A continuation multilevel Monte Carlo algorithm

Nathan Collier, Abdul-Lateef Haji-Ali, Fabio Nobile,
Erik von Schwerin, Raúl Tempone

A Continuation Multilevel Monte Carlo algorithm

Nathan Collier · **Abdul–Lateef Haji–Ali** ·
Fabio Nobile · **Erik von Schwerin** ·
Raúl Tempone

Abstract We propose a novel Continuation Multi Level Monte Carlo (CMLMC) algorithm for weak approximation of stochastic models that are described in terms of differential equations either driven by random measures or with random coefficients. The CMLMC algorithm solves the given approximation problem for a sequence of decreasing tolerances, ending with the desired one. CMLMC assumes discretization hierarchies that are defined a priori for each level and are geometrically refined across levels. The actual choice of computational work across levels is based on parametric models for the average cost per sample and the corresponding weak and strong errors. These parameters are calibrated using Bayesian estimation, taking particular notice of the deepest levels of the discretization hierarchy, where only few realizations are available to produce the estimates. The resulting CMLMC estimator exhibits a nontrivial splitting between bias and statistical contributions. We also show the asymptotic normality of the statistical error in the MLMC estimator and justify in this way our error estimate that allows prescribing both required accuracy and confidence in the final result. Numerical examples substantiate the above results and illustrate the corresponding computational savings.

N. Collier (E-mail: nathaniel.collier@gmail.com)
Oak Ridge National Lab, Climate Change Science Institute (CCSI), Environmental Sciences Division

A. Haji–Ali (E-mail: abdullateef.hajiali@kaust.edu.sa) · R. Tempone (E-mail: raul.tempone@kaust.edu.sa)
Applied Mathematics and Computational Sciences, KAUST, Thuwal, Saudi Arabia.

F. Nobile (E-mail: fabio.nobile@epfl.ch)
MOX-Dipartimento di Matematica "Francesco Brioschi", Politecnico di Milano, Italy.

F. Nobile · E. Schwerin
MATHICSE-CSQI, EPF de Lausanne, Switzerland.

1 Introduction

Multilevel Monte Carlo Sampling was first introduced for applications in the context of parametric integration by Heinrich [19, 20]. Later, to consider weak approximation of stochastic differential equations (SDEs) in mathematical finance, Kebaier [26] introduced a two-level Monte Carlo technique in which a coarse grid numerical approximation of an SDE was used as a control variate to a fine grid numerical approximation, thus reducing the number of samples needed on the fine grid and decreasing the total computational burden. This idea was extended to a multilevel Monte Carlo (MLMC) method by Giles in [12], who introduced a full hierarchy of discretizations with geometrically decreasing grid sizes. By optimally choosing the number of samples on each level this MLMC method decreases the computational burden, not only by a constant factor as standard control variate techniques do, but even reducing the rate in the computational complexity from $\mathcal{O}\left(\text{TOL}^{-3}\right)$ of the standard Euler-Maruyama Monte Carlo method to $\mathcal{O}\left(\log\left(\text{TOL}\right)^2\text{TOL}^{-2}\right)$, assuming that the work to generate a single realization is $\mathcal{O}\left(\text{TOL}^{-1}\right)$. For one-dimensional SDEs, the computational complexity of MLMC was further reduced to $\mathcal{O}\left(\text{TOL}^{-2}\right)$ by using the Milstein Scheme [17]. Moreover, the same computational complexity can be achieved by using antithetic control variates with MLMC in multi-dimensional SDEs with smooth and piecewise smooth payoffs [16].

This standard MLMC method has since then been extended and applied in a wide variety of contexts, including jump diffusions [34] and Partial Differential Equations (PDEs) with random coefficients [7–9, 13, 33]. The goal in these applications is to compute a scalar quantity of interest that is a functional of the solution of a stochastic PDE (SPDE). It is proved in [33, Theorem 2.5] that there is an optimal convergence rate that is similar to the previously mentioned complexity rates, but that depends on the relation between the rate of strong convergence of the discretization method of the SPDE and the work complexity associated with generating a single sample of the quantity of interest. In fact, in certain cases, the computational complexity can be of the optimal rate, namely $\mathcal{O}\left(\text{TOL}^{-2}\right)$.

To achieve the optimal MLMC complexity rate and to get an estimate of the statistical error, sufficiently accurate estimates of the variance on each level must be obtained. Moreover, finding the optimal number of levels requires a sufficiently accurate estimate of the bias. As such, an algorithm is needed to find these estimates without incurring a significant overhead to the estimation of the wanted quantity of interest. In [12], Giles proposed an algorithm, henceforth referred to as *Standard* MLMC or SMLMC, that works by iteratively increasing the number of levels and using sample variance estimates across levels. Moreover, SMLMC uses an arbitrary fixed accuracy splitting between the bias and the statistical error contributions. Other works [32, 14, 15, 9] listed similar versions of this algorithm. We outline this algorithm in Section 3.

In Section 4, we propose a novel continuation type of MLMC algorithm that uses models for strong and weak convergence and for average computational work per sample. We refer to this algorithm as Continuation MLMC or CMLMC. The CMLMC algorithm solves the given problem for a sequence of decreasing tolerances, which play the role of the continuation parameter, ending with the prescribed toler-

ance. Solving this sequence of problems allows CMLMC to find increasingly accurate estimates of the bias and variances on each level, in addition to the quantity of interest, which is the goal of the computation. In each case, an optimizied MLMC hierarchy is generated given the current estimate of parameters. Moreover, we use a Bayesian inference approach to robustly estimate the various problem parameters. The CMLMC algorithm is able to relax the statistical error bound given the bias estimate, to achieve the optimal splitting between the two. These techniques improve the computational complexity of the CMLMC algorihtm and increase the overall stability of the algorithm.

The outline of this work is as follows: We start in Section 2 by recalling the MLMC method and the assumed models on work, and on weak and strong convergence. After introducing the algorithms in Sections 3 and 4, Section 5 presents numerical examples, which include three-dimensional PDEs with random inputs and Itô SDEs. Finally, we finish by offering conclusions and suggesting directions for future work in Section 6.

2 Multilevel Monte Carlo

2.1 Problem Setting

Let $g(u)$ denote a real valued functional of the solution, u, of an underlying stochastic model. We assume that g is either a bounded linear functional or Lipschitz with respect to u. Our goal is to approximate the expected value, $\mathrm{E}[g(u)]$, to a given accuracy TOL and a given confidence level. We assume that individual outcomes of the underlying solution u and the evaluation of the functional $g(u)$ are approximated by a discretization-based numerical scheme characterized by a mesh size, h. The value of h will govern the weak and strong errors in the approximation of $g(u)$ as we will see below. To motivate this setting, we now give two examples and identify the numerical discretizations, the discretization parameter, h, and the corresponding rates of approximation. The first example is common in engineering applications like heat conduction and groundwater flow. Here, the value of the diffusion coefficient and the forcing are represented by random fields, yielding a random solution and a functional to be approximated in the mean. The second example is a simple one-dimensional geometric Brownian motion with European call option.

Example 2.1 Let $(\Omega, \mathscr{F}, P)$ be a complete probability space and $\mathscr{D}$ be a bounded convex polygonal domain in $\mathbb{R}^d$. Find $u : \mathscr{D} \times \Omega \to \mathbb{R}$ that solves almost surely (a.s.) the following equation:

$$
\begin{aligned}
-\nabla \cdot (a(\mathbf{x}; \omega)\nabla u(\mathbf{x}; \omega)) = f(\mathbf{x}; \omega) \qquad & \text{for } \mathbf{x} \in \mathscr{D}, \\
u(\mathbf{x}; \omega) = 0 \qquad & \text{for } \mathbf{x} \in \partial\mathscr{D}.
\end{aligned}
$$

Here, we make the standard assumptions on the coefficients: there exist two positive random variables, $0 < a_{min} \le a_{max} < \infty$ such that $a_{min}(\omega) \le a(x, \omega) \le a_{max}(\omega)$ a.s. and almost everywhere (a.e.) on $\mathscr{D}$. With respect to the right-hand side, $f : D \times \Omega \to \mathbb{R}$, we here assume that there exists a random variable, $C_f(\omega) < \infty$, such that

$\|f(\cdot,\omega)\|_{L^2(\mathscr{D})} < C_f(\omega)$ a.s. Denote the space $H_0^1(D) = \{v \in H^1(D) : \|v - \varphi_n\|_{H^1(D)} \to 0$ as $n \to \infty$, for some sequence $(\varphi_n) \subset C_0^\infty(D)\}$ endowed with the norm $\|v\|_{H_0^1(D)} = \|\nabla v\|_{L^2(D)}$. Under the previous assumptions, there exists a unique solution, $u(\cdot,\omega) \in H_0^1(D) \subset H^1(D)$, such that

$$\|u(\omega)\|_{H_0^1(D)} \leq \frac{C_P \|f\|_{L^2(D)}}{a_{min}(\omega)} \text{ a.s. },$$

where C_P is the Poincaré constant of the domain $\mathscr{D}$, i.e., $\|v\|_{L^2(\mathscr{D})} \leq C_P \|v\|_{H_0^1(\mathscr{D})}$, for all $v \in H_0^1(\mathscr{D})$.

We also assume that there exists a random variable, $0 \leq C_a(\omega) < \infty$, such that $\|\nabla a(\cdot,\omega)\|_{L^\infty(\mathscr{D})} \leq C_a(\omega)$ a.s. Thus, there exists a random variable, $0 < C_u(\omega)$, such that $\|u(\omega)\|_{H^2(\mathscr{D})} \leq C_u(\omega)$ a.s.

A standard approach to approximate the solution of this problem is to use Finite Elements on regular triangulations. In such a setting, the parameter $h > 0$ refers to either the maximum element diameter or another characteristic length and the corresponding approximate solution is denoted by $u_h(\omega)$. If g is an $L^2(\mathscr{D})$ continuous functional and with the assumptions in this example, then, for piecewise linear or piecewise bilinear continuous finite element approximations, the following approximation rates hold: there exist a random variable, $0 \leq C_g(\omega) < \infty$ such that $|g(u) - g(u_h)| \leq C_g h^2$ a.s. By assuming extra integrability on the coefficients a and f, we can even obtain the estimates $|\mathrm{E}[g(u) - g(u_h)]| = Q_W h^2 + o(h^2)$ and $\mathrm{E}[(g(u) - g(u_h))^2] = Q_S h^4 + o(h^4)$ for some constants $0 < Q_W, Q_S < \infty$.

Example 2.2 Here we study the weak approximation of Itô stochastic differential equations (SDEs),

$$du(t) = a(t,u(t))dt + b(t,u(t))dW(t), \qquad 0 < t < T,$$

where $u(t;\omega)$ is a stochastic process in $\mathbb{R}^d$, with randomness generated by a k-dimensional Wiener process with independent components, $W(t;\omega)$, cf. [25,29], and $a(t,u) \in \mathbb{R}^d$ and $b(t,u) \in \mathbb{R}^{d \times k}$ are the drift and diffusion fluxes, respectively. For any given sufficiently well behaved function, $g : \mathbb{R}^d \to \mathbb{R}$, our goal is to approximate the expected value, $\mathrm{E}[g(u(T))]$. A typical application is to compute option prices in mathematical finance, cf. [24,18], and other related models based on stochastic dynamics.

When one uses a standard Euler Maruyama (Forward Euler) method based on uniform time steps of size h to approximate (2.2), then the following rates of approximation hold: $|\mathrm{E}[g(u(T)) - g(u_h(T))]| = Q_W h + o(h)$ and $\mathrm{E}[(g(u(T)) - g(u_h(T)))^2] = Q_S h + o(h)$, for some constants, $0 < Q_W, Q_S < \infty$. For suitable assumptions on the functions a, b and g, we refer to [27,22].

To avoid cluttering the notation, we omit the reference to the underlying solution from now on, simply denoting the quantity of interest by g.

Following the standard MLMC approach, we assume, for any given non-negative integer $L \in \mathbb{N}$, that we have a hierarchy of $L+1$ meshes defined by a decreasing sequence of mesh sizes $\{h_\ell\}_{\ell=0}^L$ where $h_\ell = h_0 \beta^{-\ell}$ for some $h_0 > 0$ and a constant

integer $\beta > 1$. We denote the resulting approximation of g using mesh size h_ℓ by g_ℓ, or by $g_\ell(\omega)$ when we want to stress the dependence on an outcome on the underlying random model. Using the following notation:

$$G_\ell = \begin{cases} g_0 & \text{if } \ell = 0, \\ g_\ell - g_{\ell-1} & \text{if } \ell > 0, \end{cases}$$

the expected value of the finest approximation, g_L, can be expressed as

$$\mathrm{E}[g_L] = \sum_{\ell=0}^{L} \mathrm{E}[G_\ell],$$

where the MLMC estimator is obtained by replacing the expected values in the telescoping sum by sample averages. Denoting the sample averages by $\widetilde{G}_\ell$ as

$$\widetilde{G}_\ell = \begin{cases} M_0^{-1} \sum_{m=1}^{M_0} g_0(\omega_{0,m}) & \text{if } \ell = 0, \\ M_\ell^{-1} \sum_{m=1}^{M_\ell} \left(g_\ell(\omega_{\ell,m}) - g_{\ell-1}(\omega_{\ell,m}) \right) & \text{if } \ell > 0, \end{cases}$$

the MLMC estimator can be written as

$$\mathscr{A} = \sum_{\ell=0}^{L} \widetilde{G}_\ell. \tag{2.2}$$

Each sample average, $\widetilde{G}_\ell$, is computed using M_ℓ independent identically distributed (i.i.d.) outcomes, $\{\omega_{\ell,m}\}_{m=1}^{M_\ell}$, of the underlying, mesh-independent, stochastic model; the outcomes are also assumed to be independent between the different sample averages. The number of samples on any level is a positive integer, $M_\ell \in \mathbb{Z}_+$.

We use the following model for the expected value of the cost associated with generating one sample of G_ℓ, including generating all the underlying random variables:

$$W_\ell \propto \begin{cases} h_0^{-\gamma} & \text{if } \ell = 0, \\ h_\ell^{-\gamma} + h_{\ell-1}^{-\gamma} & \text{if } \ell > 0, \end{cases}$$

for a given γ. Note the cost of generating a sample of G_ℓ might differ for different realizations, for example due to different number of iterations in an iterative method or due to adaptivity of the used numerical method. The parameter γ depends on the number of dimensions of the underlying problem and the used numerical method. For example, $\gamma = 1$ for the one-dimensional SDE in Example 2.2. For the SPDE in Example 2.1, if the number of dimensions is $d = 3$ then $\gamma = 3\tilde{\gamma}$, where $\tilde{\gamma}$ depends on the solver used to solve the resulting linear system. In that example, iterative methods may have a smaller value of $\tilde{\gamma}$ than direct methods. The theoretical best-case scenario for iterative methods would be $\tilde{\gamma} = 1$ for multigrid methods. On the other hand, we would have $\tilde{\gamma} = 3$ if one used a direct method using a naive Gaussian elimination on dense matrices. The total work of the estimator (2.2) is

$$W = \sum_{\ell=0}^{L} M_\ell W_\ell.$$

We want our estimator to satisfy a tolerance with prescribed failure probability $0 < \alpha < 1$, i.e.,

$$P[|E[g] - \mathscr{A}| > \text{TOL}] \leq \alpha, \tag{2.3}$$

while minimizing the work, W. Here, we split the total error into bias and statistical error,

$$|E[g] - \mathscr{A}| \leq \underbrace{|E[g - \mathscr{A}]|}_{\text{Bias}} + \underbrace{|E[\mathscr{A}] - \mathscr{A}|}_{\text{Statistical error}},$$

and use a splitting parameter, $\theta \in (0, 1)$, such that

$$\text{TOL} = \underbrace{\theta\text{TOL}}_{\text{Statistical error tolerance}} + \underbrace{(1 - \theta)\text{TOL}}_{\text{Bias tolerance}}.$$

The MLMC algorithm should bound the bias, $B = |E[g - \mathscr{A}]|$, and the statistical error as follows:

$$B = |E[g - \mathscr{A}]| \leq (1 - \theta)\text{TOL}, \tag{2.4a}$$
$$|E[\mathscr{A}] - \mathscr{A}| \leq \theta\text{TOL}, \tag{2.4b}$$

where the latter bound should hold with probability $1 - \alpha$. Note that θ does not have to be a constant, indeed it can depend on TOL as we shall see in Section 4. In the literature [12], some authors have controlled the mean square error (MSE),

$$\text{MSE} = |E[g - \mathscr{A}]|^2 + E\left[|E[\mathscr{A}] - \mathscr{A}|^2\right],$$

rather than working with (2.3). We prefer to work with (2.3) since it allows us to prescribe both the accuracy TOL and the confidence level, $1 - \alpha$, in our results.

The bound (2.4b) leads us to require

$$\text{Var}[\mathscr{A}] \leq \left(\frac{\theta\text{TOL}}{C_\alpha}\right)^2, \tag{2.5}$$

for some given confidence parameter, C_α, such that $\Phi(C_\alpha) = 1 - \frac{\alpha}{2}$; here, Φ is the cumulative distribution function of a standard normal random variable. The bound (2.5) is motivated by the Lindeberg Central Limit Theorem in the limit $\text{TOL} \to 0$, cf. Lemma A.1 in the Appendix.

By construction of the MLMC estimator, $E[\mathscr{A}] = E[g_L]$, and denoting $V_\ell = \text{Var}[G_\ell]$, then by independence, we have $\text{Var}[\mathscr{A}] = \sum_{\ell=0}^{L} V_\ell M_\ell^{-1}$, and the total error estimate can be written as

$$\text{Total error estimate} = B + C_\alpha\sqrt{\text{Var}[\mathscr{A}]}. \tag{2.6}$$

Given L and $0 < \theta < 1$ and minimizing W subject to the statistical constraint (2.5) for $\{M_\ell\}_{\ell=0}^{L} \in \mathbb{R}^{L+1}$ gives the following optimal number of samples per level ℓ:

$$M_\ell = \left(\frac{C_\alpha}{\theta\text{TOL}}\right)^2 \sqrt{\frac{V_\ell}{W_\ell}} \left(\sum_{\ell=0}^{L} \sqrt{V_\ell W_\ell}\right). \tag{2.7}$$

When substituting the optimal number of samples in all levels the optimal work can be written in terms of L as follows

$$W(\text{TOL}, L) = \left(\frac{C_\alpha}{\theta \text{TOL}}\right)^2 \left(\sum_{\ell=0}^{L} \sqrt{V_\ell W_\ell}\right)^2. \tag{2.8}$$

Of course, the number of samples on each level is a positive integer. To get an approximate value of the optimal integer number of samples, we take the ceiling of the real-valued optimal values in (2.7).

In this work, we assume the following models on the weak error and variance:

$$\text{E}[g - g_\ell] \approx Q_W h_\ell^{q_1}, \tag{2.9a}$$

$$\text{Var}[g - g_\ell] \approx Q_S h_\ell^{q_2}, \tag{2.9b}$$

for some constants $Q_W \neq 0, Q_S > 0, q_1 > 0$ and $0 < q_2 \leq 2q_1$. For example, the SPDE in Example 2.1 has $q_2 = 2q_1$ and in Section 5, the SPDE is solved using a finite element method with standard trilinear basis and it has $q_1 = 2$. For the SDE in Example 2.2 with Euler discretization, $q_1 = q_2 = 1$. Collectively, we refer to the parameters q_1, q_2, Q_S, Q_W and $\{V_\ell\}_{\ell=0}^{L}$ as problem parameters. Based on these models, we can write for $\ell > 0$

$$\text{E}[G_\ell] \approx Q_W h_\ell^{q_1} \left(1 - \beta^{-q_1}\right), \tag{2.10a}$$

$$\text{Var}[G_\ell] = V_\ell \approx Q_S h_\ell^{q_2} \left(1 - \beta^{\frac{-q_2}{2}}\right)^2. \tag{2.10b}$$

Specifically, as a consequence of (2.9a), the bias model is

$$B \approx |Q_W| h_L^{q_1}. \tag{2.11}$$

Finally, we note that the algorithms presented in this work are iterative. We therefore denote by $\overline{M}_\ell, \overline{G}_\ell$ and $\overline{V}_\ell$ the total number of samples of G_ℓ generated in all iterations and their sample average and sample variance, respectively. Explicitly, we write

$$\overline{G}_\ell = \frac{1}{\overline{M}_\ell} \sum_{m=1}^{\overline{M}_\ell} G_{\ell,m}, \tag{2.12a}$$

$$\overline{V}_\ell = \frac{1}{\overline{M}_\ell} \sum_{m=1}^{\overline{M}_\ell} \left(G_{\ell,m} - \overline{G}_\ell\right)^2. \tag{2.12b}$$

3 Standard MLMC

3.1 Overview

While minor variations exist among MLMC algorithms listed in [12, 15, 16], we believe that there is sufficient commonality in them for us to outline here the overarching idea and refer to this collection of methods as the Standard MLMC algorithm or simply SMLMC.

SMLMC solves the problem by iteratively increasing the number of levels of the MLMC hierarchy. In order to find the optimal number of samples of each level ℓ, an estimate of the variance V_ℓ is needed. If there were previously generated samples in previous iterations for a level ℓ, the sample variance $\overline{V}_\ell$ is used. Otherwise, an initial fixed number of samples, $\widetilde{M}$, is generated. Moreover, in most works, the splitting between bias and statistical error, θ, is chosen to be 0.5.

After running the hierarchy, an estimate of the total error is computed. To this end, the work [12] approximates the absolute value of the constant, Q_W, using a similar expression to the following:

$$|Q_W| \approx \max\left(\frac{|\overline{G}_L|}{h_L^{q_1}(1-\beta^{q_1})}, \frac{|\overline{G}_{L-1}|}{h_{L-1}^{q_1}(1-\beta^{q_1})}\right) := \widetilde{Q}_W.$$

In other words, the absolute value of the constant Q_W is estimated using the samples generated on the last two levels. Thus, this estimate is only defined for $L \geq 2$. Next, the variance of the estimator, $\mathrm{Var}[\mathscr{A}]$, is approximated by

$$\mathrm{Var}[\mathscr{A}] \approx \sum_{\ell=0}^{L} \frac{\overline{V}_\ell}{\overline{M}_\ell} := \widetilde{V}.$$

Finally, a total error estimate can be computed as outlined by (2.6)

$$\text{Total error estimate} = \widetilde{Q}_W h_L^{q_1} + C_\alpha \sqrt{\widetilde{V}}. \tag{3.1}$$

The complete algorithm is outlined in Algorithm 1.

Algorithm 1

1: **function** STANDARDMLMC(TOL, $\widetilde{M}$, θ)
2:     Start with $L = 0$.
3:     **loop**
4:         Generate hierarchy $\{h_\ell\}_{\ell=0}^{L}$.
5:         Generate $\widetilde{M}$ samples for level L and estimate $\overline{V}_L$.
6:         Using sample variance estimates, $\{\overline{V}_\ell\}_{\ell=0}^{L}$ from all iterations, and the constant θ, compute optimal number of samples, $\{M_\ell\}_{\ell=0}^{L}$, according to (2.7).
7:         Compute with the hierarchy using the optimal number of samples.
8:         If $L \geq 2$ and the total estimate error (3.1) is less than TOL, then END
9:         Otherwise, set $L = L + 1$.
10:     **end loop**
11: **end function**

Usually all samples from previous iterations are used in the algorithm to run the hierarchy in step 7 to calculate the required quantity of interest. However, the analysis of the bias and the statistical error of the resulting estimator is difficult and has not been done before, to the best of our knowledge.

3.2 Accuracy of the parameter estimates

In the standard algorithm, Q_W and the variances $\{V_\ell\}_{\ell=0}^{L}$ are needed and estimated. In this section, we look at the accuracy of the estimators for these problem parameters.

We examine the accuracy of the sample variance by computing its squared relative error for $\ell > 1$:

$$
\begin{aligned}
\frac{\mathrm{Var}\left[\,\overline{V}_\ell\right]}{V_\ell^2} &= \frac{\left(\overline{M}_\ell-1\right)^2}{\overline{M}_\ell^3 V_\ell^2}\left(\mathrm{E}\left[(G_\ell-\mathrm{E}[G_\ell])^4\right]-\frac{V_\ell^2(\overline{M}_\ell-3)}{\overline{M}_\ell-1}\right) \\
&= C\frac{\left(\overline{M}_\ell-1\right)^2}{\overline{M}_\ell^3}\left(\mathrm{E}\left[(G_\ell-\mathrm{E}[G_\ell])^4\right]V_\ell^{-2}-\frac{\overline{M}_\ell-3}{\overline{M}_\ell-1}\right) \\
&\approx C\frac{\left(\overline{M}_\ell-1\right)^2}{\overline{M}_\ell^3}\left(\mathrm{E}\left[(G_\ell-\mathrm{E}[G_\ell])^4\right]Q_S^{-2}h_\ell^{-2q_2}-\frac{\overline{M}_\ell-3}{\overline{M}_\ell-1}\right).
\end{aligned}
$$

Unless $\mathrm{E}\left[(G_\ell-\mathrm{E}[G_\ell])^4\right] \leq Ch_\ell^{2q_2}$, for some constant $C > 0$, or M_ℓ increases sufficiently fast, the relative error in the estimator $\overline{V}_\ell$ can become unbounded as $\ell \to \infty$. Similarly, the relative error of the sample variance at level $\ell = 0$ can be shown to be bounded for instance by assuming that the second and fourth central moments of G_0 are bounded.

Next, for simplicity, we look at the squared relative error estimate of Q_W by assuming that it is estimated using samples on a single level, L, only.

$$
\begin{aligned}
\frac{\mathrm{Var}\left[\left|\frac{\overline{G}_L}{h_L^{q_1}(1-\beta^{q_1})}\right|\right]}{Q_W^2} &= \frac{V_L}{Q_W^2 M_L h_L^{2q_1}(1-\beta^{q_1})^2} \\
&= \frac{Q_S}{Q_W^2}\cdot\frac{h_L^{q_2}\left(1-\beta^{\frac{q_2}{2}}\right)^2}{M_L h_L^{2q_1}(1-\beta^{q_1})^2} \\
&= \frac{Q_S(1-\beta^{\frac{q_2}{2}})^2}{Q_W^2(1-\beta^{q_1})^2}\left(\frac{h_L^{q_2-2q_1}}{M_L}\right).
\end{aligned}
$$

Observe now that if $q_2 < 2q_1$, then, for the previous relative error estimate to be $o(1)$, we must have $M_L \propto h_L^{q_2-2q_1} \to \infty$ as $L \to \infty$. The analysis shows that in some cases, M_L will have to grow to provide an accurate estimate to Q_W, regardless of the optimal choice of the number of samples outlined in (2.7).

4 Continuation MLMC (CMLMC)

In this section we discuss the main contribution of this work, a continuation MLMC (CMLMC) algorithm that approximates the value $\mathrm{E}[g(u)]$. We begin in the next subsection by giving an overview of the general idea of algorithm. Subsequent subsections discuss how to estimate all the required problem parameters that are necessary

for running the algorithm. CMLMC is listed in Algorithm 2. We also list the parameters that control the algorithm in Table 4.1.

4.1 Overview

The main idea is to solve for $E[g(u)]$ with a sequence of decreasing tolerances, ending with the desired accuracy requirement, TOL. By doing this, CMLMC is able to improve estimates of several problem dependent parameters while solving relatively inexpensive problems corresponding to large tolerances. These parameters estimates are crucial to optimally distribute computational effort when solving for the last tolerance, which is the desired one.

Assuming that we want to approximate $E[g(u)]$ with tolerance TOL, we make the following choice for the sequence of decreasing tolerances TOL_i for $i = 0, 1, \cdots$

$$\text{TOL}_i = \begin{cases} r_1^{i_E - i} r_2^{-1} \text{TOL} & i < i_E, \\ r_2^{i_E - i} r_2^{-1} \text{TOL} & i \geq i_E, \end{cases}$$

where $r_1 \geq r_2 > 1$. By imposing $\text{TOL}_0 = \text{TOL}_{\max}$ for some maximum tolerance, we have

$$i_E = \left\lfloor \frac{-\log(\text{TOL}) + \log(r_2) + \log(\text{TOL}_{\max})}{\log(r_1)} \right\rfloor,$$

Iterations for which $i \leq i_E$ are meant to obtain increasingly more accurate estimates of the problem parameters. The iteration i_E solves the problem for the tolerance $r_2^{-1}\text{TOL}$. Notice that the problem is solved for a slightly smaller tolerance than the required tolerance TOL. This tolerance reduction is to prevent extra unnecessary iterations due to slight variations in estimates of the problem parameters. This technique improves the overall average running time of the algorithm. Similarly, iterations $i > i_E$ have tolerances that are even smaller to account for cases in which estimates of the problem parameters are unstable. The parameters r_1 and r_2 are chosen such that the total work of the algorithm does not exceed by much the work of the hierarchy that solves the problem with the required tolerance, TOL. For example, if the work of the MLMC estimator is $\mathscr{O}(\text{TOL}^{-2})$, we choose $r_1 = 2$ to ensure that the work of iteration i is roughly four times the work of iteration $i - 1$ for iterations for which $\text{TOL}_i \geq \text{TOL}$. The choice of $r_2 = 1.1$, on the other hand, ensures that for iterations for which $\text{TOL}_i < \text{TOL}$, the work of iterations of i is roughly 1.2 times the work of iteration $i - 1$.

Consider now the i-th iteration of CMLMC and assume that estimates for $\mathbf{Q} := \{q_1, q_2, Q_W, Q_S\}$ and $\{V_\ell\}_{\ell=0}^L$ are available from previous iterations; we will discuss how to get these estimate in Section 4.2. The i-th iteration begins by selecting the optimal number of levels $L[i]$ that solves the problem for the given tolerance, TOL_i, as follows

$$L[i] = \text{argmin}_{L_{\min}[i] \leq L \leq L_{\max}[i]} W(\text{TOL}_i, L), \tag{4.1}$$

where W is defined by (2.8) and depends on all the parameters $\mathbf{Q}$ and $\{V_\ell\}_{\ell=0}^L$ and $\theta = \theta(L)$ given by

$$\theta = 1 - \frac{|Q_W| h_L^{q_1}}{\text{TOL}_i}, \tag{4.2}$$

which comes from enforcing that the bias model (2.11) equals $(1-\theta)\text{TOL}_i$. Moreover, $L_{\min}$ should satisfy $Q_W h_{L_{\min}}^{q_1} = \text{TOL}_i$ or, since we have $h_\ell = h_0 \beta^{-\ell}$,

$$L_{\min}[i] = \max\left(L[i-1], \frac{q_1 \log(h_0) - \log\left(\frac{\text{TOL}_i}{|Q_W|}\right)}{q_1 \log \beta}\right),$$

where $L[i-1]$ is the number of levels from the previous iteration. This ensures that L does not decrease from one iteration to the next, which agrees with our intuition that L increases with $\log\left(\text{TOL}_i^{-1}\right)$. On the other hand, $L_{\max}$ is given by other considerations. For instance, it could be related to the minimum mesh size imposed by memory or computational restrictions. More practically, to ensure robustness, $L_{\max}$ can be chosen to be $L_{\min} + L_{\text{inc}}$, for a given fixed integer L_{inc}, so that L has limited increments from one iteration to the next. Since only few values of L are considered in the optimization (4.1), it is easy to find the optimal L by exhaustive search. The choice (4.2) implies that the statistical constraint (2.5) is relaxed (or tightened) depending on the estimated bias of each hierarchy. The iteration then continues by building the hierarchy $\{h_\ell\}_{\ell=0}^{L}$ and computing with the optimal number of samples $\{M_\ell\}_{\ell=0}^{L}$ according to (2.7). Finally the iteration ends by improving the estimates of the problem parameters $\mathbf{Q}$ and $\{V_\ell\}_{\ell=0}^{L}$ as well as the quantity of interest based on the newly available samples as described in Section 4.2.

To start CMLMC we compute with an initial, relatively inexpensive, hierarchy. The purpose of using this initial hierarchy is to get rough estimates of the problem parameters. Such a hierarchy cannot depend on estimates of problem parameters and should have at least three levels to allow estimating $\mathbf{Q}$. The algorithm stops when the total error estimate is below the required tolerance TOL.

4.2 Parameters estimation

In this section, we discuss how to improve estimates of the parameters $\mathbf{Q}$ as well as the variances V_ℓ base on the generated samples in all iterations and all levels. For easier presentation, we will also use the following notation

$$w_\ell(q_1) = h_\ell^{q_1}(1 - \beta^{-q_1}),$$
$$s_\ell(q_2) = h_\ell^{-q_2}\left(1 - \beta^{\frac{-q_2}{2}}\right)^{-2}.$$

Thus, using the notation above, (2.10) becomes

$$\text{E}[G_\ell] \approx Q_W w_\ell(q_1), \tag{4.3a}$$
$$\text{Var}[G_\ell] = V_\ell \approx Q_S s_\ell^{-1}(q_2). \tag{4.3b}$$

4.2.1 Estimating variances V_ℓ

We first assume that we have estimates of q_1, q_2, Q_W and Q_S and discuss estimating the variances, $\{V_\ell\}_{\ell=0}^L$, and the total statistical error after computing with a given hierarchy. Estimating q_1, q_2, Q_W and Q_S is discussed in the next subsection.

Usually the variances $\{V_\ell\}_{\ell=0}^L$ are estimated by using the sample variance estimator (2.12b) to estimate the statistical error as well as the optimal number of samples $\{M_\ell\}_{\ell=0}^L$. However, sometimes there are too few samples in a given level to give a corresponding accurate variance estimate. This is specially acute on the deepest levels, and unlike the standard MLMC algorithm, we do not impose a minimum number of samples across levels to get a stable estimate of the sample variance. Recalling that we have the variance model (4.3b) at our disposal, we can use this model to estimate the variance at all levels $\ell > 0$. However, the model (4.3b) is only accurate asymptotically. We can use the generated samples on each level to locally improve the accuracy of the V_ℓ estimates. To this end, we use a Bayesian setting [31].

We assume that G_ℓ follows a normal distribution with mean μ_ℓ and precision λ_ℓ (precision is simply the inverse of the variance). To simplify the computation, we choose a normal-gamma prior on (μ_ℓ, λ_ℓ) – the conjugate prior of the normal likelihood. The resulting posterior probability density function (pdf) is also a normal-gamma distribution function. We choose the parameters $(\widehat{\mu}_\ell, \kappa_0, 0.5 + \widehat{\lambda}_i \kappa_1, \kappa_1)$ for the normal-gamma prior, such that it is maximized at $\widehat{\mu}_i$ and $\widehat{\lambda}_i$. The parameter $\widehat{\mu}_\ell$ and $\widehat{\lambda}_\ell$ serve as initial guesses for μ_ℓ and λ_ℓ, respectively. Moreover, κ_0 and κ_1 are positive constants that model our certainty in those respective guesses. We use the assumed models of the weak and strong errors (4.3) to give the initial guesses

$$\widehat{\mu}_\ell = Q_W w_\ell(q_1), \tag{4.4a}$$

$$\widehat{\lambda}_\ell = Q_S^{-1} s_\ell(q_2). \tag{4.4b}$$

As mentioned, the posterior pdf is also a normal-gamma with parameters $(\Upsilon_{1,\ell}, \Upsilon_{2,\ell}, \Upsilon_{3,\ell}, \Upsilon_{4,\ell})$ and it is maximized at $\left(\Upsilon_{1,\ell}, \frac{\Upsilon_{3,\ell} - 0.5}{\Upsilon_{4_\ell}}\right)$. Specifically

$$\Upsilon_{3,\ell} = 0.5 + \kappa_1 \widehat{\lambda}_\ell + \frac{\overline{M}_\ell}{2},$$

$$\Upsilon_{4,\ell} = \kappa_1 + \frac{1}{2}\left(\sum_{m=1}^{\overline{M}_\ell} \left(G_{\ell,m} - \overline{G}_\ell\right)^2\right) + \frac{\kappa_0 \overline{M}_\ell (\overline{G}_\ell - \widehat{\mu}_\ell)^2}{2(\kappa_0 + \overline{M}_\ell)}.$$

As such, we use the following estimate of the variance V_ℓ for $\ell > 0$

$$V_\ell \approx \frac{\Upsilon_{4,\ell}}{\Upsilon_{3,\ell} - 0.5}. \tag{4.5}$$

Estimating the variance at the coarsest mesh, V_0, can be done using the sample variance. The number of samples on the coarsest level, M_0, is usually large enough to produce a stable and accurate estimate. Using these estimates and the bias estimate (2.11), the total error can be estimated as (2.6).

4.2.2 Estimating **Q**

To incorporate prior knowledge on q_1 and q_2 including initial guesses and the relation $q_2 \leq 2q_1$, we again follow a Bayesian setting to estimate these parameters and assume that G_ℓ follows a Gaussian distribution with mean $Q_W w_\ell(q_1)$ and variance $Q_S s_\ell^{-1}(q_2)$. In what follows, ℓ_0 is a non-negative integer. With these assumptions, the corresponding likelihood is

$$\mathscr{L} = \left(\prod_{\ell=\ell_0}^{L} \left(2\pi Q_S s_\ell^{-1}(q_2) \right)^{\frac{-\overline{M}_\ell}{2}} \right) \exp \left(-\frac{1}{2Q_S} \sum_{\ell=\ell_0}^{L} s_\ell(q_2) \sum_{m=1}^{\overline{M}_\ell} \left(G_{\ell,m} - Q_W w_\ell(q_1) \right)^2 \right).$$
(4.6)

Assuming a improper prior on Q_W and Q_S and maximizing the resulting posterior pdf with respect to Q_W and Q_S gives the following weighted least-squares solution:

$$Q_W = \left(\sum_{\ell=\ell_0}^{L} \overline{M}_\ell w_\ell^2(q_1) s_\ell(q_2) \right)^{-1} \sum_{\ell=\ell_0}^{L} w_\ell(q_1) s_\ell(q_2) \overline{M}_\ell \overline{G}_\ell, \qquad (4.7a)$$

$$Q_S = \left(\sum_{\ell=\ell_0}^{L} \overline{M}_\ell \right)^{-1} \sum_{\ell=\ell_0}^{L} s_\ell(q_2) \sum_{m=1}^{\overline{M}_\ell} \left(G_{\ell,m} - Q_W w_\ell(q_1) \right)^2. \qquad (4.7b)$$

We can substitute the previous Q_S and Q_W in (4.6) to get a likelihood in terms of q_1 and q_2. Denoting $\overline{M} = \sum_{\ell=\ell_0}^{L} \overline{M}_\ell$, we write

$$\mathscr{L}(q_1, q_2) = \exp \left(-\frac{\overline{M}}{2} \right) \left(\sum_{\ell=\ell_0}^{L} \sum_{m=0}^{\overline{M}_\ell} s_\ell(q_2) G_{\ell,m}^2 - \frac{\left(\sum_{\ell=\ell_0}^{L} s_\ell(q_2) w_\ell(q_1) \overline{M}_\ell \overline{G}_\ell \right)^2}{\sum_{\ell=\ell_0}^{L} \overline{M}_\ell w_\ell(q_1)^2 s_\ell(q_2)} \right)^{-\frac{\overline{M}}{2}}.$$

We can then assume a prior on q_1 and q_2. However, remember that $q_2 \leq 2q_1$, and $q_1 > 0$. As such, we introduce the unconstrained parameters $x_0(q_1) = \log(q_1) \in \mathbb{R}$ and $x_1(q_1, q_2) = \log(2q_1 - q_2) \in \mathbb{R}$ and assume a Gaussian prior on them

$$\rho_{\text{prior}}(q_1, q_2) = \frac{1}{2\pi \sqrt{\sigma_0^2 \sigma_1^2}} \exp \left(-\frac{(x_0(q_1) - \widehat{x}_0)^2}{2\sigma_0^2} - \frac{(x_1(q_1, q_2) - \widehat{x}_1)^2}{2\sigma_1^2} \right).$$

Here, $\widehat{x}_0$ and $\widehat{x}_1$ represent our initial guesses of x_0 and x_1, respectively, which we can get from a rough analysis of the problem. Moreover, σ_1 and σ_2 model our confidence in those guesses. The more accurate our initial guesses are, the faster the algorithm converges. Finally, we numerically maximize the log of the posterior pdf with respect to $(x_0, x_1) \in \mathbb{R}^2$ using a suitable numerical optimization algorithm. For robustness, we choose $\ell_0 = 1$ to estimate q_1 and q_2. In other words we include samples from all levels $\ell > 0$ for this estimation.

Given estimates of q_1 and q_2, we can produce estimates of Q_S and Q_W by using the least-squares estimates (4.7). However, usually not all levels follow the assumed asymptotic models (2.10) and as such special care must be taken to choose ℓ_0 in

Parameter	Purpose
$\widehat{x}_0, \widehat{x}_1, \sigma_0$ and σ_1	Parameters to model the initial guess of q_1 and q_2 and the confidence in those estimates.
κ_0 and κ_1	The confidence in the weak and strong error models, respectively.
TOL_{max}	The maximum tolerance with which to start the algorithm.
r_1 and r_2	Controls the computational burden to calibrate the problem parameters compared to the one taken to solve the problem.
Initial hierarchy	The initial hierarchy to start the algorithm. Must be relatively inexpensive and has at least three levels.
L_{inc}	Maximum number of values to consider when optimizing for L.
$\mathfrak{L}$	Maximum number of levels used to compute estimates of Q_W and Q_S.
C_α	Parameter related to the confidence in the statistical constraint.

Table 4.1 Summary of parameters in CMLMC

these estimates. The parameter Q_W must be accurate on deeper levels since it is used to compute the bias (2.11). Similarly, Q_S must be accurate on deeper levels where not many samples are available and the variance estimate (4.5) is mainly determined by the initial guess (4.4b). For these reasons, when estimating Q_S and Q_W, we choose $\ell_0 = \max(1, L - \mathfrak{L})$ in (4.7) for some positive integer $\mathfrak{L}$ that denotes the maximum number of levels use to compute the estimates. Finally, Since Q_W has an improper prior, its posterior is also the Gaussian (4.6) with variance

$$V_W := \sum_{\ell=\ell_0}^{L} \frac{Q_S}{\overline{M} w_\ell^2(q_2) s_\ell(q_1)}.$$

With $1 - \alpha$ confidence, the sampling error of Q_W is $C_\alpha \sqrt{V_W}$. Motivated by the accuracy analysis of the Q_W estimate in Section 3.2, we produce a worst estimate of Q_W by adding the sampling error multiplied by the sign of Q_W estimate.

Algorithm 2

1: **function** CMLMC(Parameters summarized in Table 4.1)
2:     Compute with an initial hierarchy.
3:     Estimate problem parameters $\{V_\ell\}_{\ell=0}^{L}, Q_S, Q_W, q_1$ and q_2 according to section 4.2.
4:     Set $i = 0$.
5:     **repeat**
6:         Find L according to (4.1).
7:         Generate hierarchy $\{h_\ell\}_{\ell=0}^{L}$.
8:         Using the variance estimates (4.5) and θ from (4.2), compute the optimal number of samples according to (2.7).
9:         Compute with the resulting hierarchy using the optimal number of samples.
10:        Estimate problem parameters, $\{V_\ell\}_{\ell=0}^{L}, Q_S, Q_W, q_1$ and q_2, according to section 4.2.
11:        Estimate the total error according to (2.6).
12:        Set $i = i + 1$
13:    **until** $i > i_E$ and the total error estimate is less than TOL
14: **end function**

---

5 Numerical Tests

In this section, we first introduce the test problems. We then describe several implementation details and finish by presenting the actual numerical results.

5.1 Test Problems

We look at three test problems: the first two are PDEs with random inputs and the last one is an Itô SDE.

5.1.1 Ex.1

This problem is based on Example 2.1 in Section 2.1 with some particular choices that satisfy the assumptions therein. First, we choose $\mathscr{D} = [0,1]^3$ and assume that the forcing is

$$f(\mathbf{x};\omega) = f_0 + \widehat{f} \sum_{i=0}^{K} \sum_{j=0}^{K} \sum_{k=0}^{K} \Phi_{ijk}(\mathbf{x}) Z_{ijk},$$

where

$$\Phi_{ijk}(\mathbf{x}) = \sqrt{\lambda_i \lambda_j \lambda_k} \phi_i(x_1) \phi_j(x_2) \phi_k(x_3),$$

and

$$\phi_i(x) = \begin{cases} \cos\left(\frac{10\Lambda i}{2}\pi x\right) & i \text{ is even}, \\ \sin\left(\frac{10\Lambda (i+1)}{2}\pi x\right) & i \text{ is odd}, \end{cases}$$

$$\lambda_i = (2\pi)^{\frac{14}{12}} \Lambda^{\frac{22}{12}} \begin{cases} \frac{1}{2} & i = 0, \\ \exp\left(-2\left(\pi\frac{i}{2}\Lambda\right)^2\right) & i \text{ is even}, \\ \exp\left(-2\left(\pi\frac{i+1}{2}\Lambda\right)^2\right) & i \text{ is odd}, \end{cases}$$

for given $\Lambda > 0$, and positive integer K and $\mathbf{Z} = \{Z_{ijk}\}$ a set of $(K+1)^3$ i.i.d. standard normal random variables. Moreover, we choose the diffusion coefficient to be a function of two random variables as follows:

$$a(\mathbf{x};\omega) = a_0 + \exp\left(4Y_1\Phi_{121}(\mathbf{x}) + 40Y_2\Phi_{877}(\mathbf{x})\right).$$

Here, $\mathbf{Y} = \{Y_1, Y_2\}$ is a set of i.i.d. normal Gaussian random variables, also independent of $\mathbf{Z}$. Finally we make the following choice for the quantity of interest, g:

$$g = (2\pi\sigma)^{\frac{-3}{2}} \int_{\mathscr{D}} \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_0\|_2^2}{2\sigma^2}\right) u(\mathbf{x}) d\mathbf{x},$$

and select the parameters $a_0 = 0.01, f_0 = 50, \widehat{f} = 10, \Lambda = \frac{0.2}{\sqrt{2}}, K = 10, \sigma = 0.02622863$ and $\mathbf{x}_0 = [0.5026695, 0.26042876, 0.62141498]$. Since the diffusion coefficient, a, is independent of the forcing, f, a reference solution can be calculated to sufficient accuracy by scaling and taking expectation of the weak form with respect to $\mathbf{Z}$ to get a

formula with constant forcing for the conditional expectation with respect to Y. We then use stochastic collocation [3] with a sufficiently accurate quadrature to produce the reference value $\mathrm{E}[g]$. From this method, the reference value 1.6026 is computed with an error estimate of 10^{-4}.

5.1.2 Ex.2

The second example is a slight variation of the first. First, we choose the following diffusion coefficient instead:

$$a(\mathbf{x};\omega) = 1 + \exp\left(Y_1\phi_{121}(\mathbf{x}) + Y_2\phi_{877}(\mathbf{x})\right).$$

Moreover, in this example $\mathbf{Y}$ is a set of two i.i.d. uniform random variables in the range $[-1,1]$, again independent of $\mathbf{Z}$. We also make the following choice for the quantity of interest g

$$g = 100\,(2\pi\sigma)^{\frac{-3}{2}} \int_{\mathscr{D}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_0\|_2^2}{2\sigma^2}\right) u(\mathbf{x})d\mathbf{x},$$

and select the parameters $a_0 = 1, f_0 = 1, \widehat{f} = 1, \Lambda = 0.2, K = 10, \sigma = 0.01194691$ and
$\mathbf{x}_0 = [0.62482261, 0.45530923, 0.49862328]$. The computed reference solution $\mathrm{E}[g]$ in this case is 2.3627 with an error estimate of 10^{-4}.

5.1.3 Ex.3

The third example is a one-dimensional geometric Brownian motion based on Example 2.2. We make the following choices:

$$\begin{aligned} T &= 1, \\ a(t,u) &= 0.05u, \\ b(t,u) &= 0.2u, \\ g(u) &= 10\max(u(1) - 1, 0). \end{aligned}$$

The exact solution can be computed using a standard change of variables and Itô's formula. For the selected parameters, the solution is $\mathrm{E}[g] = 1.04505835721856$.

5.2 Implementation and Runs

All the algorithms mentioned in this work were implemented using the C programming language, with the goal that the software be as optimal as possible, while maintaining generality.

For implementing the solver for the SPDE test problems (**Ex.1** and **Ex.2**), we use PetIGA [10,28]. While the primary intent of this framework is to provide high-performance B-spline-based finite element discretizations, it is also useful for applications where the domain is topologically square and subject to uniform refinements.

	d	$\tilde{\gamma}$	q_1	q_2	s_1	s_2
Ex.1 and **Ex.2** with GMRES solver	3	1	2	4	2	0
Ex.1 and **Ex.2** with MUMPS solver	3	1.5	2	4	2.25	0
Ex.3	1	1	1	1	2	2

Table 5.1 Summary of problem parameters

As its name suggests, PetIGA is designed to tightly couple to PETSc [5,6,4]. The framework can be thought of as an extension of the PETSc library, which provides methods for assembling matrices and vectors related to the discretization of integral equations.

In our SPDE numerical tests (**Ex.1** and **Ex.2**), we use a standard trilinear basis to discretize the weak form of the model problem, integrating with eight quadrature points. We also generate results for two linear solvers that PETSc provides an interface to. The first solver is an iterative GMRES solver that solves a linear system in almost linear time with respect to the number of degrees of freedom for the mesh sizes of interest; in other words $\tilde{\gamma} = 1$ in this case. The second solver we tried is a direct one, called MUMPS [1,2]. For the mesh sizes of interest, the running time of MUMPS varies from quadratic to linear in the total number of degrees of freedom. The best fit turns out to be $\tilde{\gamma} = 1.5$ in the case.

From [33, Theorem 2.5], the complexity rate for all the examples is expected to be $\mathcal{O}(TOL^{-s_1} \log(\text{TOL})^{s_2})$, where s_1 and s_2 depend on q_1, q_2 and $d\gamma$. These and other problem parameters are summarized in Table 5.1 for the different examples.

We run each algorithm 100 times and show in plots in the next section the medians with vertical bars spanning from the 5% percentile to the 95% percentile. Finally, all results were generated on the same machine with 52 gigabytes of memory to ensure that no overhead is introduced due to hard disk access during swapping that could occur when solving the three-dimensional SPDEs with a fine mesh.

In order to compare CMLMC to SMLMC, and since the latter does not include a step to fit q_1 and q_2, we assume that these parameters are both known as discussed in Example 2.1 and Example 2.2. Moreover, we use the parameters listed in Table 5.2.

5.3 Results

Figure 5.1 shows that the running time of CMLMC follows the expected complexity rates $\mathcal{O}(\text{TOL}^{s_1} \log(\text{TOL})^{s_2})$ as summarized in Table 5.1. Next, Figure 5.2 shows the number of levels, L, in the last iteration of CMLMC for different tolerances. As expected, even though L depends on the particular realization, it is well approximated by a linear function of $\log(\text{TOL}^{-1})$.

Next, Figure 5.3 shows the computational errors of CMLMC that were computed using the reference solutions as listed in Section 5.1. This indicates that the imposed accuracy is achieved with the required confidence of 95% – since $C_\alpha = 2$. Compare this figure to Figure 5.4 which shows the computational errors of SMLMC. One can see that, in certain cases, SMLMC solves the problem for a smaller tolerance than the imposed TOL. This is because θ is fixed and the statistical error is not relaxed when the bias is small. This can be especially seen in **Ex.2** where the choice $h_0 =$

Parameter	Value for SPDE examples (**Ex.1** and **Ex.2**)	Value for SDE example (**Ex.3**)
h_0	1/4 for **Ex.1**, 1/8 for **Ex.2**	1
β	2	2
κ_0 and κ_1	0.1 for both	0.1 for both
$\text{TOL}_{\max}$	0.5	0.1
r_1 and r_2	2 and 1.1, respectively	2 and 1.1, respectively
Initial hierarchy	$L = 2$ and $h_\ell = \{4,6,8\}$ and $M_\ell = 10$ for all ℓ.	$L = 2$ and $h_\ell = \{1,2,4\}$ and $M_\ell = 10$ for all ℓ.
L_{inc}	2	2
$\mathfrak{L}$	3	5
C_α	2	2

Table 5.2 Summary of parameters values to used in numerical tests

1/8 produces a bias much smaller than 0.5TOL for the shown tolerances. On the other hand, Figure 5.5 is a QQ-plot showing that the empirical cumulative distribution function (CDF) of the MLMC estimates is well approximated by the standard normal CDF, even for finite tolerances.

Figure 5.6 shows a comparison of the running time of CMLMC and SMLMC. Notice that a good value of $\widetilde{M}$ in SMLMC is not known a priori and the computational time varies considerably for different values of $\widetilde{M}$, especially for smaller tolerances in **Ex.1** and **Ex.2**. Specifically, a larger $\widetilde{M}$ in SMLMC increases the computational time of the algorithm, but also its stability. A smaller $\widetilde{M}$ gives a smaller computational time at the expense of increased variation. The variation of the running time is due to inaccurate estimates of V_ℓ due to the smaller number of initial samples. On the other hand, the running time of CMLMC is more stable, which is a reflection of the stability of the estimates of V_ℓ. The computational savings of CMLMC over SMLMC is an aggregate effect of the different improvements. This includes 1) a more stable variance and bias estimates as already discussed, 2) a better splitting of bias and statistical tolerances. This second point can be seen in Figure 5.7, which shows the tolerance splitting parameter, θ, used in CMLMC as computed by (4.2). We can clearly see here that θ is not trivial and changes with the tolerance. Looking closely, one can notice sudden jumps in the values of θ due to changes in the discrete number of levels, L. Between jumps, θ changes continuously due to inaccuracies in the estimation of the weak error constant, Q_W. Specifically, notice that for TOL = 0.014 in **Ex.1** when using the direct solver, the splitting parameter θ used in CMLMC is very close to 0.5 which explains why, for this case, the computational time of SMLMC is very close to the computational time of CMLMC as shown in Figure 5.6.

Finally, the bias of the MLMC estimator when using samples generated in previous iterations to compute the quantity of interest is not well understood. Using CMLMC, generating new samples at each iteration, instead of using samples from previous iterations, does not add a significant overhead to the total running time of the algorithm. Figure 5.8 explains this point by comparing the running time of CMLMC for both cases for both CMLMC and SMLMC. This figure shows that computational savings of CMLMC over SMLMC whether we reuse samples or not in the former, mainly due to better splitting of the tolerance between bias and statistical errors. Moreover, it shows that reusing samples in CMLMC does not offer significant com-

putational savings that justify the increased complexity in the analysis of the resulting estimator.

6 Conclusions

We have proposed a novel Continuation Multi Level Monte Carlo (CMLMC) algorithm for weak approximation of stochastic models that are described in terms of differential equations either driven by random measures or with random coefficients. Our algorithm uses discretization hierarchies that are defined a priori for each level and are geometrically refined across levels. These hierarchies are either uniform at each level or obtained by regular subdivision of a non-uniform mesh.

The actual choice of computational work across levels uses the optimal amount of samples per level given the variance and the work contribution from each level. Accurate computation of these relevant quantities is based on parametric models.

These parameters are calibrated using approximate samples, either produced before running the CMLMC and/or during the actual runs. We also propose a novel Bayesian estimation of the strong and weak error model parameters, taking particular notice of the deepest levels of the discretization hierarchy, where only a few realizations are available to produce the required estimates. The idea is to use results from coarser levels, where more samples are available, to stabilize the estimates in the deeper levels. The resulting MLMC estimator exhibits a non-trivial splitting between bias and statistical contributions. Indeed, the actual split depends on the given accuracy and other problem parameters. In fact, as the numerical examples show, there are cases where most of the accuracy budget is devoted to the statistical error. Finally, using the Lindeberg-Feller theorem, we also show the asymptotic normality of the statistical error in the MLMC estimator and justify in this way our error estimate that allows prescribing both required accuracy and confidence in the final result.

We presented three numerical examples to substantiate the above results, exhibiting the robustness of the new CMLMC Algorithm and to demonstrate its corresponding computational savings.

Other aspects of MLMC estimators can also be explored, such as the optimality of geometric hierarchies compared to non-geometric ones. This will be the subject of a forthcoming work, where extensions of the CMLMC to that setting will be considered.

A Normality of MLMC estimator

Theorem A.1 *[11, Lindeberg-Feller Theorem, p. 114] For each n, let $X_{n,m}$, for $1 \leq n \leq m$, be independent random variables (not necessarily identical). Denote*

$$a_n = \sum_{m=1}^{n} X_{n,m},$$

$$Y_{n,m} = X_{n,m} - \mathrm{E}[X_{n,m}],$$

$$s_n^2 = \sum_{m=1}^{n} \mathrm{E}\left[Y_{n,m}^2\right].$$

Suppose the following Lindeberg condition is satisfied for all $\varepsilon > 0$:

$$\lim_{n\to\infty} s_n^{-2} \sum_{m=1}^{n} E\left[Y_{n,m}^2 \mathbf{1}_{|Y_{n,m}|>\varepsilon s_n}\right] = 0. \tag{A.1}$$

Then,

$$\lim_{n\to\infty} P\left[\frac{a_n - E[a_n]}{s_n} \leq z\right] = \Phi(z),$$

where $\Phi(z)$ is the normal cumulative density function of a standard normal random variable.

Lemma A.1 *For the MLMC estimator $\mathscr{A}$ given by*

$$\mathscr{A} = \sum_{\ell=0}^{L} \sum_{m=1}^{M_\ell} \frac{G_\ell(\omega_{\ell,m})}{M_\ell},$$

where $G_\ell(\omega_{\ell,m})$ denote as usual i.i.d. samples of the random variable G_ℓ. The the family of random variables, $(G_\ell)_{\ell\geq 0}$, is also assumed independent. Denoting $Y_\ell = |G_\ell - E[G_\ell]|$ and assuming

$$0 < E\left[Y_0^2\right], \tag{A.2a}$$

$$E\left[Y_0^{2+\delta}\right] < \infty, \tag{A.2b}$$

$$C_1 \beta^{-q_3\ell} \leq E\left[Y_\ell^2\right] \qquad\qquad \textit{for all } \ell > 0, \tag{A.2c}$$

$$E\left[Y_\ell^{2+\delta}\right] \leq C_2 \beta^{-\tau\ell} \qquad\qquad \textit{for all } \ell > 0, \tag{A.2d}$$

for some $\beta > 1$ and strictly positive constants C_1, C_2, q_3, δ and τ. Choose the number of samples on each level M_ℓ to satisfy, for $q_2 > 0$ and a strictly positive sequence $\{H_\ell\}_{\ell\geq 0}$

$$M_\ell \geq \begin{cases} \text{TOL}^{-2} H_0^{-1} \left(\sum_{\ell=0}^{L} H_\ell\right) & \textit{if } \ell = 0, \\ \beta^{-q_2\ell} \text{TOL}^{-2} H_\ell^{-1} \left(\sum_{\ell=0}^{L} H_\ell\right) & \textit{for all } \ell > 0, \end{cases} \tag{A.3}$$

If, in addition to the above, we have that

$$\textit{either} \qquad 2\tau \geq (2+\delta)q_3 + \delta q_2, \tag{A.4a}$$

$$\textit{or} \qquad L \leq \max\left(0, \frac{c\log\left(\text{TOL}^{-1}\right)}{\log\beta} + C\right) \tag{A.4b}$$

for some constants C, and c satisifying

$$0 < c < \frac{2\delta}{((2+\delta)q_3 + \delta q_2 - 2\tau)} \tag{A.5}$$

then

$$\lim_{\text{TOL}\to 0} P\left[\frac{\mathscr{A} - E[\mathscr{A}]}{\sqrt{\text{Var}[\mathscr{A}]}} \leq z\right] = \Phi(z).$$

Proof We prove this theorem by ensuring that the Lindeberg condition (A.1) is satisfied. The condition becomes in this case

$$\lim_{\text{TOL}\to 0} \underbrace{\frac{1}{\text{Var}[\mathscr{A}]} \sum_{\ell=0}^{L} \sum_{m=1}^{M_\ell} E\left[\frac{Y_\ell^2}{M_\ell^2} \mathbf{1}_{\frac{Y_\ell}{M_\ell} > \varepsilon\sqrt{\text{Var}[\mathscr{A}]}}\right]}_{:=F} = 0,$$

for all $\varepsilon > 0$. Below we make repeated use of the following identity for non-negative sequences $\{a_\ell\}$ and $\{b_\ell\}$ and $q \geq 0$.

$$\sum_\ell a_\ell^q b_\ell \leq \left(\sum_\ell a_\ell\right)^q \sum_\ell b_\ell. \tag{A.6}$$

First we use the Markov inequality to bound

$$F = \frac{1}{\text{Var}[\mathscr{A}]} \sum_{\ell=0}^{L} \sum_{m=1}^{M_\ell} \text{E}\left[\frac{Y_\ell^2}{M_\ell^2} \mathbf{1}_{Y_\ell > \varepsilon\sqrt{\text{Var}[\mathscr{A}]}M_\ell}\right]$$

$$\leq \frac{\varepsilon^{-\delta}}{\text{Var}[\mathscr{A}]^{1+\delta/2}} \sum_{\ell=0}^{L} M_\ell^{-1-\delta} \text{E}\left[Y_\ell^{2+\delta}\right].$$

Using (A.6) and substituting for the variance $\text{Var}[\mathscr{A}]$ where we denote $\text{Var}[G_\ell] = \text{E}\left[(G_\ell - \text{E}[G_\ell])^2\right]$ by V_ℓ, we find

$$F \leq \frac{\varepsilon^{-\delta} \left(\sum_{\ell=0}^{L} M_\ell^{-1} V_\ell\right)^{1+\delta/2}}{\left(\sum_{\ell=0}^{L} V_\ell M_\ell^{-1}\right)^{1+\delta/2}} \sum_{\ell=0}^{L} V_\ell^{-1-\delta/2} M_\ell^{-\delta/2} \text{E}\left[Y_\ell^{2+\delta}\right]$$

$$\leq \varepsilon^{-\delta} \sum_{\ell=0}^{L} V_\ell^{-1-\delta/2} M_\ell^{-\delta/2} \text{E}\left[Y_\ell^{2+\delta}\right].$$

Using the lower bound on the number of samples M_ℓ (A.3) and (A.6) again yields

$$F \leq \varepsilon^{-\delta} \text{TOL}^\delta \left(V_0^{-1-\delta/2} H_0^{\delta/2} \text{E}\left[Y_0^{2+\delta}\right] + \sum_{\ell=1}^{L} V_\ell^{-1-\delta/2} \beta^{\frac{\delta q_2 \ell}{2}} H_\ell^{\delta/2} \text{E}\left[Y_\ell^{2+\delta}\right]\right) \left(\sum_{\ell=0}^{L} H_\ell\right)^{-\delta/2}$$

$$\leq \varepsilon^{-\delta} \text{TOL}^\delta \left(V_0^{-1-\delta/2} \text{E}\left[Y_0^{2+\delta}\right] + \sum_{\ell=1}^{L} V_\ell^{-1-\delta/2} \beta^{(\delta/2)q_2\ell} \text{E}\left[Y_\ell^{2+\delta}\right]\right).$$

Finally using the bounds (A.2c) and (A.2d)

$$F \leq \varepsilon^{-\delta} \text{TOL}^\delta \left(V_0^{-1-\delta/2} \text{E}\left[Y_0^{2+\delta}\right] + C_1^{-1-\delta/2} C_2 \sum_{\ell=1}^{L} \beta^{(1+\delta/2)q_3} \beta^{(\delta/2)q_2\ell} \beta^{-\tau\ell}\right)$$

$$= \varepsilon^{-\delta} \text{TOL}^\delta \left(V_0^{-1-\delta/2} \text{E}\left[Y_0^{2+\delta}\right] + C_1^{-1-\delta/2} C_2 \beta^p \frac{\beta^{Lp}-1}{\beta^p-1}\right),$$

where

$$p = (1+\delta/2)q_3 + (\delta/2)q_2 - \tau.$$

We distinguish two cases here, namely:

- If (A.4a) is satisfied then $\lim_{\text{TOL}\to 0} F = 0$ for any choice of number of levels $L \geq 0$.
- Otherwise, substituting (A.4b) gives

$$F \leq \varepsilon^{-\delta} \text{TOL}^\delta \left(V_0^{-1-\delta/2} \text{E}\left[Y_0^{2+\delta}\right] + C_3 \beta^p \frac{\text{TOL}^{-cp}\beta^{Cp}-1}{\beta^p-1}\right)$$

$$= \mathscr{O}\left(\text{TOL}^{\delta-cp}\right),$$

and since in this case (A.5) is satisfied then $\lim_{\text{TOL}\to 0} F = 0$. $\qquad\square$

Remark A.1 The choice (A.3) mirrors the choice (2.7) up to constants, the latter being the optimal number of samples to bound the statisitcal error of the estimator by TOL. Specifically, $H_\ell \propto \sqrt{V_\ell W_\ell}$ where W_ℓ is the work per sample on level ℓ. Moreover, the choice (2.7) uses the variances $\{V_\ell\}_{\ell=0}^{L}$ or an estimate of it in the actual implementation. On the other hand, the choice (A.3) uses the upper bound of V_ℓ instead, if q_2 is the rate of strong convergence therein. Furthermore, if we assume the weak error model (2.9a) holds and $h_L = h_0\beta^{-L}$ then we must have

$$Q_W h_L^{q_1} = Q_W h_0^{q_1} \beta^{-Lq_1} \leq (1-\theta)\text{TOL},$$

which gives a lower bound on the number of levels L, namely

$$L \geq \frac{\log(\text{TOL}^{-1})}{q_1 \log(\beta)} + \frac{-\log(1-\theta) + \log(Q_W) + q_1\log(h_0)}{q_1\log(\beta)},$$

to bound the bias by TOL.

Finally, in Example 2.1 the conditions (A.2) are satisfied for $q_3 = 2$ and, for example, $\delta = 2$ and $\tau = 4$. Similarly, Example 2.2 satisfies the conditions (A.2) are for $q_3 = 1$ and $\delta = 2$ and $\tau = 2$, cf. [21].

Remark A.2 The assumption (A.2c) can be relaxed. For instance, one can assume instead that

$$V_{\ell+1} \leq V_\ell \qquad \text{for all } \ell \geq 1,$$
$$0 < \lim_{\ell \to \infty} \mathrm{Var}[Y_\ell]\beta^{q_3 \ell} < \infty,$$

and slightly different conditions than (A.4) and (A.5).

References

1. Amestoy, P.R., Duff, I.S., L'Excellent, J.Y., Koster, J.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl. **23**, 15–41 (2001). DOI 10.1137/S0895479899358194. URL http://portal.acm.org/citation.cfm?id=587708.587825
2. Amestoy, P.R., Guermouche, A., L'Excellent, J.Y., Pralet, S.: Hybrid scheduling for the parallel solution of linear systems. Parallel Computing **32**(2), 136 – 156 (2006). DOI DOI: 10.1016/j.parco.2005.07.004. URL http://www.sciencedirect.com/science/article/pii/S0167819105001328. Parallel Matrix Algorithms and Applications (PMAA'04)
3. Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Journal on Numerical Analysis **45**(3), 1005–1034 (2007)
4. Balay, S., Brown, J., , Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory (2013)
5. Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc Web page (2013). Http://www.mcs.anl.gov/petsc
6. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (eds.) Modern Software Tools in Scientific Computing, pp. 163–202. Birkhäuser Press (1997)
7. Barth, A., Schwab, C., Zollinger, N.: Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. Numerische Mathematik **119**(1), 123–161 (2011)
8. Charrier, J., Scheichl, R., Teckentrup, A.: Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods. SIAM Journal on Numerical Analysis **51**(1), 322–352 (2013)
9. Cliffe, K., Giles, M., Scheichl, R., Teckentrup, A.: Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. Computing and Visualization in Science **14**(1), 3–15 (2011)
10. Dalcin, L., Collier, N.: PetIGA: A framework for high performance isogeometric analysis (2013). Https://bitbucket.org/dalcinl/petiga
11. Durrett, R.: Probability: theory and examples. second edn. Duxbury Press, Belmont, CA (1996)
12. Giles, M.: Multilevel Monte Carlo path simulation. Operations Research **56**(3), 607–617 (2008)
13. Giles, M., Reisinger, C.: Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance. SIAM Journal of Financial Mathematics **3**(1), 572–592 (2012)
14. Giles, M., Szpruch, L.: Antithetic multilevel Monte Carlo estimation for multidimensional SDEs. In: Monte Carlo and Quasi-Monte Carlo Methods 2012 (submitted). Springer (2013)
15. Giles, M., Szpruch, L.: Multilevel Monte Carlo methods for applications in finance, pp. 3–48. World Scientific (2013)
16. Giles, M., Szpruch, L.: Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation. To appear in Annals of Applied Probability (2013/4)

17. Giles, M.B.: Improved multilevel monte carlo convergence using the milstein scheme. Monte Carlo and Quasi-Monte Carlo Methods – Check this year! pp. 343–358 (2006)
18. Glasserman, P.: Monte Carlo methods in financial engineering, *Applications of Mathematics (New York)*, vol. 53. Springer-Verlag, New York (2004). Stochastic Modelling and Applied Probability
19. Heinrich, S.: Monte Carlo complexity of global solution of integral equations. Journal of Complexity **14**(2), 151–175 (1998)
20. Heinrich, S., Sindambiwe, E.: Monte Carlo complexity of parametric integration. Journal of Complexity **15**(3), 317–341 (1999)
21. Hoel, H., Schwerin, E.v., Szepessy, A., Tempone, R.: Adaptive multilevel Monte Carlo simulation. In: Engquist, B., Runborg, O., Tsai, Y.H. (eds.) Numerical Analysis of Multiscale Computations, no. 82 in Lecture Notes in Computational Science and Engineering, pp. 217–234. Springer (2012)
22. Hoel Håkon, v.S.E.S.A., Tempone, R.: Adaptive weak approximation of stochastic differential equations. Monte Carlo Methods and Applications (2014)
23. Hunter, J.D.: Matplotlib: A 2D graphics environment. Computing In Science & Engineering **9**(3), 90–95 (2007)
24. Jouini, E., Cvitanić, J., Musiela, M. (eds.): Option pricing, interest rates and risk management. Handbooks in Mathematical Finance. Cambridge University Press, Cambridge (2001)
25. Karatzas, I., Shreve, S.E.: Brownian motion and stochastic calculus, *Graduate Texts in Mathematics*, vol. 113. Second edn. Springer-Verlag, New York (1991)
26. Kebaier, A.: Statistical Romberg extrapolation: A new variance reduction method and applications to option pricing. Ann. Appl. Probab. **15**(4), 2681–2705 (2005). URL http://projecteuclid.org/euclid.aoap/1133965776
27. Mordecki, E., Szepessy, A., Tempone, R., Zouraris, G.E.: Adaptive weak approximation of diffusions with jumps. SIAM J. Numer. Anal. **46**(4), 1732–1768 (2008)
28. N. Collier L. Dalcin, V.C.: PetIGA: High-performance isogeometric analysis. arxiv (1305.4452) (2013). Http://arxiv.org/abs/1305.4452
29. Øksendal, B.: Stochastic differential equations. Universitext, fifth edn. Springer-Verlag, Berlin (1998). An introduction with applications
30. Oliphant, T.E.: Guide to NumPy. Trelgol Publishing (2006)
31. Sivia, D.S.: Data Analysis.: A Bayesian Tutorial. Oxford University Press (1996)
32. Teckentrup, A.: Multilevel Monte Carlo methods and uncertainty quantification. PhD thesis, University of Bath (2013)
33. Teckentrup, A., Scheichl, R., Giles, M., Ullmann, E.: Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. Numerische Mathematik **125**(3), 569–600 (2013)
34. Xia, Y., Giles, M.: Multilevel path simulation for jump-diffusion SDEs. In: Plaskota, L., Woźniakowski, H. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 695–708. Springer (2012)
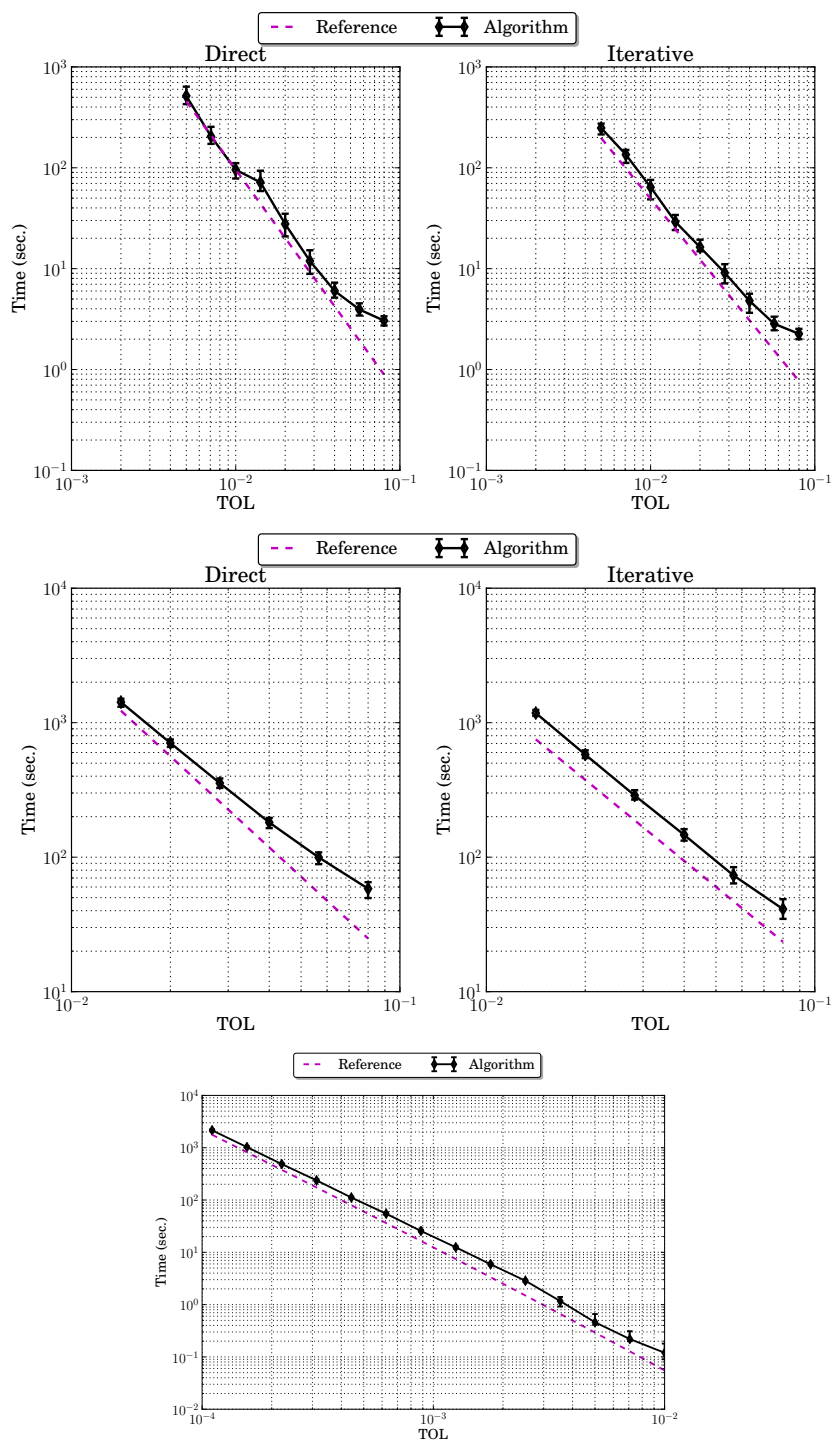
Fig. 5.1 From top: **Ex.1, Ex.2, Ex.3**. These plots show the running time of CMLMC and its the last iteration. The reference dashed line is $\mathcal{O}\left(\mathrm{TOL}^{-s_1}\log(\mathrm{TOL})^{s_2}\right)$ as summarized in Table 5.1. Notice that, asymptotically, the running times seem to follow the expected rates.
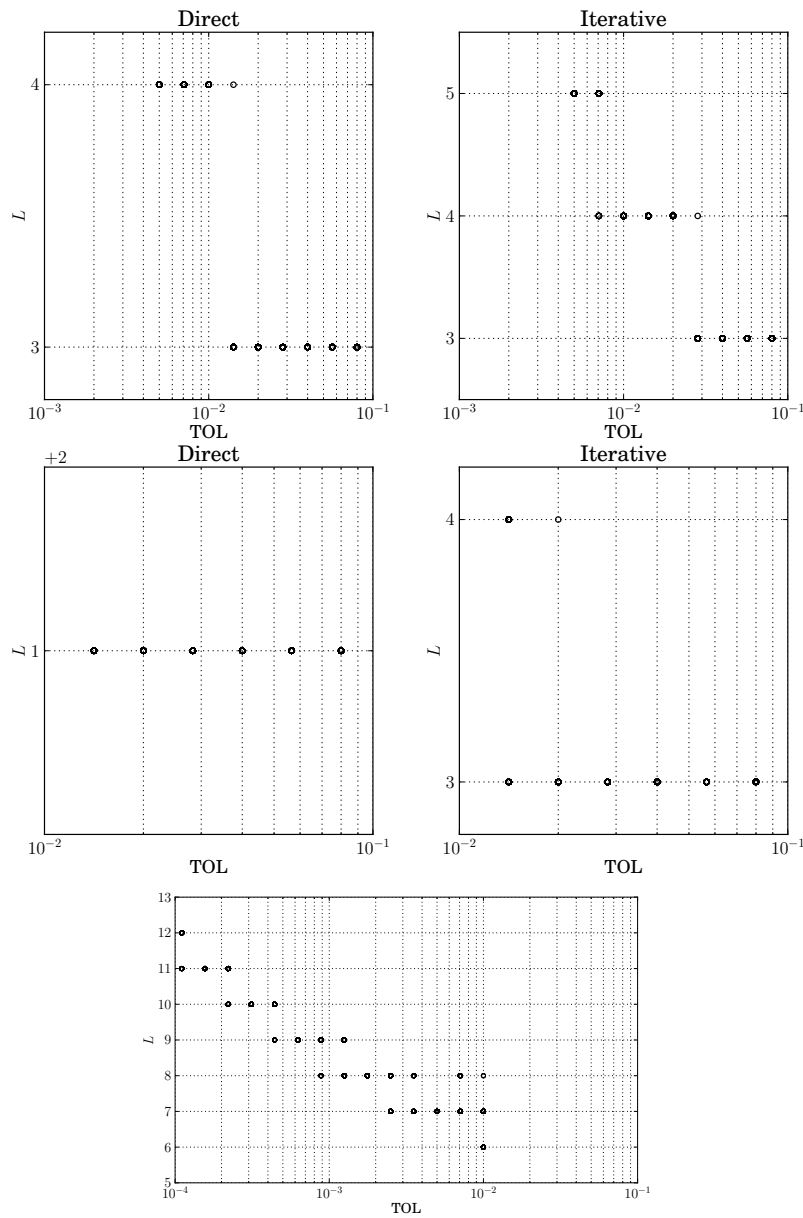
Fig. 5.2 From top: **Ex.1, Ex.2, Ex.3**. These plots show the number of levels, L, for different tolerances, as produced in the last iteration of CMLMC. Here, it is clear that L is depends on the particular realization. However, the relation between L and $\log(\text{TOL})$ looks linear, as expected.
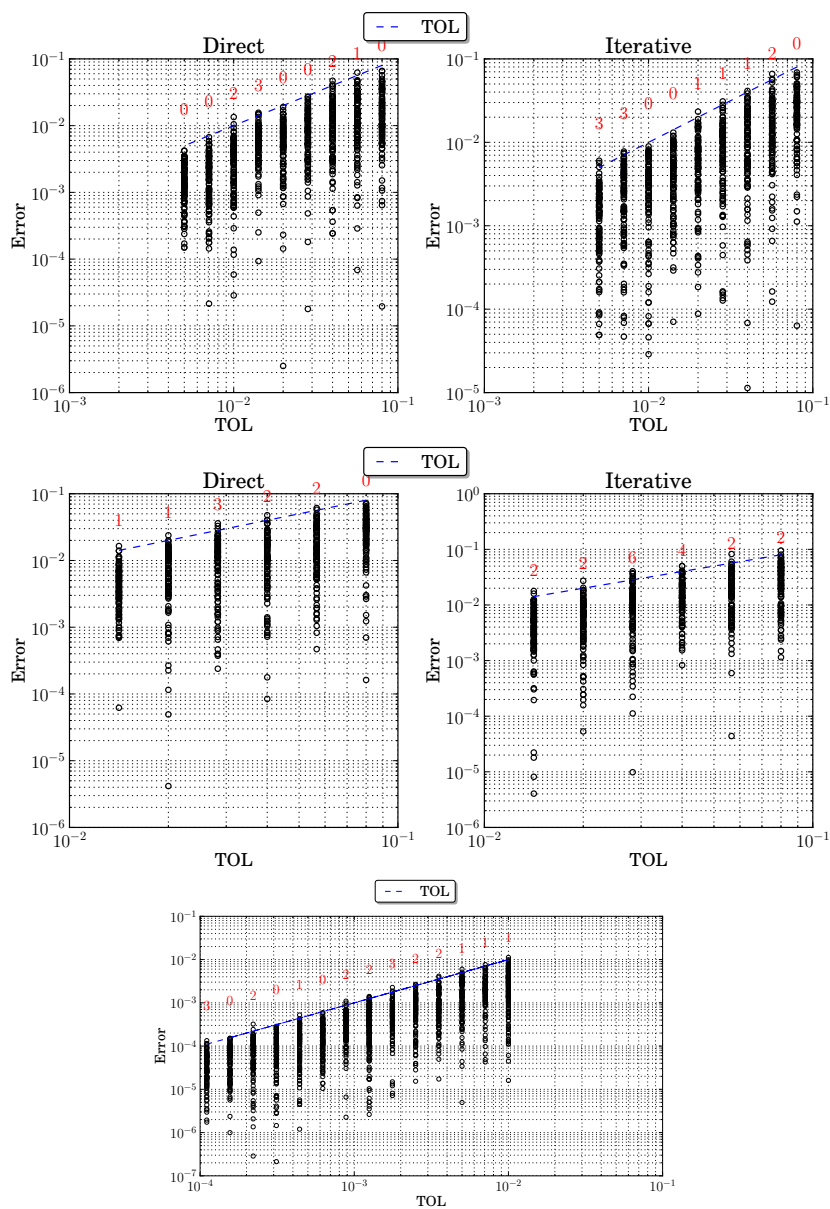
Fig. 5.3 From top: **Ex.1, Ex.2, Ex.3**. Actual computational errors based on the reference solutions when using CMLMC. The numbers above the dashed line show the percentage of runs that had errors larger than the required tolerance. We observe that in all cases the computational error follows the imposed tolerance closely with the expected confidence of 95%.
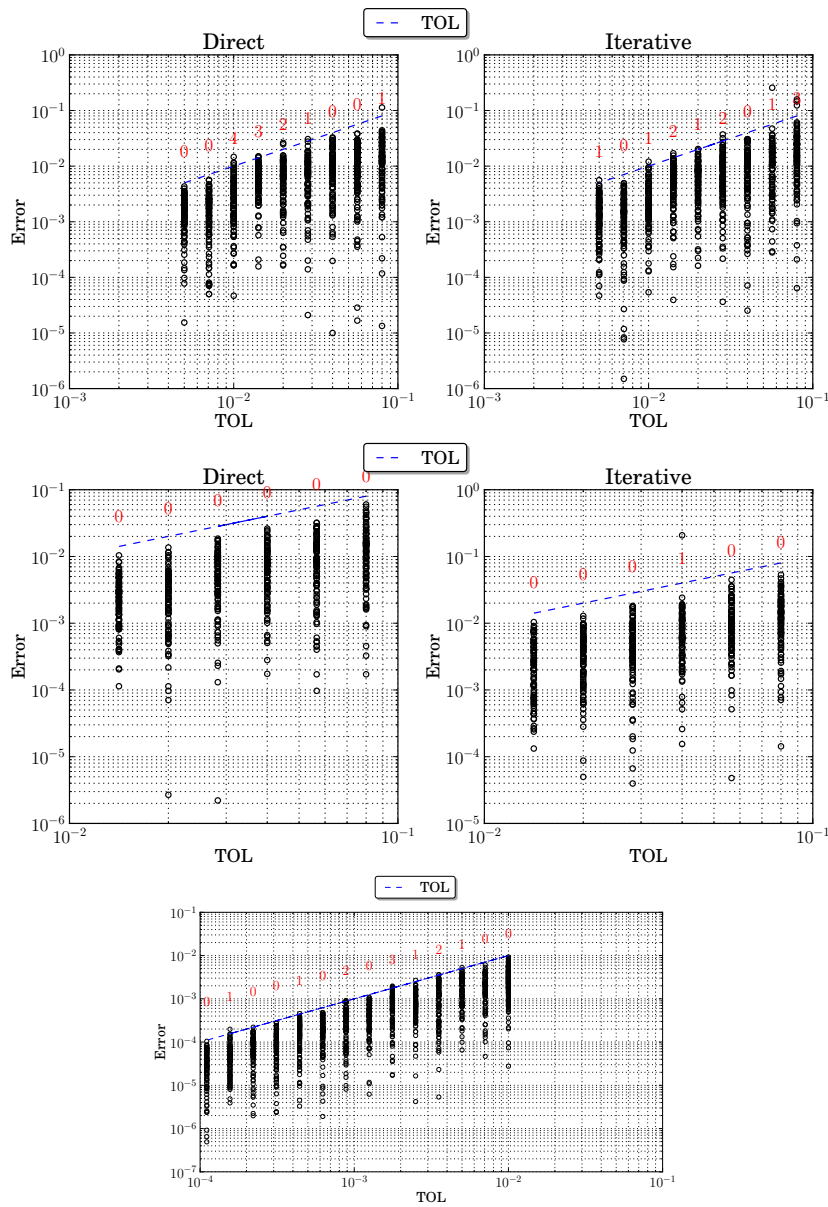
Fig. 5.4 From top: **Ex.1, Ex.2, Ex.3**. Actual computational errors based on the reference solutions when using SMLMC. The numbers above the dashed line show the percentage of runs that had errors larger than the required tolerance. We observe that in most cases the computational errors are below the imposed tolerance with the expected confidence of 95%. For particular tolerances, the error is smaller than TOL because the statistical error is not relaxed when the bias is small since θ is constant.
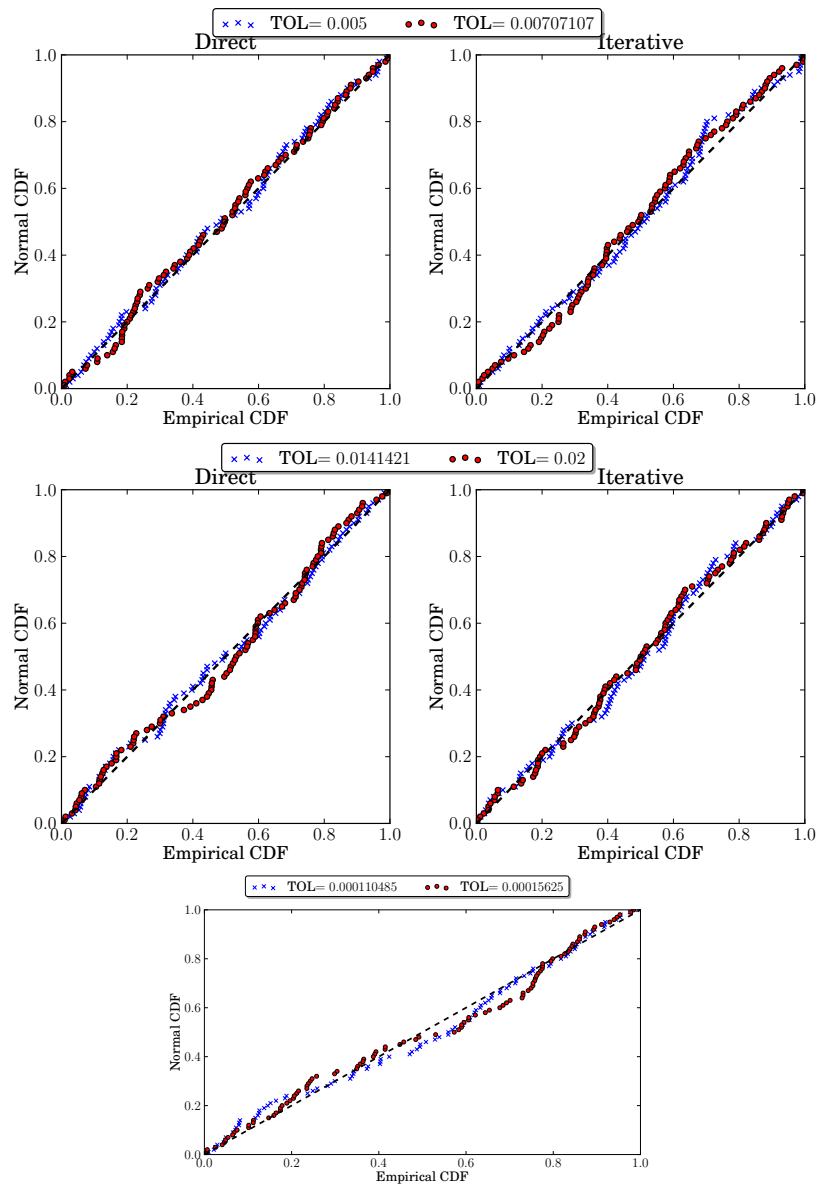
Fig. 5.5 From top: **Ex.1, Ex.2, Ex.3**. Normalized empirical cumulative distribution function (CDF) of MLMC estimates for different tolerances versus the standard normal CDF.
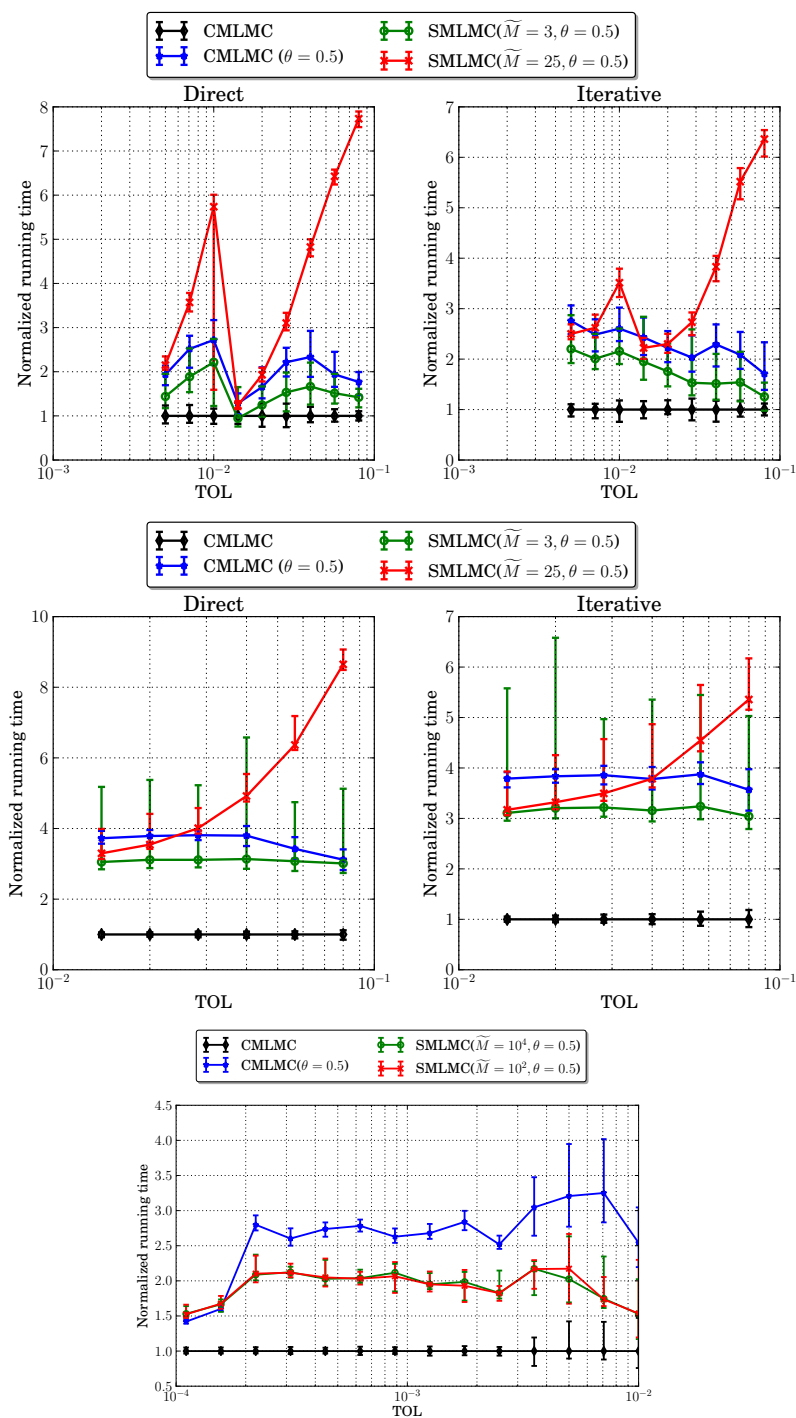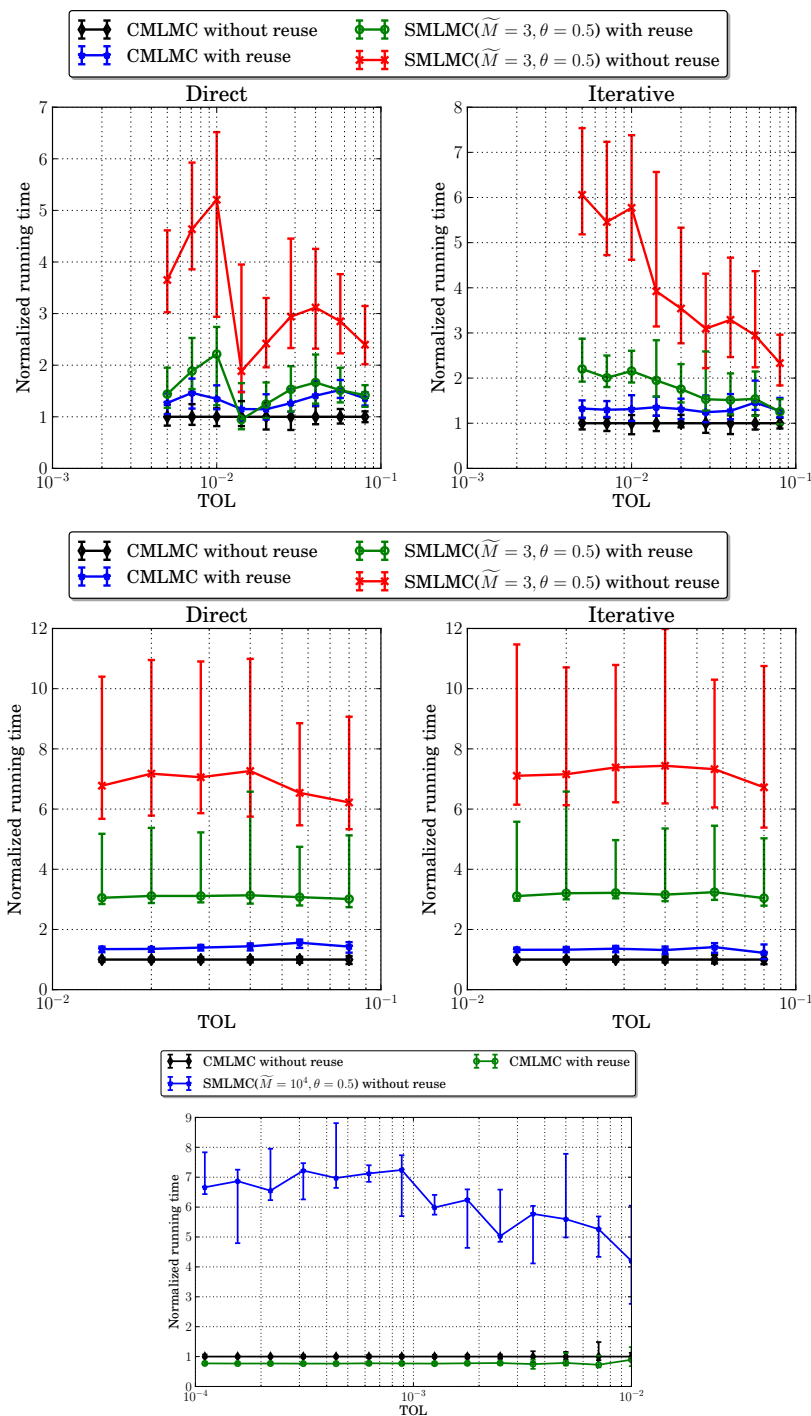
Fig. 5.6 From top: **Ex.1, Ex.2, Ex.3**. The running time of CMLMC and SMLMC for different $\widetilde{M}$ and θ, normalized by the median running time of CMLMC. This plot shows that a larger $\widetilde{M}$ increases the median running time of the SMLMC but also increases its stability. One sees that CMLMC outperforms SMLMC even for a small $\widetilde{M}$ in all numerical examples.
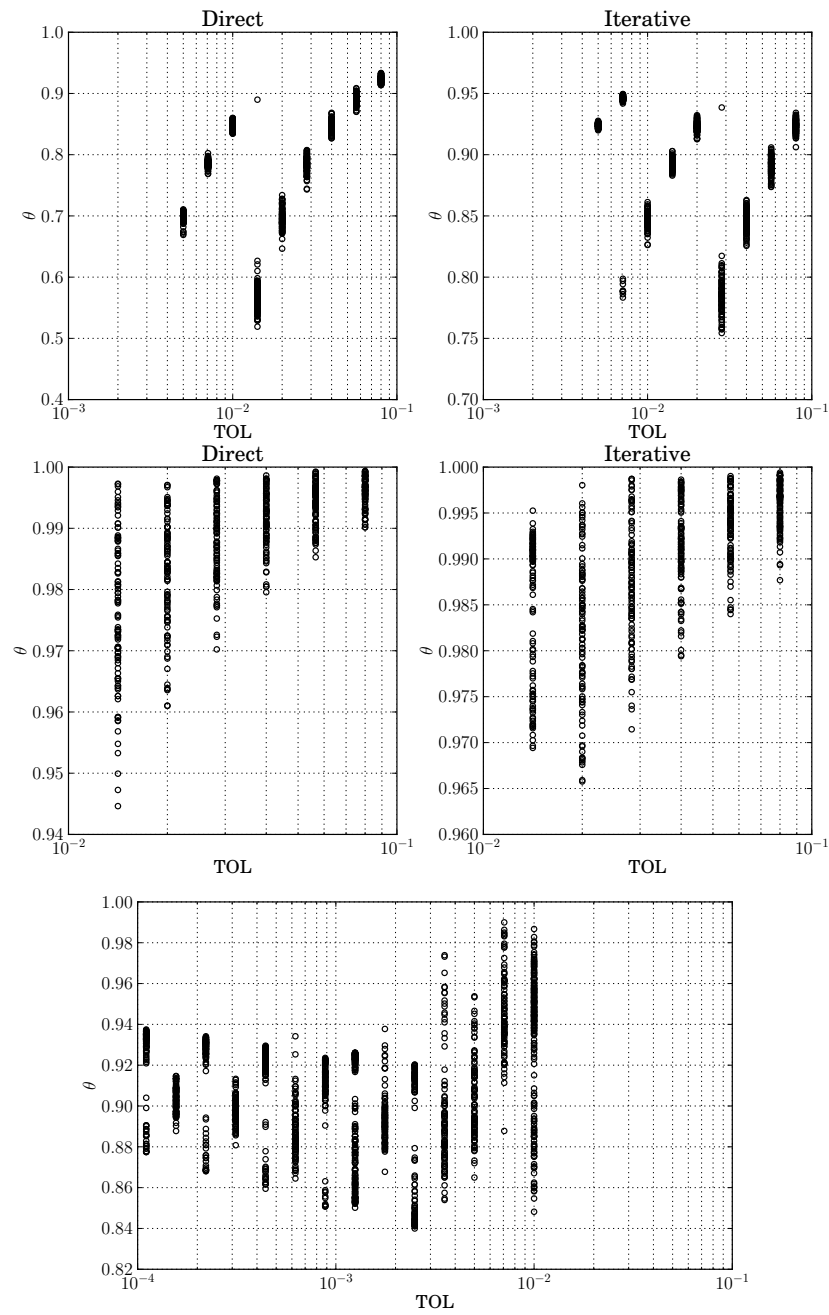
Fig. 5.7 From top: **Ex.1, Ex.2, Ex.3**. The error splitting, θ, as computed in (4.2) an used in CMLMC, versus TOL.

Fig. 5.8 From top: **Ex.1, Ex.2, Ex.3**. Running time of CMLMC versus SMLMC when reusing samples for both. Also included, is CMLMC without reusing samples. All running times are normalized by the median of the running time of CMLMC without reusing samples. Notice that reusing samples in CMLMC does not add a significant advantage. Moreover, CMLMC still produces savings over SMLMC, even when reusing samples in the latter.

43.2013   P. PACCIARINI, G. ROZZA:
Stabilized reduced basis method for parametrized advection-diffusion PDEs

44.2013   A. KOSHAKJI, A. QUARTERONI, G. ROZZA:
Free form deformation techniques applied to 3D shape optimization problems

45.2013   J. E. CATRILLON-CANDAS, F. NOBILE, R. F. TEMPONE:
Analytic regularity and collocation approximation for PDEs with random domain deformations

01.2014   GIOVANNI MIGLIORATI:
Multivariate Markov-type and Nicolskii-type inequalities for polynomials associated with downward closed multi-index sets

02.2014   FEDERICO NEGRI, ANDREA MANZONI, GIANLUIGI ROZZA:
Certified reduced basis method for parametrized optimal control problems governed by the Stokes equations

03.2014   CEDRIC EFFENBERGER, DANIEL KRESSNER:
On the residual inverse iteration for nonlinear eigenvalue problems admitting a Rayleigh functional

04.2014   TAKAHITO KASHIWABARA, CLAUDIA M. COLCIAGO, LUCA DEDÈ, ALFIO QUARTERONI:
Numerical Well-posedness, regularity, and convergence analysis of the finite element approximation of a generalized Robin boundary value problem

05.2014   BJÖRN ADLERBORN, BO KAGSTRÖM, DANIEL KRESSNER:
A parallel QZ algorithm for distributed memory HPC systems

06.2014   MICHELE BENZI, SIMONE DEPARIS, GWENOL GRANDPERRIN, ALFIO QUARTERONI:
Parameter estimates for the relaxed dimensional factorization preconditioner and application to hemodynamics

07.2014   ASSYR ABDULLE, YUN BAI:
Reduced order modelling numerical homogenization

08.2014   ANDREA MANZONI, FEDERICO NEGRI:
Rigorous and heuristic strategies for the apporximation of stability factors in nonlinear parametrized PDEs

09.2014   PENG CHEN, ALFIO QUARTERONI:
A new algorithm for high-dimensional uncertainty quantification problems based on dimension-adaptive and reduced basis methods

10.2014   NATHAN COLLIER, ABDUL-LATEEF HAJI-ALI, FABIO NOBILE, ERIK VON SCHWERIN, RAÚL TEMPONE:
A continuation multilevel Monte Carlo algorithm