport in random heterogeneous porous media, J. Comput. Phys. 268 (2014) 39–50.

[16] B. Peherstorfer, K. Willcox, M. Gunzburger, Optimal model management for multifidelity Monte Carlo estimation, SIAM J. Sci. Comput. 38 (2016) A3163–A3194.

[17] D. O'Malley, S. Karra, J.D. Hyman, H.S. Viswanathan, G. Srinivasan, Efficient Monte Carlo with graph-based subsurface flow and transport models, Water Resour. Res. 54 (5) (2018) 3758–3766.

[18] K.A. Cliffe, M.B. Giles, R. Scheichl, A.L. Teckentrup, Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients, Comput. Vis. Sci. 14 (1) (2011) 3–15.

[19] S. Mishra, C. Schwab, Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data, Math. Comput. 81 (280) (2012) 1979–2018.

[20] F. Müller, P. Jenny, D.W. Meyer, Multilevel Monte Carlo for two phase flow and Buckley-Leverett transport in random heterogeneous porous media, J. Comput. Phys. 250 (2013) 685–702.

[21] D. Lu, D. Ricciuto, K. Evans, An efficient Bayesian data-worth analysis using a multilevel Monte Carlo method, Adv. Water Resour. 113 (2018) 223–235.

[22] A. Tversky, D. Kahneman, Advances in prospect theory: cumulative representation of uncertainty, J. Risk Uncertain. 5 (1992) 297–323.

[23] P. Goovaerts, Geostatistical modeling of uncertainty in soil science, Geoderma 103 (2001) 3–26.

[24] F. Boso, D.M. Tartakovsky, The method of distributions for dispersive transport in porous media with uncertain hydraulic properties, Water Resour. Res. 52 (2016) 4700–4712.

[25] M.B. Giles, T. Nagapetyan, K. Ritter, Multilevel Monte Carlo approximation of distribution functions and densities, SIAM/ASA J. Uncertain. Quantificat. 3 (2015) 267–295.

[26] C. Bierig, A. Chernov, Approximation of probability density functions by the multilevel Monte Carlo maximum entropy method, J. Comput. Phys. 314 (2016) 661–681.

[27] D. Wilson, R.E. Baker, Multi-level methods and approximating distribution functions, AIP Adv. 6 (2016) 075020.

[28] D. Elfverson, F. Hellman, A. Målqvist, A multilevel Monte Carlo method for computing failure probabilities, SIAM/ASA J. Uncertain. Quantificat. 4 (2016) 312–330.

[29] D. Lu, G. Zhang, C. Webster, C. Barbier, An improved multilevel Monte Carlo method for estimating probability distribution functions in stochastic oil reservoir simulations, Water Resour. Res. 52 (12) (2016) 9642–9660.

[30] S. Krumscheid, F. Nobile, Multilevel Monte Carlo approximation of functions, SIAM/ASA J. Uncertain. Quantificat. 6 (2018) 1256–1293.

[31] F.Y. Kuo, R. Scheichl, C. Schwab, I.H. Sloan, E. Ullmann, Multilevel quasi-Monte Carlo methods for lognormal diffusion problems, Math. Comput. 86 (2017) 2827–2860.

[32] A. Kebaier, J. Lelong, Coupling importance sampling and multilevel Monte Carlo using sample average approximation, Methodol. Comput. Appl. Probab. 20 (2018) 611–641.

[33] E. Ullmann, I. Papaioannou, Multilevel estimation of rare events, SIAM/ASA J. Uncertain. Quantificat. 3 (2015) 922–953.

[34] S.-K. Au, J.L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probab. Eng. Mech. 16 (4) (2001) 263–277.

[35] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Stat. 27 (3) (1956) 832–837.

[36] X. Zhang, D.M. Tartakovsky, Effective ion diffusion in charged nonaporous materials, J. Electrochem. Soc. 164 (4) (2017) E53–E61.