

Uncertainty Quantification by MLMC and Local Time-stepping For Wave Propagation

Marcus J. Grote*, Simon Michel†, and Fabio Nobile†

Abstract. Because of their robustness, efficiency and non-intrusiveness, Monte Carlo methods are probably the most popular approach in uncertainty quantification to computing expected values of quantities of interest (QoIs). Multilevel Monte Carlo (MLMC) methods significantly reduce the computational cost by distributing the sampling across a hierarchy of discretizations and allocating most samples to the coarser grids. For time dependent problems, spatial coarsening typically entails an increased time-step. Geometric constraints, however, may impede uniform coarsening thereby forcing some elements to remain small across all levels. If explicit time-stepping is used, the time-step will then be dictated by the smallest element on each level for numerical stability. Hence, the increasingly stringent CFL condition on the time-step on coarser levels significantly reduces the advantages of the multilevel approach. To overcome that bottleneck we propose to combine the multilevel approach of MLMC with local time-stepping (LTS). By adapting the time-step to the locally refined elements on each level, the efficiency of MLMC methods is restored even in the presence of complex geometry without sacrificing the explicitness and inherent parallelism. In a careful cost comparison, we quantify the reduction in computational cost for local refinement either inside a small fixed region or towards a reentrant corner.

Key words. Uncertainty quantification, Multilevel Monte Carlo, wave propagation, finite element methods, local time-stepping, explicit time integration.

AMS subject classifications. 65C05, 65L06, 65M20, 65M60, 65M75.

1. Introduction. Mathematical models based on partial differential equations (PDE) are widely used to describe complex phenomena and to make predictions in real-world applications. All such mathematical models, however, are affected by a certain degree of uncertainty that may arise because of imperfect characterization or intrinsic variability of model parameters, constitutive laws, forcing terms, initial states, etc. Uncertainty is typically included in PDE based models by replacing input parameters by stochastic variables or processes. That uncertainty is then propagated across space (and possibly time) by the solution u of the corresponding stochastic PDE and thereby determines the uncertainty in any observed quantity of interest (QoI) $Q[u]$.

In mathematical models from acoustics, electromagnetics or elasticity, waves, as ubiquitous information carriers, will also propagate uncertainty about input parameters over long distances with little regularizing or smoothing effects. The inherent lack of regularity hampers the use of computational uncertainty quantification methods, such as polynomial chaos expansions or sparse quadratures, which rely on the smoothness of the so-called *parameter-to-solution* map and/or on the low dimensionality of the input space [46, 36, 37].

In contrast, Monte Carlo (MC) methods [15], probably the most popular alternative to

*Department of Mathematics and Computer Science, University of Basel, Spiegelgasse 1, 4051 Basel, Switzerland (marcus.grote@unibas.ch, simon.michel@unibas.ch)

†Calcul Scientifique et Quantification de l'Incertitude (CSQI), Institute of Mathematics, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (fabio.nobile@epfl.ch)

quantifying the uncertainty in any QoI, are robust to the dimension of the input parameters and the lack of regularity of the parameter-to-QoI map. By drawing independent realizations from the input probability distribution on a sample space Ω , they compute for each sample $\omega \in \Omega$ a solution $u(\omega, \cdot)$ of the forward problem and thus permit to estimate the statistics of the QoI $Q[u(\omega, \cdot)]$. Although MC methods are easy to implement, their convergence in the number of samples is rather slow.

Multilevel Monte Carlo (MLMC) methods, first introduced for applications in parametric integration by Heinrich [26, 27] and later extended by Giles in his seminal paper [16] to multi-level approximations of stochastic differential equations, significantly reduce the computational cost by distributing the sampling across a hierarchy of discretizations and computing most samples on coarser grids. In recent years, MLMC methods thus have proved extremely efficient, versatile and robust for uncertainty quantification (UQ) in a wide range of problems governed by stochastic PDEs, including, to name just a few, elliptic equations with random coefficients [9, 45, 6], parabolic PDEs [5], conservation laws and compressible aerodynamics [32, 33, 40], acoustic and seismic wave propagation [34, 4], obstacle problems [7], and multiscale problems [2].

For time dependent problems, spatial coarsening on higher levels usually entails a larger time-step, thereby reducing even further the computational cost of individual sample solutions $u(t, x; \omega)$. Geometric constraints or singularities, however, may impede uniform coarsening, thus forcing some elements in the mesh to remain small across all levels. If explicit time-stepping is used, the time-step will then be dictated by the smallest element on each level due to the CFL stability condition. Hence, standard explicit time-stepping schemes will become increasingly inefficient on coarser levels due to the ever more restrictive CFL condition.

To overcome the increasingly stringent bottleneck across all levels, we propose to use local time-stepping (LTS) [12, 18] methods for the time integration on each level. By using a smaller time-step only inside the locally refined region, LTS methods thus permit to greatly improve the efficiency of MLMC methods even in the presence of complex geometry without sacrificing the explicitness and inherent parallelism. We carefully analyze the computational cost of the MLMC algorithm in the presence of locally refined meshes, first with a standard explicit time-stepping method and then with a local time-stepping method. In doing so, we differentiate between local refinement inside a small fixed region or towards a reentrant corner. In the former case, we quantify the gain of using LTS over standard time-stepping methods, whereas in the latter we prove that LTS even improves the asymptotic complexity. In a series of numerical experiments, we illustrate the significant gain over standard time-stepping obtained by using LTS methods for the time integration at all levels.

The rest of our paper is structured as follows. In Section 2, we recall the standard MLMC Algorithm [16] to estimate a generic QoI in any (finite or infinite dimensional) Hilbert space; for instance, the QoI may be a functional of u , or the solution itself. In Section 3, we present the cost comparison between MLMC with or without local time-stepping. Finally, in Section 4, we present numerical experiments including an example with complex geometry in two space dimensions.

2. Multilevel Monte Carlo method for wave equations with random coefficients.

2.1. Model problem. We consider the wave equation with stochastic coefficient in a bounded domain $D \subset \mathbb{R}^d$,

$$(2.1) \quad \begin{cases} \frac{\partial^2}{\partial t^2} u(\mathbf{x}, t, \omega) - \nabla \cdot (c^2(\mathbf{x}, \omega) \nabla u(\mathbf{x}, t, \omega)) = f(\mathbf{x}, t) & \forall \mathbf{x} \in D, t \in (0, T], \omega \in \Omega, \\ u(\mathbf{x}, 0, \omega) = u_0(\mathbf{x}) & \forall \mathbf{x} \in D, \omega \in \Omega, \\ \frac{\partial}{\partial t} u(\mathbf{x}, 0, \omega) = v_0(\mathbf{x}) & \forall \mathbf{x} \in D, \omega \in \Omega \end{cases}$$

with appropriate (deterministic) boundary conditions. Here, we model the uncertainty in the wave speed $c > 0$ as a time independent random field $c : D \times \Omega \rightarrow \mathbb{R}$, where Ω is the sample space of a complete probability space. More general models that also allow randomness in the geometry, the forcing term, or an acceleration term could also be considered. We are interested in estimating the expected value $\mathbb{E}[Q]$ of some quantity of interest (QoI) $Q : \omega \mapsto Q(\omega) = Q(u(\cdot, \cdot, \omega)) \in V$ related to the solution $u = u(\mathbf{x}, t, \omega)$.

We consider a generic case where V is a Hilbert space. For instance, if Q is the value of some functional of u , we simply set $V = \mathbb{R}$ (cf. [35, 45]). On the other hand, if the QoI is the (weak) solution itself at a fixed time $T > 0$, $Q(\omega) := u(\cdot, T, \omega) \in V = H^1(D) \forall \omega \in \Omega$, we let V correspond to the solution space. This setting can easily be generalized to the (weaker) case when $Q(\omega) \in V$ for almost every $\omega \in \Omega$.

2.2. Construction of the MLMC method. To derive the MLMC approximation for (2.1), let Q_h denote a (numerical) finite element approximation to the QoI $Q = Q(u)$, with h the discrete mesh size. To estimate $\mathbb{E}[Q]$, one computes approximations or estimators \widehat{Q}_h to $\mathbb{E}[Q_h]$. The accuracy of the approximations is quantified by the mean square error (MSE)

$$(2.2) \quad e(\widehat{Q}_h)^2 := \mathbb{E} \left[\left\| \widehat{Q}_h - \mathbb{E}[Q] \right\|_V^2 \right].$$

The main idea of the MLMC method is to sample the QoI Q from several approximations $Q_\ell := Q_{H_\ell}$ on a sequence of discretizations $\ell = 0, 1, \dots, L$. Then, each level uses its individual mesh size $H_\ell = H_0/2^\ell$ in space and time-step Δt_ℓ in time, where the latter must satisfy a standard CFL condition, $\Delta t_\ell \leq CH_\ell$, for numerical stability, if explicit time-stepping is used.

For the approximate solution on the finest level with mesh size H_L , it holds that

$$\mathbb{E}[Q_L] = \mathbb{E}[Q_0] + \sum_{\ell=1}^L \mathbb{E}[Q_\ell - Q_{\ell-1}] = \sum_{\ell=0}^L \mathbb{E}[\Delta Q_\ell]$$

with

$$\Delta Q_\ell := \begin{cases} Q_0, & \ell = 0, \\ Q_\ell - Q_{\ell-1}, & \ell = 1, \dots, L, \end{cases}$$

random variables on Ω . This motivates the MLMC estimator of $\mathbb{E}[Q_L]$,

$$(2.3) \quad \widehat{Q}_h^{\text{ML}} : \Omega^M \rightarrow V$$

$$\widehat{\omega} := \left\{ \omega^{(i,\ell)} \mid i = 1, \dots, N_\ell, \ell = 0, \dots, L \right\} \mapsto \sum_{\ell=0}^L \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(\Delta Q_\ell \left(\omega^{(i,\ell)} \right) \right),$$

where $M = \sum_\ell N_\ell$ with N_ℓ denoting the size of the sample on each level ℓ and where the probability measure on Ω^M is the tensor product measure $\mathbb{P}^{\otimes M}$, i.e. $\widehat{\omega}$ is an independent sample. For efficiency of the estimator, it is crucial to judiciously choose the parameter values N_ℓ and L . If we set $\mu = \mathbb{E}[Q] \in V$ in (2.2), we obtain

$$\begin{aligned} e \left(\widehat{Q}_h^{\text{ML}} \right)^2 &= \mathbb{E} \left[\left\langle \widehat{Q}_h^{\text{ML}} - \mu, \widehat{Q}_h^{\text{ML}} - \mu \right\rangle_V \right] \\ &= \mathbb{E} \left[\left\langle \widehat{Q}_h^{\text{ML}} - \mathbb{E}[Q_L] + \mathbb{E}[Q_L] - \mu, \widehat{Q}_h^{\text{ML}} - \mathbb{E}[Q_L] + \mathbb{E}[Q_L] - \mu \right\rangle_V \right], \\ &= \mathbb{E} \left[\left\| \widehat{Q}_h^{\text{ML}} - \mathbb{E}[Q_L] \right\|_V^2 \right] + \|\mathbb{E}[Q_L] - \mu\|_V^2 + 2 \mathbb{E} \left[\left\langle \widehat{Q}_h^{\text{ML}} - \mathbb{E}[Q_L], \mathbb{E}[Q_L] - \mu \right\rangle_V \right], \end{aligned}$$

where the expectation is understood with respect to the product probability measure $\mathbb{P}^{\otimes M}$ and the last term equals to zero as $\mathbb{E}[\widehat{Q}_h^{\text{ML}}] = \mu$. For the first term in the last equation, we now insert the definition of the MLMC estimator (2.3), where we denote by $Q^{(i,\ell)}$, $i = 1, \dots, N_\ell$, the independent, identically distributed (i.i.d.) ‘‘copies’’ of the random variable Q on level ℓ . Thus, we obtain

$$\begin{aligned} e \left(\widehat{Q}_h^{\text{ML}} \right)^2 &= \mathbb{E} \left[\left\| \sum_{\ell=0}^L N_\ell^{-1} \sum_{i=1}^{N_\ell} \left(\Delta Q_\ell^{(i,\ell)} - \mathbb{E}[\Delta Q_\ell] \right) \right\|_V^2 \right] + \|\mathbb{E}[Q_L - Q]\|_V^2 \\ &= \sum_{\ell=0}^L N_\ell^{-1} \mathbb{E} \left[\|\Delta Q_\ell - \mathbb{E}[\Delta Q_\ell]\|_V^2 \right] + \|\mathbb{E}[Q_L - Q]\|_V^2. \end{aligned}$$

Note that the last step is due to the fact that $\Delta Q_\ell^{(i,\ell)}$ are indeed i.i.d. Let

$$(2.4) \quad V_\ell = \mathbb{E} \left[\|\Delta Q_\ell - \mathbb{E}[\Delta Q_\ell]\|_V^2 \right] = \mathbb{E} \left[\|\Delta Q_\ell\|_V^2 \right] - \|\mathbb{E}[\Delta Q_\ell]\|_V^2$$

denote the variance on a level $\ell = 0, 1, \dots, L$. Then, the mean square error can be split as

$$(2.5) \quad e \left(\widehat{Q}_h^{\text{ML}} \right)^2 = \sum_{\ell=0}^L N_\ell^{-1} V_\ell + \|\mathbb{E}[Q_L - Q]\|_V^2,$$

where the first term is interpreted as the stochastic error or total variance of the estimator and the second term as the numerical error or bias term.

One may now want to equilibrate those two parts. This means that for any given root MSE tolerance ε , we want to choose the number of refinement levels L and number of samples

N_ℓ such that both error contributions are bounded by $\varepsilon^2/2$. Note, however, that splitting the error equally is neither necessary nor optimal [25] and is therefore only a simplification.

Let C_ℓ denote the cost of computing a single sample $\Delta Q_\ell^{(i,\ell)}$. As a consequence, the total cost for computing the MLMC estimator is then given by the sum

$$(2.6) \quad \mathcal{C} \left[\widehat{Q}_h^{\text{ML}} \right] = \sum_{\ell=0}^L N_\ell C_\ell.$$

If we assume L and the overall cost to be fixed, the optimal number of samples N_ℓ is determined by minimizing the total variance $\sum_{\ell=0}^L N_\ell^{-1} V_\ell$, which yields the lower bound

$$(2.7) \quad N_\ell \geq \frac{2}{\varepsilon^2} \sqrt{\frac{V_\ell}{C_\ell}} \sum_{\ell'=0}^L \sqrt{V_{\ell'} C_{\ell'}}.$$

When estimates for the numerical part of the error (2.5) are explicitly known a priori, it is possible to derive theoretically optimal choices for L [6]. In general, however, those constants are not known a priori. Therefore, we instead opt for the approach as in [9], which chooses L ‘‘on-the-fly’’.

Algorithm 2.1 Multilevel Monte-Carlo

- 1: Initialize $L = 2$ and set initial values for N_ℓ on levels $\ell = 0, 1, 2$.
 - 2: **while** N_ℓ was increased previously in [Line 1](#), [Line 5](#) or [Line 7](#) for any ℓ **do**
 - 3: Compute remaining $Q_\ell(w^{(i,\ell)})$ and $Q_{\ell-1}(w^{(i,\ell)})$ for $i = 1, \dots, N_\ell$ on each level ℓ .
 - 4: Compute $\widehat{Q}_h^{\text{ML}}$ according to (2.3) and update estimates for V_ℓ , $\ell = 0, 1, \dots, L$.
 - 5: Update N_ℓ for $\ell = 0, 1, \dots, L$ according to (2.7) using the new estimates for V_ℓ .
 - 6: **if** test for convergence of the bias term fails, **then**
 - 7: set $L := L + 1$ and initialize N_L .
 - 8: **end if**
 - 9: **end while**
-

In [Line 4](#) in the above algorithm, the variances V_ℓ are estimated according to (2.4) by approximating,

$$(2.8) \quad V_\ell = \mathbb{E} \left[\|\Delta Q_\ell\|_V^2 \right] - \|\mathbb{E}[\Delta Q_\ell]\|_V^2 \approx \frac{1}{N_\ell - 1} \left(\sum_{i=1}^{N_\ell} \|\Delta Q_\ell^{(i,\ell)}\|_V^2 - \frac{1}{N_\ell} \left\| \sum_{i=1}^{N_\ell} \Delta Q_\ell^{(i,\ell)} \right\|_V^2 \right).$$

In [Line 7](#), we test for convergence by verifying, if

$$\|\mathbb{E}[Q_L - Q]\|_V^2 < \varepsilon^2/2$$

is satisfied for the root MSE tolerance ε . Since $\mathbb{E}[Q]$ depends on the (a priori unknown) exact solution u , we approximate the remaining error from previous levels. If we assume that $\|\mathbb{E}[Q_\ell - Q_{\ell-1}]\|_V^2 = \mathcal{O}(2^{-\alpha\ell})$ for $\ell \rightarrow \infty$ with $\alpha \geq 1$, we obtain

$$(2.9) \quad \mathbb{E}[Q - Q_L] = \sum_{\ell=L+1}^{\infty} \mathbb{E}[Q_\ell - Q_{\ell-1}] \simeq \frac{\mathbb{E}[Q_L - Q_{L-1}]}{2^\alpha - 1},$$

where we use the symbol “ \simeq ” in the following sense:

$$A \simeq B \iff cB \leq A \leq \hat{c}B, \quad c, \hat{c} > 0.$$

The total cost for computing the MLMC estimate $\widehat{Q}_h^{\text{ML}}$ in a Hilbert space is characterized by the following theorem, similar to [9, Theorem 1].

Theorem 2.1. *Suppose there exist constants $\alpha, \beta, \gamma > 0$, such that $\alpha \geq \frac{1}{2} \min(\beta, \gamma)$,*

- $\|\mathbb{E}[Q_h - Q]\|_V \leq \mathcal{O}(h^\alpha)$ as $h \rightarrow 0$,
- $V_\ell \leq \mathcal{O}(H_\ell^\beta)$ and
- $C_\ell \leq \mathcal{O}(H_\ell^{-\gamma})$ as $\ell \rightarrow \infty$.

Then for any ε small enough there exist a total number of levels L and a number of samples N_ℓ , $\ell = 0, \dots, L$, such that the root mean square error $e(\widehat{Q}_h^{\text{ML}})$ is bounded by ε and the total cost behaves like

$$(2.10) \quad c \left[\widehat{Q}_h^{\text{ML}} \right] \leq \mathcal{O} \left(\begin{cases} \varepsilon^{-2}, & \beta > \gamma \\ \varepsilon^{-2} (\log \varepsilon)^2, & \beta = \gamma \\ \varepsilon^{-2 - \frac{\gamma - \beta}{\alpha}}, & \beta < \gamma \end{cases} \right).$$

Since the proof of the above theorem closely follows along the lines of [9, Appendix A], it is omitted here. The main difference results from assuming that $\mathbb{E}[Q]$ is an element of a generic Hilbert space and from the corresponding definitions of the estimator’s total variance and numerical bias according to (2.4) and (2.5).

To interpret the above theorem, it is useful to have a look at the core idea of its proof. Starting with the basic definition of the costs (2.6), we insert the optimal choice for the number of samples (2.7) and the assumptions on V_ℓ and C_ℓ ,

$$c \left[\widehat{Q}_h^{\text{ML}} \right] = \sum_{\ell=0}^L N_\ell C_\ell \simeq \frac{2}{\varepsilon^2} \left(\sum_{\ell=0}^L \sqrt{V_\ell C_\ell} \right)^2 \simeq \varepsilon^{-2} \left(\sum_{\ell=0}^L H_\ell^{\frac{\beta - \gamma}{2}} \right)^2.$$

A case-by-case analysis of the sum then leads to the estimate in (2.10). Note that the case $\beta > \gamma$ means that the total cost is dominated by the coarsest levels, whereas $\beta < \gamma$ corresponds to a case where most of the computational effort is found on the finest levels.

3. MLMC and local time-stepping for the wave equation. Here, we estimate the computational cost of the MLMC algorithm in the presence of locally refined meshes, first with a standard explicit time-stepping method and then with a local time-stepping method. In doing so, we differentiate between local refinement inside a small fixed region or towards a reentrant corner.

3.1. Standard discretizations on a (quasi-)uniform mesh. In Algorithm 2.1, the numerical method to compute any approximation for a fixed sample $\omega \in \Omega$ was not specified further. In this section, we will take a closer look on the methods used to compute numerical solutions to (2.1) for fixed $\omega \in \Omega$. We start by discretizing the wave equation (2.1) in space with either

standard continuous (H^1 -conforming) finite elements with mass-lumping or an appropriate discontinuous Galerkin (DG) discretization, for example symmetric IP-DG [23] or HDG [44]. The standard continuous Galerkin formulation of the wave equation (2.1) starts from its weak formulation [30]. We then wish to approximate the solution $u(t, \cdot)$ in a suitable Finite Element space V_H and thus consider the semidiscrete Galerkin approximation: find $u_H : [0, T] \rightarrow V_H$ such that

$$\begin{aligned} \left(\frac{\partial^2}{\partial t^2} u_H, v \right) + (c \nabla u_H, c \nabla v) &= (f, v) & \forall v \in V_H, t \in (0, T], \\ u_H|_{t=0} &= \Pi_H u_0, \\ \frac{\partial}{\partial t} u_H|_{t=0} &= \Pi_H v_0, \end{aligned}$$

where (\cdot, \cdot) and Π_H denote the standard L^2 scalar product and the L^2 -projection onto V_H , respectively. Furthermore, let \mathbf{u} denote the vector of coefficients of u_H with respect to a basis $(\varphi_i)_{i=1, \dots, n}$ of V_H and let the mass and stiffness matrices, \mathbf{M} and \mathbf{K} , together with the load vector \mathbf{F} be defined as

$$M_{i,j} = (\varphi_j, \varphi_i), \quad K_{i,j} = (c \nabla \varphi_j, c \nabla \varphi_i), \quad F_i(t) = (f(t, \cdot), \varphi_i),$$

respectively. Here to compute the entries in the mass matrix \mathbf{M} , we use judicious quadrature rules which yield a diagonal matrix while preserving the order of accuracy (aka "mass-lumping") [11, 10]. This leads to a second-order system of ordinary differential equations

$$(3.1) \quad \begin{cases} \mathbf{M} \ddot{\mathbf{u}}(t) + \mathbf{K} \mathbf{u}(t) = \mathbf{F}(t), \\ \mathbf{u}(0) = \mathbf{u}_0, \\ \dot{\mathbf{u}}(0) = \mathbf{v}_0. \end{cases}$$

Since \mathbf{K} depends continuously on c , a random field with values in $L^\infty(D)$, \mathbf{u} also depends continuously on c and is therefore measurable with respect to Ω . Setting $\mathbf{z} = \mathbf{M}^{1/2} \mathbf{u}$, $\mathbf{A} = \mathbf{M}^{-1/2} \mathbf{K} \mathbf{M}^{-1/2}$ and $\tilde{\mathbf{F}} = \mathbf{M}^{-1/2} \mathbf{F}$, we rewrite (3.1) as

$$(3.2) \quad \ddot{\mathbf{z}}(t) + \mathbf{A} \mathbf{z}(t) = \tilde{\mathbf{F}}(t),$$

which can now be discretized in time by a standard explicit time-stepping scheme.

For $\tilde{\mathbf{F}}_n = \tilde{\mathbf{F}}(t_n)$, the second-order leapfrog (LF) scheme with time step $\Delta t > 0$ is then given by

$$(3.3) \quad \begin{cases} \mathbf{z}_{n+1} - 2\mathbf{z}_n + \mathbf{z}_{n-1} = \Delta t^2 (\tilde{\mathbf{F}}_n - \mathbf{A} \mathbf{z}_n) & \forall n \geq 1, \\ \mathbf{z}_0 = \mathbf{M}^{1/2} \mathbf{u}_0, \\ \mathbf{z}_1 = \mathbf{z}_0 - \Delta t \mathbf{M}^{1/2} \mathbf{v}_0 + \frac{\Delta t^2}{2} (\tilde{\mathbf{F}}_0 - \mathbf{A} \mathbf{z}_0), \end{cases}$$

where $\mathbf{z}_n \simeq \mathbf{z}(t_n, \omega)$ for a fixed $\omega \in \Omega$.

We now apply Theorem 2.1 on the computational complexity of MLMC methods to the above discrete Galerkin formulation of the stochastic wave equation. Although in (3.3) we

opt for the popular second-order leapfrog scheme [24], all the estimates derived below in fact remain identical for any standard explicit time-stepping method, such as Runge-Kutta or Adams-Bashforth methods.

Corollary 3.1. *Let $\widehat{Q}_h^{\text{ML}}$ be the MLMC estimator of $\mathbb{E}[Q] \in V$, where $Q : C^0(0, T; L^2(D)) \rightarrow V$ is Lipschitz continuous, $D \subset \mathbb{R}^d$. Assume $u(\cdot, \cdot, \omega)$ to be sufficiently regular uniformly in $\omega \in \Omega$ and let $Q_\ell = Q(u_{H_\ell})$, where u_{H_ℓ} is computed using a finite element space discretization of order k and explicit time integration of order m , such that*

$$(3.4) \quad \|\mathbb{E}[Q_\ell - Q]\|_V \leq \mathcal{O}\left((H_\ell)^{k+1} + (\Delta t_\ell)^m\right),$$

where Δt_ℓ satisfies a CFL stability condition $\Delta t_\ell \simeq H_\ell/k^2$. Furthermore let $\beta > 0$ be a constant such that $V_\ell \leq \mathcal{O}\left((H_\ell)^\beta\right)$ and

$$2 \min\{k + 1, m\} \geq \min\{\beta, d + 1\},$$

and assume that the cost for evaluating the random field c at the quadrature nodes is bounded by $\mathcal{O}\left(k^{2d} H_\ell^{-d}\right)$.

Then for $\varepsilon > 0$ sufficiently small, there exist a total number of levels L and a number of samples N_ℓ , $\ell = 0, \dots, L$, such that the root mean square error $e(\widehat{Q}_h^{\text{ML}})$ is bounded by ε and the total cost behaves like

$$(3.5) \quad \mathcal{C}\left[\widehat{Q}_h^{\text{ML}}\right] \leq \mathcal{O}\left(\begin{cases} \varepsilon^{-2}, & \beta > d + 1 \\ \varepsilon^{-2}(\log \varepsilon)^2, & \beta = d + 1 \\ \varepsilon^{-2 - \frac{d+1-\beta}{\min\{m, k+1\}}}, & \beta < d + 1 \end{cases}\right).$$

Remark 3.2. For Q Lipschitz continuous, it is reasonable to assume for sufficiently regular solutions that the weak convergence rates assumed in (3.4) are in fact a consequence of the other assumptions. The above result can be generalized to rougher solutions $u(\cdot, \cdot, \omega)$ or a broader class of quantities of interest Q , with possibly modified convergence rates in (3.5).

Proof. The computational cost C_ℓ of solving one wave equation with an explicit time-stepping scheme on a (quasi-)uniform mesh of size H_ℓ is computed as the number of time-steps times the costs in each step, which are dominated by one or more matrix-vector multiplications of type “ $\mathbf{A}\mathbf{u}_n$ ”. The cost of each such matrix-vector multiplication is approximately the number of degrees of freedom per element squared, k^{2d} , times the number of elements, which is proportional to H_ℓ^{-d} . Due to the CFL stability condition, the number of time-steps used on each level is inversely proportional to $\Delta t_\ell \simeq H_\ell/k^2$. Hence, the total computational cost for solving one wave equation on level ℓ is $C_\ell = c_3 k^{2(d+1)} H_\ell^{-(d+1)}$, as by assumption the cost for evaluating the random field at the quadrature nodes is negligible. Since $\alpha = \min\{k + 1, m\}$ because of the CFL restriction on Δt_ℓ , the corollary directly follows from Theorem 2.1. ■

3.2. Effect of local mesh refinement. Due to geometric constraints or accuracy requirements, it may not be optimal or even possible to coarsen the entire mesh uniformly in the presence of singularities or complex geometry, see Figure 3.1. This results in a splitting of the

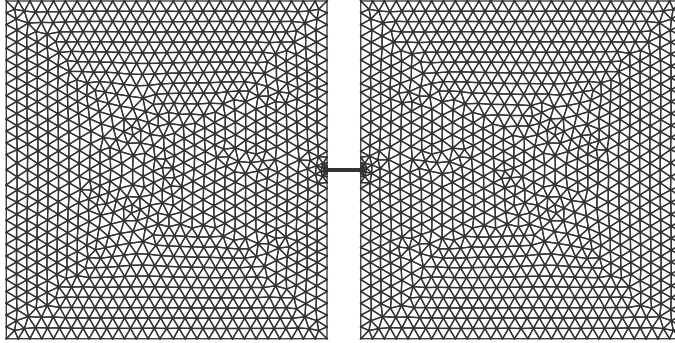


Figure 3.1. 2D Example of a computational domain with a locally refined mesh.

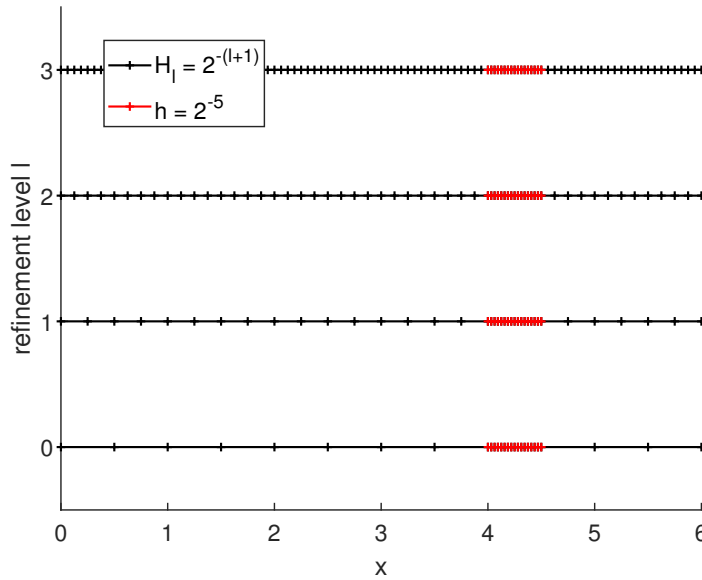


Figure 3.2. Locally refined sequence of meshes on $[0, 6]$ for levels $\ell = 0, 1, 2, 3$. In the coarse part the mesh width is halved on each level, $h_\ell^c = 1/2^{\ell+1}$. In the fine part the width h^f stays constant at $1/32$.

computational domain, where small parts of the geometry D_{fine} might not allow for elements larger than a given h^f , and a coarse part D_{coarse} , where the mesh size on the coarsest level $H_0 \gg h^f$ can be much larger, while still resolving the dominant wave-length in the problem. An example is illustrated in [Figure 3.2](#) for the domain $D = (0, 6) \subset \mathbb{R}$.

One of the key ideas of the MLMC method is to evaluate most of the samples on the coarsest levels and only a few on the finest. For explicit time-stepping schemes the maximal time-step depends on the size of the smallest elements of the mesh. Hence, if parts of the mesh

consist of a few tiny elements across all levels, for every sampling the maximal time-step will be constrained by those elements in D_{fine} . On coarser levels, standard explicit time-stepping schemes then become increasingly inefficient due to the ever more restrictive CFL condition.

In order to quantify this effect, we want to estimate the computational cost of multilevel Monte Carlo with finite elements and leapfrog for the wave equation in the presence of local refinement on a fixed hierarchy of discretizations. Note that another type of local mesh refinement will be addressed in section 3.5, which are graded meshes for domains with a reentrant corner.

For the sake of simplicity, we consider (3.1) with right-hand side equal to zero and assume that the (integer) coarse-to-fine mesh size ratio $p_\ell = \lceil H_\ell/h^\ell \rceil$ on any level $\ell = 0, 1, \dots, L$ is of the form $p_\ell = p_0/2^\ell = 2^{L-\ell}p_L$, where we assume that $p_L \geq 1$ and, thus, $H_L \geq h_{\min}$, i.e. the mesh in D_{coarse} with mesh size $H_\ell = H_0/2^\ell$ remains coarser than the mesh in D_{fine} through all levels (except possibly the finest). In the following, let $\Delta t_\ell \simeq H_\ell/k^2$ be the ‘‘optimal’’ time-step on a uniform mesh, where k denotes the polynomial degree of the FEM basis functions. Then, the number of FEM degrees of freedom in the coarse and fine parts are respectively given by

$$(3.6) \quad n_\ell^c \simeq (1-r)k^d \frac{1}{(H_\ell)^d}, \quad n_\ell^f \simeq rk^d \frac{1}{(h^\ell)^d} = rk^d \left(\frac{p_\ell}{H_\ell} \right)^d \quad \forall \ell,$$

where r denotes the relative volume of the locally refined part D_{fine} with respect to the whole domain D . As the computational cost of the leapfrog method (3.3) is dominated by matrix-vector multiplications $\mathbf{A}\mathbf{u}_n$, we will focus on these. With (3.6), each of these require approximately

$$k^d (n_\ell^c + n_\ell^f) \simeq \frac{k^{2d}}{H_\ell^d} \left((1-r) + rp_\ell^d \right)$$

operations. Due to CFL restriction, in order to proceed the simulation from 0 to T , standard LF needs approximately

$$\frac{T}{\Delta t_\ell/p_\ell} \simeq T \frac{k^2 p_\ell}{H_\ell}$$

time-steps of size $\Delta t_\ell/p_\ell \simeq h_{\min}/k^2 \forall \ell$. Hence, ignoring constants, the cost of solving *one* wave equation on any level ℓ for a particular $\omega \in \Omega$ approximately equals

$$(3.7) \quad \mathcal{C}_{\text{LF}}[u_\ell(\omega)] \simeq \left(T \frac{k^2 p_\ell}{H_\ell} \right) \cdot \left(\frac{k^{2d}}{H_\ell^d} \left((1-r) + rp_\ell^d \right) \right) = T k^{2(d+1)} \frac{p_\ell}{H_\ell^{d+1}} \left((1-r) + rp_\ell^d \right).$$

Thus, we can estimate the cost of computing *one* sample $\Delta Q_\ell^{(i,\ell)}$ using standard LF, where we assume that the cost of computing any $Q_\ell^{(i,\ell)}$ is dominated by computing $u_\ell^{(i,\ell)}$. Again, we also assume that the cost for evaluating the random field c at the quadrature nodes is bounded by $\mathcal{O}(k^{2d}H_\ell^{-d})$ and thus negligible with respect to the overall cost for computing the numerical solution. For $\ell = 0$, we have

$$C_0^{\text{LF}} = \mathcal{C}_{\text{LF}} \left[u_0 \left(\omega^{(i,0)} \right) \right] \simeq \frac{T k^{2(d+1)}}{H_0^{d+1}} \left((1-r) p_0 + r p_0^{d+1} \right)$$

and for $\ell \geq 1$, where we use that $H_\ell = H_0/2^\ell$ and $p_\ell = 2^{-\ell}p_0$,

$$\begin{aligned}
 C_\ell^{\text{LF}} &= \mathcal{C}_{\text{LF}} \left[u_\ell \left(\omega^{(i,\ell)} \right) \right] + \mathcal{C}_{\text{LF}} \left[u_{\ell-1} \left(\omega^{(i,\ell)} \right) \right] \\
 &\simeq T k^{2(d+1)} \left(\frac{p_\ell}{H_\ell^{d+1}} \left((1-r) + r p_\ell^d \right) + \frac{p_{\ell-1}}{H_{\ell-1}^{d+1}} \left((1-r) + r p_{\ell-1}^d \right) \right) \\
 (3.8) \quad &\simeq \frac{T k^{2(d+1)}}{H_0^{d+1}} \left((1-r) \frac{2^d + 1}{2^d} 2^{d\ell} p_0 + r \cdot 2 p_0^{d+1} \right).
 \end{aligned}$$

By using (2.7), we obtain for the estimate of the total computational cost:

$$\begin{aligned}
 \mathcal{C}_{\text{LF}} &= \sum_{\ell=0}^L N_\ell C_\ell^{\text{LF}} \simeq \frac{2}{\varepsilon^2} \left(\sum_{\ell=0}^L \sqrt{V_\ell C_\ell^{\text{LF}}} \right)^2 \\
 &\simeq \frac{2 T k^{2(d+1)}}{\varepsilon^2 H_0^{d+1}} \left(\sqrt{\left((1-r) p_0 + r p_0^{d+1} \right) V_0} \right. \\
 (3.9) \quad &\quad \left. + \sum_{\ell=1}^L \sqrt{\left((1-r) \frac{2^d + 1}{2^d} 2^{d\ell} p_0 + r \cdot 2 p_0^{d+1} \right) V_\ell} \right)^2.
 \end{aligned}$$

3.3. Local time-stepping. To overcome the bottleneck due to local mesh refinement on explicit time-stepping methods, we now consider explicit local time-stepping (LTS) methods [12] for the numerical solution of (3.2). First, we split the vector of unknowns \mathbf{z} into coarse and fine parts as

$$\mathbf{z} = (\mathbf{I} - \mathbf{P})\mathbf{z} + \mathbf{P}\mathbf{z},$$

where \mathbf{P} is a diagonal matrix with all entries equal to zero or one, identifying the degrees of freedom in the refined part of the mesh D_{fine} and all elements adjacent to it. Hence, $\mathbf{P}\mathbf{z}$ contains those degrees of freedom associated with the locally refined part of the mesh.

The original leapfrog-based local time-stepping (LF-LTS) methods for solving second-order wave equations (3.2) with arbitrarily high accuracy was proposed in [12, 21]. Inside the “coarse” part of the mesh, it uses the standard LF method with a global time-step Δt . Inside the “fine” part, however, the method loops over p local LF steps of size $\Delta t/p$, where $p \geq H/h^f$ is a positive integer. When combined with a mass-lumped conforming [11, 10] or discontinuous Galerkin FE discretization [23] in space, the resulting method is truly explicit and inherently parallel; it was successfully applied to 3D seismic wave propagation [31]. A multilevel version was later proposed [13] and achieved high parallel efficiency on an HPC architecture [41].

Optimal convergence rates for the LF-LTS method from [12] were derived for a conforming FEM discretization, albeit under a CFL condition where Δt in fact depends on the smallest elements in the mesh [19]. To prove optimal L^2 convergence rates under a CFL condition independent of p , a stabilized version of LF-LTS was devised recently in [20]. The same algorithm was also proposed independently by the group of Hochbruck [8]. For this method, we consider stabilized Chebyshev polynomials [29], based on Chebyshev polynomials of the

first kind [42], denoted by T_p , and a stabilization parameter $0 \leq \nu \leq 1$. Let us further define the constants

$$\delta_{p,\nu} := 1 + \frac{\nu}{p^2}, \quad \omega_{p,\nu} := 2 \frac{T'_p(\delta_{p,\nu})}{T_p(\delta_{p,\nu})}$$

and

$$\beta_k := \frac{T_{k-1}(\delta_{p,\nu})}{T_{k+1}(\delta_{p,\nu})}, \quad \beta_{k+1/2} := \frac{T_k(\delta_{p,\nu})}{T_{k+1}(\delta_{p,\nu})} \quad \forall 1 \leq k \leq p-1.$$

For example, for $p = 2$, $\nu = 0.01$, we have $\delta_{p,\nu} = 1.005$ and

$$\begin{aligned} \omega_{p,\nu} &= 2 \frac{T'_2(\delta_{p,\nu})}{T_2(\delta_{p,\nu})} = 2 \frac{4\delta_{p,\nu}}{2\delta_{p,\nu}^2 - 1} \approx 7.882, \\ \beta_1 &= \frac{T_0(\delta_{p,\nu})}{T_2(\delta_{p,\nu})} = \frac{1}{2\delta_{p,\nu}^2 - 1} \approx 0.981, \\ \beta_{3/2} &= \frac{T_1(\delta_{p,\nu})}{T_2(\delta_{p,\nu})} = \frac{\delta_{p,\nu}}{2\delta_{p,\nu}^2 - 1} \approx 0.985. \end{aligned}$$

Then, the stabilized LF-LTS algorithm to compute $\mathbf{z}_{n+1} \simeq \mathbf{z}(t_{n+1})$ for given $\mathbf{z}_n, \mathbf{z}_{n-1}$ for the wave equation, here with zero forcing for simplicity, is given as follows.

Algorithm 3.1 Stabilized LF-LTS

- 1: Set $\mathbf{q}_0^n := \mathbf{z}_n$ and $\mathbf{w}_n = \mathbf{A}(\mathbf{I} - \mathbf{P})\mathbf{q}_0^n$.
- 2: Compute

$$\mathbf{q}_{1/p}^n = \mathbf{q}_0^n - \frac{1}{2} \left(\frac{\Delta t}{p} \right)^2 \frac{2p^2}{\omega_{p,\nu}\delta_{p,\nu}} (\mathbf{w}_n + \mathbf{A}\mathbf{P}\mathbf{q}_0^n).$$

- 3: **for** $m = 1, \dots, p-1$ **do**
- 4: Compute

$$\mathbf{q}_{(m+1)/p}^n = (1 + \beta_k) \mathbf{q}_{m/p}^n - \beta_k \mathbf{q}_{(m-1)/p}^n - \left(\frac{\Delta t}{p} \right)^2 \frac{2p^2}{\omega_{p,\nu}} \beta_{k+1/2} (\mathbf{w}_n + \mathbf{A}\mathbf{P}\mathbf{q}_{m/p}^n).$$

- 5: **end for**
 - 6: Compute $\mathbf{z}_{n+1} = -\mathbf{z}_{n-1} + 2\mathbf{q}_1^n$.
-

If the fraction of nonzero entries in \mathbf{P} is small, the overall cost will be dominated by the computation of \mathbf{w}_n , which requires a single multiplication with $\mathbf{A}(\mathbf{I} - \mathbf{P})$ per time-step Δt . All further matrix-vector multiplications with $\mathbf{A}\mathbf{P}$ only involve those unknowns inside, or immediately next to, the refined region. Inside D_{coarse} , away from the coarse/fine interface, the algorithm reduces to the standard LF method with time-step Δt , regardless of p or ν . This is especially the case for $\mathbf{P} = 0$, that is without any local time-stepping. For $\nu = 0$, [Algorithm 3.1](#) coincides with the original one from [12], since $\delta_{p,\nu} = 1$, and therefore $\beta_k = \beta_{k+1/2} = 1$ and $\omega_{p,\nu} = 2p^2$.

In the following, we wish to repeat the computational cost analysis for MLMC combined with standard LF on locally refined meshes from the previous section, but this time for the

(stabilized) LF-LTS method. On any level ℓ , solving a single wave equation will require approximately

$$\frac{T}{\Delta t_\ell} \simeq \frac{Tk^2}{H_\ell}$$

time-steps of size Δt_ℓ . For each of these time-steps, the computational cost is dominated by $p_\ell \simeq H_\ell/h^f$ operations of the type “ $\mathbf{A}\mathbf{P}\mathbf{v}$ ” and one operation “ $\mathbf{A}(\mathbf{I} - \mathbf{P})\mathbf{v}$ ”, which only affect the n^f or n_ℓ^c unknowns in the fine or coarse part of the domain, respectively. While again ignoring constants, it follows from (3.6) that the cost of solving *one* wave equation on level ℓ for a particular $\omega \in \Omega$ with LF-LTS approximately equals

$$(3.10) \quad \mathcal{C}_{\text{LF-LTS}}[u_\ell(\omega)] \simeq \frac{Tk^2}{H_\ell} \cdot \left(\frac{k^{2d}}{H_\ell^d} \left((1-r) + rp_\ell^{d+1} \right) \right) = \frac{Tk^{2(d+1)}}{H_\ell^{d+1}} \left((1-r) + rp_\ell^{d+1} \right).$$

As before, this allows us to estimate the cost of computing *one* sample $\Delta Q_\ell^{(i,\ell)}$ using LF based LTS for any ℓ , where, again, we assume that the cost for generating the random field c in the quadrature nodes is bounded by $\mathcal{O}(k^{2d}H_\ell^{-d})$ and thus negligible with respect to the overall costs for computing the solution. For $\ell = 0$, this results simply from inserting $\ell = 0$ in (3.10). For $\ell \geq 1$, we receive by similar arguments as before,

$$(3.11) \quad \begin{aligned} \mathcal{C}_\ell^{\text{LF-LTS}} &= \mathcal{C}_{\text{LF-LTS}} \left[u_\ell \left(\omega^{(i,\ell)} \right) \right] + \mathcal{C}_{\text{LF-LTS}} \left[u_{\ell-1} \left(\omega^{(i,\ell)} \right) \right] \\ &\simeq \frac{Tk^{2(d+1)}}{H_0^{d+1}} \left((1-r) \frac{2^{d+1} + 1}{2^{d+1}} 2^{(d+1)\ell} + r \cdot 2p_0^{d+1} \right). \end{aligned}$$

With (2.7), this leads to a total computational cost estimate of

$$(3.12) \quad \begin{aligned} \mathcal{C}_{\text{LF-LTS}} &= \sum_{\ell=0}^L N_\ell \mathcal{C}_\ell^{\text{LF-LTS}} \simeq \frac{2}{\varepsilon^2} \left(\sum_{\ell=0}^L \sqrt{V_\ell \mathcal{C}_\ell^{\text{LF-LTS}}} \right)^2, \\ &\simeq \frac{2Tk^{2(d+1)}}{\varepsilon^2 H_0^{d+1}} \left[\sqrt{\left((1-r) + rp_0^{d+1} \right) V_0} \right. \\ &\quad \left. + \sum_{\ell=1}^L \sqrt{\left((1-r) \frac{2^{d+1} + 1}{2^{d+1}} 2^{(d+1)\ell} + r \cdot 2p_0^{d+1} \right) V_\ell} \right]^2. \end{aligned}$$

3.4. Standard vs. local time-stepping: a cost comparison. Here, we estimate the increase in computational cost for MLMC with standard LF time-stepping over the (stabilized) LF-LTS. For this, we compute the theoretical speed-up \mathcal{S} ,

$$(3.13) \quad \mathcal{S}(d, r, p_0, L, \{V_\ell\}) = \frac{\mathcal{C}_{\text{LF}} \left[\widehat{Q}_h^{\text{ML}} \right]}{\mathcal{C}_{\text{LF-LTS}} \left[\widehat{Q}_h^{\text{ML}} \right]},$$

where \mathcal{C}_{LF} denotes the total computational cost for computing the MLMC estimate to the solution of (2.1) with the second-order LF method (3.3) and $\mathcal{C}_{\text{LF-LTS}}$ the cost for computing

it with the LF-LTS scheme (Algorithm 3.1). In particular, we study the effects of various parameters on \mathcal{S} , such as the relative volume r of the locally refined region, D_{fine} , with respect to the entire computational domain D . The quotient of (3.12) over (3.9) yields the following proposition.

Proposition 3.3. *The theoretical speed-up \mathcal{S} in (3.13) of MLMC combined with LF-LTS, $\mathcal{C}_{\text{LF-LTS}}$, over MLMC with standard LF, \mathcal{C}_{LF} , is given by*

$$(3.14) \quad \mathcal{S}(d, r, p_0, L, \{V_\ell\}) = \frac{\mathcal{C}_{\text{LF}}}{\mathcal{C}_{\text{LF-LTS}}} \simeq \left(\frac{\sqrt{\left((1-r)p_0 + rp_0^{d+1} \right) V_0 + \sum_{\ell=1}^L \sqrt{V_\ell \left(r \cdot 2p_0^{d+1} + (1-r) 2^{(d+1)\ell} \cdot \frac{2^{d+1}}{2^{d+\ell}} p_0 \right)}}}{\sqrt{\left((1-r) + rp_0^{d+1} \right) V_0 + \sum_{\ell=1}^L \sqrt{V_\ell \left(r \cdot 2p_0^{d+1} + (1-r) 2^{(d+1)\ell} \cdot \frac{2^{d+1}+1}{2^{d+1}} \right)}}} \right)^2.$$

Remark 3.4. Proposition 3.3 also holds for LTS schemes based on other explicit methods, such as the fourth-order modified equation approach [12], Runge Kutta schemes [18] or Adams-Bashforth methods [22]. Here, for simplicity, we have assumed the variances $\{V_\ell\}_{\ell=0,1,\dots,L}$ to be equal for both time integration methods, with or without LTS; in practice, this may not be true – see also Section 4.3.

The speed-up \mathcal{S} derived in Proposition 3.3 calls for a more detailed interpretation. In doing so, we restrict ourselves to the case where the mesh on the finest level L is (quasi-)uniform. More precisely, we assume that $H_{L-1}/h^f \in (1, 2)$ and refine the mesh in both D_{coarse} and D_{fine} such that the resulting mesh on level L is (quasi-)uniform with $p_L = 1$. Then, the number of MLMC levels is simply given by $L = \lceil \log_2 p_0 \rceil$, so that \mathcal{S} in (3.14) only depends on the four parameters d, r, p_0 and β , where $V_\ell = V_0/2^{\beta\ell}$. Hence, local time-stepping only occurs on the coarser levels $0, 1, \dots, L-1$. Clearly, an even greater speed-up might result from a mesh locally refined even on the finest level L .

In Figures 3.3 to 3.5, the speed-up \mathcal{S} is shown as a function of the single parameters r or p_0 , while keeping all other parameters fixed, as in Table 3.1. We observe that the speed-up rapidly increases with decreasing relative volume of the refined part r , until it reaches the maximal speed-up of LTS over standard LF on the coarsest level $\ell = 0$. For fixed r , the maximal speed-up occurs for $10 \leq p_0 \leq 30$, but decreases again for larger p_0 . At first glance, it might seem counterintuitive that \mathcal{S} decreases for higher values of p_0 . However, as the ratio of degrees of freedom in the “fine” part over those in the “coarse” part further increases with p_0 , the cost for every local time-step relative to the overall cost of one global time-step also increases, which results in LTS being less efficient. In fact, even for a single (deterministic) forward solve, the speed-up of LTS with p local time-steps over standard time integration, given by the ratio of (3.7) over (3.10), is also maximal for the same range of p , as shown on the left of Figure 3.6.

In the right frame of Figure 3.6, we see that the performance of MLMC is improved most by LTS at higher convergence rates β for the variance $V_\ell = V_0/2^{\beta\ell}$. Indeed, the larger β , the more samples are computed on the coarsest levels, where the benefit of using LTS is the greatest.

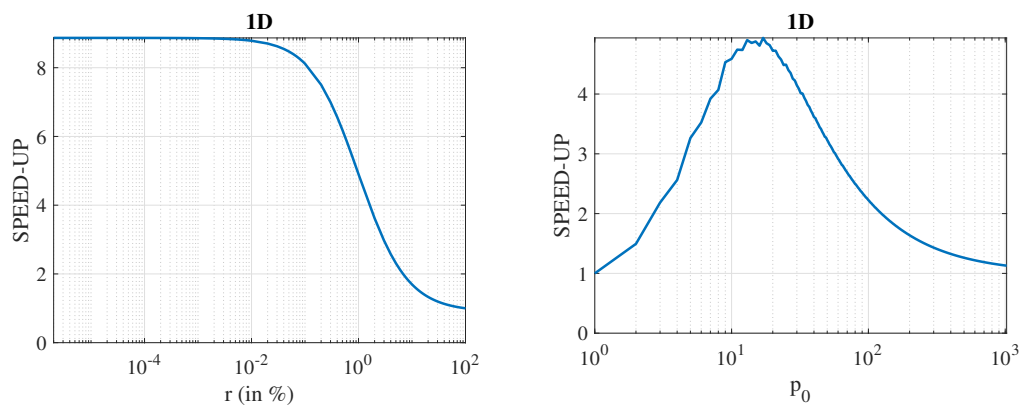


Figure 3.3. Theoretical speed-up \mathcal{S} in 1D vs. relative volume r of refined part (left) and local refinement factor $p_0 = H_0/h^f$ on the coarsest level (right) with other parameters fixed as in Table 3.1.

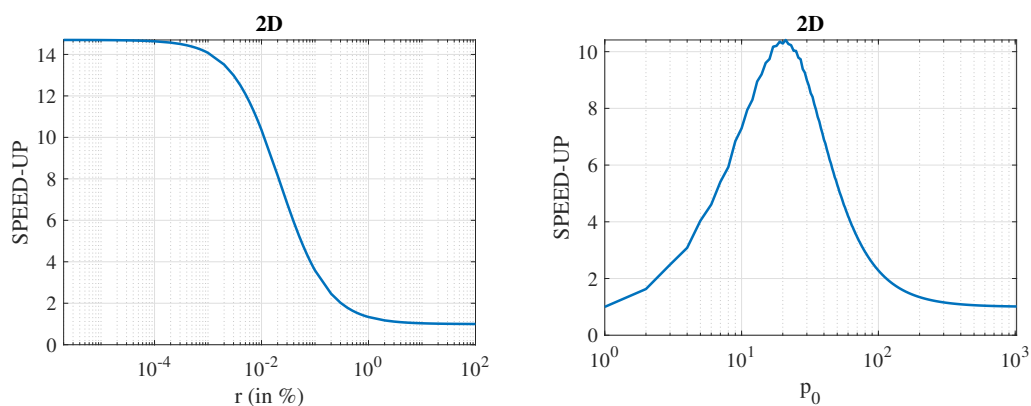


Figure 3.4. Theoretical speed-up \mathcal{S} in 2D vs. relative volume r of refined part (left) and local refinement factor $p_0 = H_0/h^f$ on the coarsest level (right) with other parameters fixed as in Table 3.1.

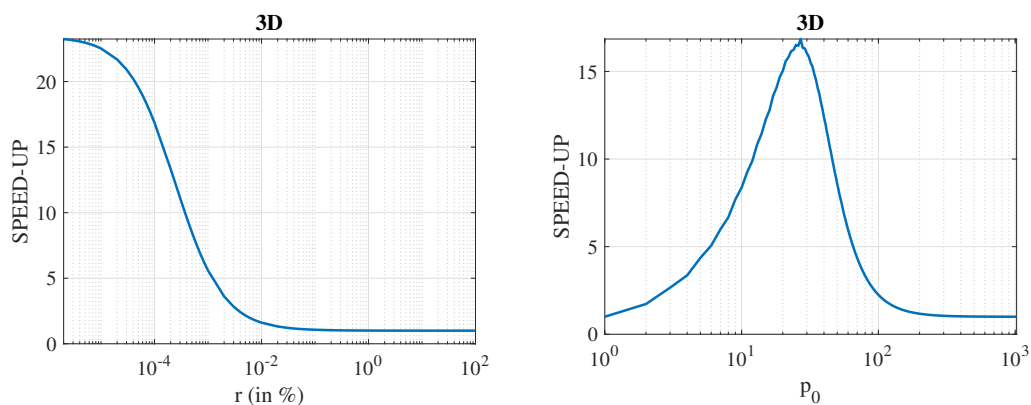


Figure 3.5. Theoretical speed-up \mathcal{S} in 3D vs. relative volume r of refined part (left) and local refinement factor $p_0 = H_0/h^f$ on the coarsest level (right) with other parameters fixed as in Table 3.1.

Table 3.1

Fixed parameter values used in Figures 3.3 to 3.5 with r the relative volume of the refined part, p_0 the local refinement factor on the coarsest level and β the variance convergence rate, i.e. $V_\ell = V_0/2^{\beta\ell}$.

d	r	p_0	β
1	10^{-2}	13	4
2	10^{-4}	19	6
3	10^{-6}	27	8

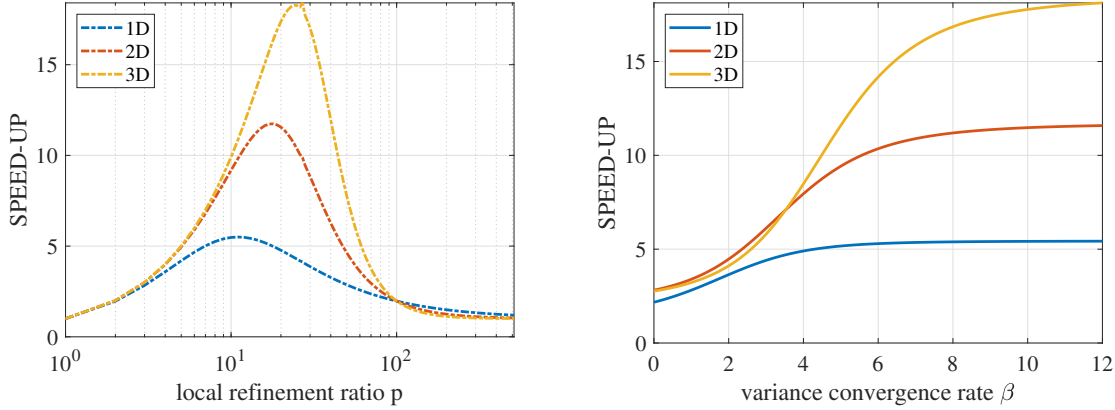


Figure 3.6. Left: Theoretical speed-up for a single (deterministic) solve given by the ratio of (3.7) over (3.10) relative to the number of local time-steps p with relative volume of the refined region $r = 1/100^d$. Right: Theoretical speed-up \mathcal{S} for MLMC vs. variance convergence rate β for $V_\ell = V_0/2^{\beta\ell}$ with local refinement factor p_0 and relative volume r fixed as in Table 3.1.

3.5. Graded mesh refinement towards a reentrant corner. Here, we consider a different type of local mesh refinement due to a graded mesh toward a reentrant corner. Let D be an L-shaped domain in \mathbb{R}^2 with a reentrant corner at $(0.5, 0.5)$, shown in Figure 3.7. Due to characteristic singularities of the solution at reentrant corners, uniform meshes generally do not yield optimal convergence rates [39]. In the elliptic case, a common remedy to restore the accuracy and achieve optimal convergence rates is to use (a-priori) graded meshes toward the reentrant corner with appropriate weighted Sobolev spaces [3], see, e.g. [43, Chapt. 3.3.7]. Optimal convergence rates were also recently proved for a semi-discrete Galerkin formulation of the wave equation [38] on graded meshes.

Hence, we first partition D into six triangles of equal size with a common vertex at the center $(0.5, 0.5)$. Then, on every edge e connected to the center, we allocate $m + 1$ points at distance

$$|e| \left(\frac{k}{m} \right)^s, \quad k = 0, 1, \dots, m,$$

from it, where $s \geq 1$ is a fixed grading parameter; the larger s , the more strongly the triangles will cluster near the reentrant corner, whereas for $s = 1$ the mesh is uniform throughout D . All other vertices within the same k -th layer are distributed uniformly, as shown in Figure 3.7

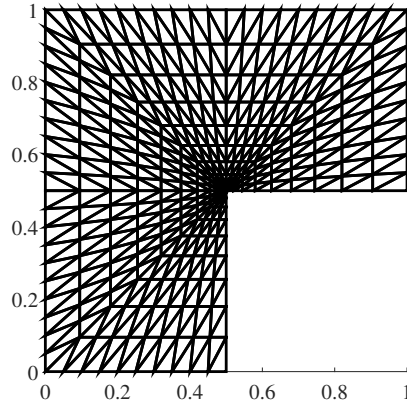


Figure 3.7. Graded mesh on the L-shaped domain D with $m = 10$, $s = 2$.

for a graded mesh with $s = 2$ and $m = 10$. For more details on the construction of these graded meshes, we refer to [38] and the references therein. Furthermore, this strategy can be extended to general dimensions $d = 1, 2, 3$.

By construction, the elements in a graded mesh $\mathcal{T}_{m,s}$ are distributed among m different tiers or layers with equally sized triangles with

$$h_k \approx \frac{k^s - (k-1)^s}{2m^s}, \quad k = 1, \dots, m.$$

Hence, the smallest and largest elements of the mesh are approximately of size $h_1 \approx 1/(2m^s)$ and $h_m \approx (1 - (m-1)^s/m^s)/2$, respectively. The number of elements in the inner k layers scales like k^d .

Thus, we can estimate the computational cost for solving the wave equation for one particular sample $\omega \in \Omega$ with any standard explicit time-stepping method by multiplying the number of time-steps with the number of elements. Since the time-step must be proportional to h_1 for stability, the cost for solving the wave equation on level ℓ in the MLMC algorithm with the standard LF method scales as

$$(3.15) \quad \mathcal{C}_{\text{LF}}[u_\ell(\omega)] \simeq \frac{\text{no. of elements in } D}{h_1} \simeq \frac{m_\ell^d}{h_1} \simeq m_\ell^{s+d},$$

where m_ℓ denotes the number of layers on level ℓ , e.g. $m_\ell = m_0 2^\ell$.

To estimate the computational cost for local time-stepping, we first choose the subdomain D_f , where a smaller time-step is used, as the union of the smaller first q layers with q still to be determined. In D_f , the time-step is then again proportional to the smallest mesh size h_1 , whereas in $D_c = D \setminus D_f$ it is proportional to the smallest element in the outer tiers $q+1, \dots, m_\ell$ of mesh size h_{q+1} . Hence, the computational cost for solving the wave equation

for one particular $\omega \in \Omega$ with an explicit LTS method scales as

$$\begin{aligned}
\mathcal{C}_{\text{LF-LTS}} [u_\ell(\omega)] &\simeq \frac{\text{no. of elements in } D_f}{h_1} + \frac{\text{no. of elements in } D_c}{h_{q+1}} \\
&\simeq q^d m_\ell^s + (m_\ell^d - q^d) \frac{m_\ell^s}{(q+1)^s - q^s} \\
(3.16) \quad &= \frac{m_\ell^{s+d} + m_\ell^s q^d ((q+1)^s - q^s - 1)}{(q+1)^s - q^s}.
\end{aligned}$$

The ratio of (3.15) to (3.16) yields the expected relative speed-up

$$(3.17) \quad \frac{\mathcal{C}_{\text{LF}} [u_\ell(\omega)]}{\mathcal{C}_{\text{LF-LTS}} [u_\ell(\omega)]} \simeq \frac{m_\ell^d ((q+1)^s - q^s)}{m_\ell^d + q^d ((q+1)^s - q^s - 1)}.$$

To determine the optimal value for $q \in [1, m_\ell]$, which minimizes $\mathcal{C}_{\text{LF-LTS}}$, or equivalently maximizes the relative speed-up, we now set the derivative of (3.17) with respect to q to zero:

$$\begin{aligned}
(3.18) \quad &s \left((q+1)^{s-1} - q^{s-1} \right) \left(m_\ell^d + q^d ((q+1)^s - q^s - 1) \right) \\
&- \left((q+1)^s - q^s \right) q^{d-1} \left(d \left((q+1)^s - q^s - 1 \right) + s q \left((q+1)^{s-1} - q^{s-1} \right) \right) = 0.
\end{aligned}$$

Since the left-hand side is positive for small q and negative for larger values such as $q = m_\ell$, there exists an optimal value q_ℓ^{opt} , $1 < q_\ell^{\text{opt}} < m_\ell$, for $m_\ell \geq 2$ and $s \geq 1$. Since we do not seek the precise value of q_ℓ^{opt} but only wish to determine its asymptotic behavior as $m_\ell \rightarrow \infty$, we consider (3.18) for large q . By using that, for any $\alpha > -1$,

$$(3.19) \quad (q+1)^\alpha - q^\alpha = \alpha q^{\alpha-1} + \mathcal{O}(q^{\alpha-2}), \quad q \rightarrow \infty,$$

the equation (3.18) for q_ℓ^{opt} reduces to

$$(s-1) m_\ell^d - s d q^{d+s-1} + \mathcal{O} \left(\frac{m_\ell^d}{q} + q^d + q^{d+s-2} \right) = 0.$$

For $m_\ell, q \rightarrow \infty$, the first two terms clearly dominate the other ones, which yields

$$(3.20) \quad q_\ell^{\text{opt}} = \mathcal{O} \left(m_\ell^{d/(d+s-1)} \right), \quad m_\ell \rightarrow \infty.$$

Applying (3.19) to (3.16) and inserting (3.20) leads to

$$(3.21) \quad \mathcal{C}_{\text{LF-LTS}} [u_\ell(\omega)] \simeq \frac{m_\ell^{d+s}}{s q^{s-1}} + m_\ell^s q^d \left(1 - \frac{1}{s q^{s-1}} \right) = \mathcal{O} \left(m_\ell^{s+d^2/(d+s-1)} \right).$$

Corollary 3.5. *Let $\widehat{Q}_h^{\text{ML}}$ be the MLMC estimator to $\mathbb{E}[Q] \in V$, where Q is a Lipschitz map $C^0(0, T; L^2(D)) \rightarrow V$, $D \in \mathbb{R}^d$. Assume $u(\cdot, \cdot, \omega)$ in a sufficiently regular weighted Sobolev space, uniformly in ω , and let $Q_\ell = Q(u_{H_\ell})$, where u_{H_ℓ} is computed using FE space*

discretization on s -graded meshes, $s \geq 1$, as above and explicit time integration schemes of the same order k , and

$$\|\mathbb{E}[Q_{H_\ell} - Q]\|_V \leq \mathcal{O}\left((H_\ell)^{k+1} + (\Delta t_\ell)^{k+1}\right),$$

where H_ℓ is the largest element of the mesh and the time-step Δt_ℓ fulfills the CFL condition $\Delta t_\ell \simeq H_\ell/k^2$. Further let $\beta > 0$ be a constant, such that $V_\ell \leq \mathcal{O}\left((H_\ell)^\beta\right)$ and $2(k+1) \geq \min\{\beta, s + d^2/(d+s-1)\}$, and assume that the cost for evaluating the random field c at the quadrature nodes is bounded by $\mathcal{O}\left(k^{2d}H_\ell^{-d}\right)$.

Then for any ε small enough, there exist a total number of levels L and a number of samples N_ℓ , $\ell = 0, \dots, L$, such that the root mean square error $e(\widehat{Q}_h^{\text{ML}})$ is bounded by ε .

If standard time-stepping is used, the total cost for $\varepsilon \rightarrow 0$ behaves like

$$(3.22) \quad \mathcal{C}\left[\widehat{Q}_h^{\text{ML}}\right] \leq \mathcal{O}\left(\begin{cases} \varepsilon^{-2}, & \beta > s + d \\ \varepsilon^{-2}(\log \varepsilon)^2, & \beta = s + d \\ \varepsilon^{-2 - \frac{s+d-\beta}{k+1}}, & \beta < s + d \end{cases}\right),$$

whereas if LTS is used, the total cost behaves like

$$(3.23) \quad \mathcal{C}\left[\widehat{Q}_h^{\text{ML}}\right] \leq \mathcal{O}\left(\begin{cases} \varepsilon^{-2}, & \beta > s + d^2/(d+s-1) \\ \varepsilon^{-2}(\log \varepsilon)^2, & \beta = s + d^2/(d+s-1) \\ \varepsilon^{-2 - \frac{s+d^2/(d+s-1)-\beta}{k+1}}, & \beta < s + d^2/(d+s-1) \end{cases}\right).$$

Proof. The result follows from similar arguments as for [Corollary 3.1](#) by applying [Theorem 2.1](#) with $\gamma = s + d$ for standard time-stepping [\(3.15\)](#) and $\gamma = s + d^2/(d+s-1)$ for LTS [\(3.21\)](#). \blacksquare

Remark 3.6. The ratio of [\(3.22\)](#) to [\(3.23\)](#) yields the theoretical speed-up [\(3.13\)](#) for $\varepsilon \rightarrow 0$:

$$(3.24) \quad \mathcal{S} \leq \begin{cases} \mathcal{O}(1), & \beta > s + d, \\ \mathcal{O}\left(\varepsilon^{-\frac{s+d-\beta}{k+1}}\right), & s + d^2/(d+s-1) < \beta < s + d, \\ \mathcal{O}\left(\varepsilon^{-\frac{d(s-1)}{(k+1)(s+d-1)}}\right), & \beta < s + d^2/(d+s-1). \end{cases}$$

For $\beta > s + d$, the total cost is dominated by the computational effort on the coarsest levels for both time-stepping methods; hence, the total cost has the same asymptotic behavior up to a constant factor. In the two other cases with $\beta < s + d$, however, the speed-up of using LF-LTS over standard LF actually grows as the error tolerance ε decreases. In other words, the smaller the desired error level ε , the larger the gain in using local time-stepping in MLMC. For \mathcal{P}^1 -elements in $d = 3$ space dimensions with grading parameter $s = 2$, for instance, the speed-up grows like $\mathcal{S} = \mathcal{O}\left(\varepsilon^{-3/8}\right)$ if $\beta < 17/4$.

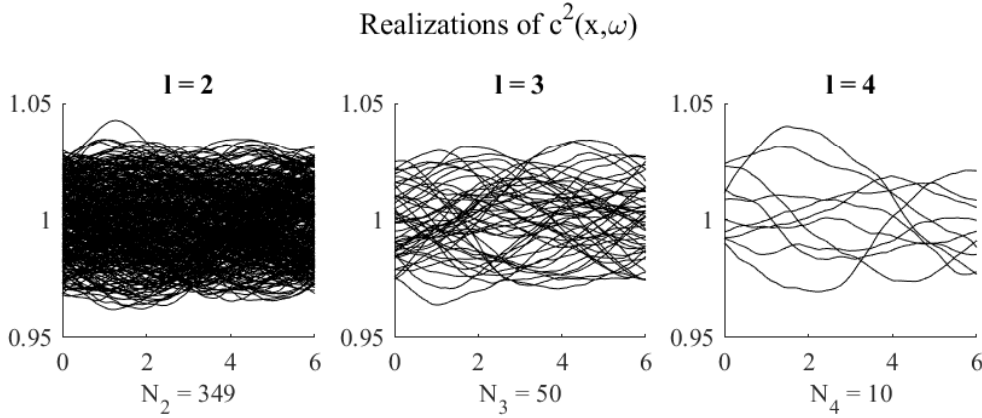


Figure 4.1. Continuous random wave speed. Realizations of the random wave speed $c(x, \omega^{(i, \ell)})$ on levels $\ell = 2, 3, 4$ with $L = 4$.

For the above cost estimation, the computational mesh is divided into a "coarse" and a "fine" region, each associated with a distinct time-step. Since the region of local refinement itself contains sub-regions of further refinement, those "very fine" elements yet again will dictate the time-step, albeit local, to the entire "fine" region. Then, it would be more efficient to let the time-marching strategy mimic the multilevel hierarchy of the mesh organized into tiers of "coarse", "fine", "very fine", etc. elements by introducing a corresponding hierarchy into the time-stepping method. By using within each tier of equally sized elements the corresponding optimal time-step, the resulting multi-level local time-stepping (MLTS) method [13] would achieve an even greater speed-up.

4. Numerical results. To illustrate the efficiency of the combined LF-LTS-MLMC algorithm, we now apply it in three distinct situations with random wave speed.

4.1. Continuous random wave speed. First, we consider (2.1) in $D = (0, 6)$ with $f \equiv 0$, $u_0(x) = e^{-(x-3)^2/0.09}$, $v_0(x) \equiv 0$ and homogeneous Neumann boundary conditions. The random wave speed c^2 is given by the Karhunen-Loève expansion,

$$c^2(x, \omega) = 1 + \sum_{k=1}^{100} \frac{1}{4\pi^2 k^2} \left(\cos\left(\frac{k\pi x}{6}\right) \xi_k^{(1)}(\omega) + \sin\left(\frac{k\pi x}{6}\right) \xi_k^{(2)}(\omega) \right),$$

where $\xi_k^{(1)}, \xi_k^{(2)} \sim U(-1, 1)$ are i.i.d. uniform random variables. In Figure 4.1, we display different realizations of c^2 for different random samples on different levels with $L = 4$.

For discretization in space, we choose continuous, piecewise linear \mathcal{P}^1 finite elements on meshes of size $H_\ell = 2^{-(\ell+4)}$ for $\ell = 0, 1, 2, \dots$. For time discretization, we use the (stabilized) LF-LTS scheme listed in Algorithm 3.1. Here, we arbitrarily set the locally refined part of the mesh to $D_{\text{fine}} = [5 - H_0, 5]$ with elements of size $h^f = 2^{-8}$. In Figure 4.2, the respective FE-solutions $u_\ell(x, T, \omega^{(i)})$ of (2.1) at $T = 11$ on level $\ell = 2$ are shown for 10 particular samples

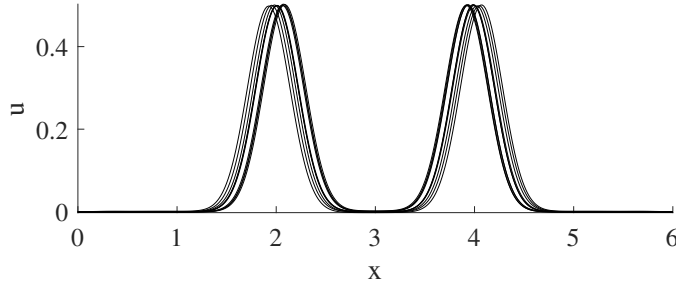


Figure 4.2. Continuous random wave speed. Overlay of solutions $u(x, t, \omega)$ at time $T = 11$ for different random samples of smooth wave speed $c^2(\omega)$.

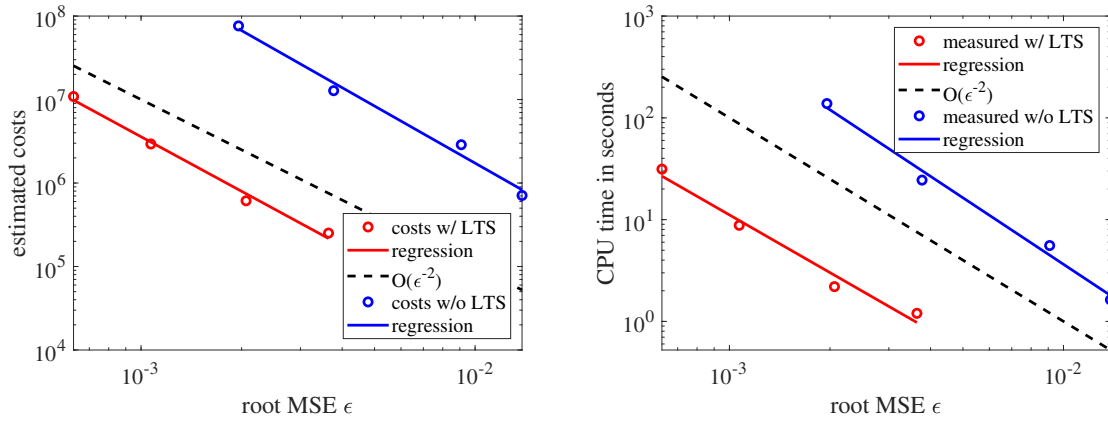


Figure 4.3. Continuous random wave speed. Work to error ratio for the MLMC-FE leapfrog method with, or without, LTS. Left: pre-estimated costs, right: measured CPU times.

$c(x, \omega^{(i)})$.

Next, we apply the MLMC [Algorithm 2.1](#) to estimate the expected value of the mean solution at a fixed time $T > 0$, $\mathbb{E}[u(\cdot, T, \cdot)] \in V = L^2(D)$. In [Figure 4.3](#), we compare the computational cost of MLMC with LF-LTS or with standard LF relative to the root mean square error ε according to [\(2.5\)](#), where the total variance and bias term are estimated using [\(2.8\)](#) and [\(2.9\)](#) with $\alpha = 2$, respectively. Here, the computational cost is either measured via CPU time or estimated using

$$c[\hat{Q}_h^{\text{ML}}] = \sum_{\ell=0}^L N_\ell C_\ell,$$

where N_ℓ and L are determined on the fly by the MLMC algorithm and C_ℓ is estimated from [\(3.7\)](#) for standard LF or [\(3.10\)](#) for LF-LTS. As expected from [Theorem 2.1](#), the costs of both algorithms behave inversely proportional to ε^2 . Nonetheless, the MLMC algorithm using LTS is about one order of magnitude cheaper than that using standard LF.

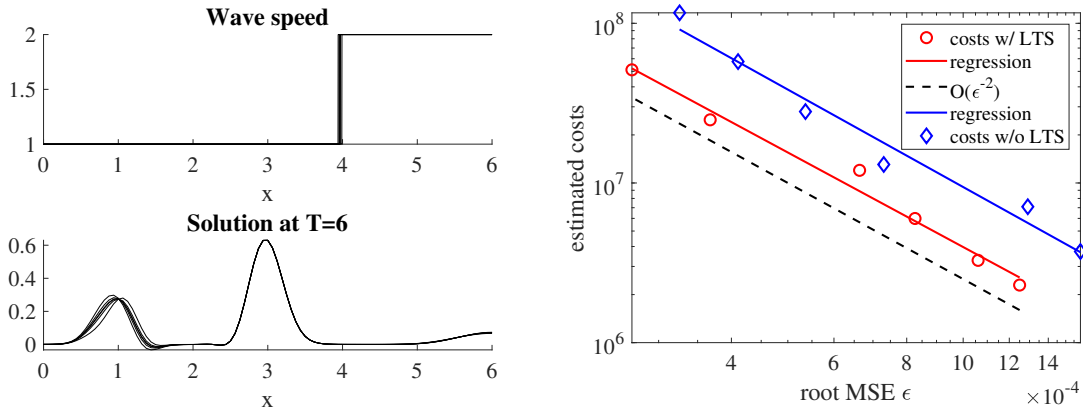


Figure 4.4. Discontinuous random wave speed. Left: Overlay of realizations of the wave speed $c(x, \omega^{(i,0)})$ (top) and FE solutions $u_0(x, T, \omega^{(i,0)})$ at $T = 6$ (bottom). Right: Computational work vs. root MSE tolerance ϵ with, or without, LTS.

4.2. Discontinuous random wave speed. Next, we again consider (2.1) as in the previous example, yet with a discontinuous, piecewise constant wave speed,

$$c(x, \omega) = \begin{cases} 1, & 0 \leq x < \xi(\omega), \\ 2, & \xi(\omega) \leq x \leq 6. \end{cases}$$

Here, the exact jump position is not known precisely and thus modelled as a uniform random variable $\xi \sim U(4 - H_0, 4)$ centered about $x = 4 - H_0/2$, with the mesh size on the coarsest level $H_0 = 1/16$. Inside $[4 - H_0, 4]$, the mesh is locally refined with $h^f = 2^{-9}$, independently of the realization of the random variable. For spatial discretization, we choose continuous \mathcal{P}^1 finite elements. For the time discretization, we use the leap-frog method (3.3), either with or without LTS, as in Algorithm 3.1.

Again, we apply the MLMC Algorithm 2.1, either with or without LTS, to estimate the expected value of the mean solution at a fixed time $T > 0$, $\mathbb{E}[u(\cdot, T, \cdot)] \in V = L^2(D)$. On the left, Figure 4.4 shows an overlay of six particular samples $c(x, \omega^{(i)})$ and the respective FE-solutions $u_\ell(x, T, \omega^{(i)})$ of (2.1) at $T = 6$ on the coarsest level. On the right, we compare the performance of MLMC using either LF-LTS or the standard LF method with respect to the root mean square error ϵ according to (2.5), where again the total variance and bias term are estimated using (2.8) and (2.9) with $\alpha = 2$, respectively. Although the total computational cost behaves inversely proportional to ϵ^2 in both cases, the LF-LTS based MLMC method achieves a significant reduction in overall computational cost.

4.3. Two-dimensional narrow channel. Finally, we consider wave propagation with constant unit speed $c \equiv 1$, vanishing source $f \equiv 0$ and homogeneous Neumann boundary conditions in the two-dimensional domain $D_{b(\omega)}$, shown in Figure 3.1, with varying random width $b(\omega)$ of the narrow channel connecting the two rectangular regions. This leads to the weak

formulation: Find $u : [0, T] \times \Omega \rightarrow H^1(D_{b(\omega)})$ such that

$$(4.1) \quad \frac{\partial^2}{\partial t^2} (u(t, \omega), v) + (\nabla u(t, \omega), \nabla v) = 0 \quad \forall v \in H^1(D_{b(\omega)}).$$

Here, $D_{b(\omega)}$ consists of two 0.95×1 rectangles connected by a narrow channel $0.1 \times b(\omega)$ of varying random width $b(\omega) \in [0.001, 0.007]$, where the origin of the coordinate axes is located at the center of the narrow channel. Next, we reformulate (4.1) as in (2.1) on a (deterministic) reference domain \bar{D} of fixed channel width $\bar{b} = 0.004$, yet with inhomogeneous random velocity $c(x, \omega)$. To do so, we introduce the geometric transformation

$$(4.2) \quad \begin{cases} \Psi(\omega) : & \bar{D} \rightarrow D_{b(\omega)}, \\ & (x, y) \mapsto (x, y + \varphi(x)\psi(y, \omega)), \end{cases}$$

which maps the reference domain \bar{D} to the actual computational domain $D_{b(\omega)}$. Here, φ is smooth and ψ piecewise linear and continuous:

$$\varphi(x) = \begin{cases} 1, & |x| \leq 0.05, \\ 0, & |x| \geq 0.1, \end{cases} \quad \psi(y, \omega) = \begin{cases} 0, & y = 0, \pm 0.5, \\ \pm \left(\frac{b(\omega)}{2} - \frac{\bar{b}}{2} \right), & y = \pm \frac{\bar{b}}{2}. \end{cases}$$

Then, (4.1) is equivalent to

$$(4.3) \quad \frac{\partial^2}{\partial t^2} (|J(\omega)|u, v) + (|J(\omega)|J(\omega)^{-\top} \nabla u, J(\omega)^{-\top} \nabla v) = 0 \quad \forall v \in H^1(\bar{D})$$

with $J(\omega)$ the Jacobian of $\Psi(\omega)$. As initial conditions we choose the Gaussian pulse

$$u_0(\mathbf{x}) = \begin{cases} \exp\left(1 - \frac{R^2}{R^2 - \|\mathbf{x} - \mathbf{x}_0\|^2}\right), & \|\mathbf{x} - \mathbf{x}_0\| < R, \\ 0, & \text{else,} \end{cases} \\ v_0(\mathbf{x}) = 0$$

centered about $\mathbf{x}_0 = (0.5, 0)$ and $R = 0.2$.

For the spatial discretization inside \bar{D} , we use continuous, piecewise linear, finite elements and generate a sequence of triangular meshes independent of the realization $b(\omega)$, with mesh size $H_\ell = 1/60 \cdot 2^{-\ell}$, $\ell = 0, 1, 2, \dots$, outside the channel. Inside the channel we use a mesh size $h^f \approx 7.6 \cdot 10^{-4}$ to resolve the narrow gap geometry. For simplicity in the implementation, we simulate the varying channel width by applying the transformation (4.2) directly to the vertices of the mesh for each realization of the uniform random variable $b \sim U(0.001, 0.007)$. For time discretization, we use the (stabilized) LF-LTS method listed in Algorithm 3.1.

We now apply the MLMC Algorithm 2.1 to estimate as QoI the expected value of the mean solution along the vertical line $x = -0.4$ at time $T = 1$,

$$\mathbb{E}[Q(u)](y) := \mathbb{E} \left[u(\mathbf{x}, T, \cdot) |_{\mathbf{x}=(-0.4, y)} \right] \in V := L^2([-0.5, 0.5]).$$

In Figure 4.5, the FE-LF-LTS solution is shown at time $t = 1$ for a particular sample $b \approx 0.00589$. We observe how the wave initiated on the right crosses the channel whose exit acts

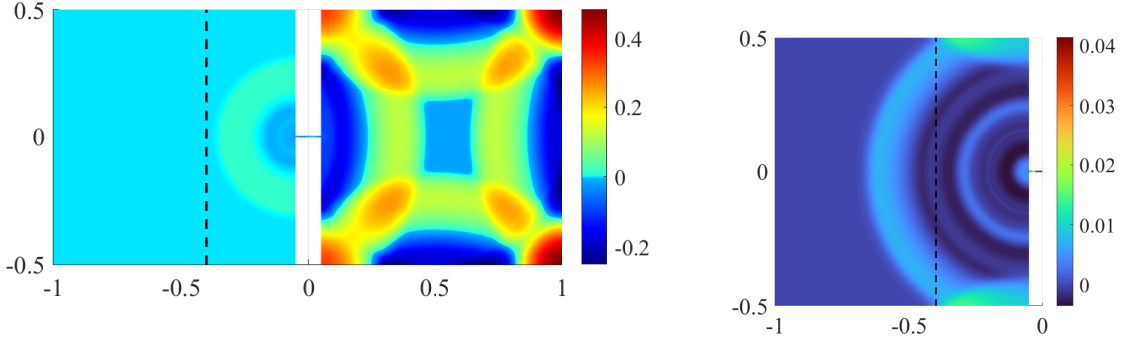


Figure 4.5. Two-dimensional narrow channel. FE solution of the wave equation on D_b for $b \approx 0.00589$ at time $t = 1$ with a dashed line at $x = -0.4$. On the right: The same solution restricted to the left rectangle ($x < 0$) with an adjusted color map.

Table 4.1

Two-dimensional narrow channel. Comparison of using LF-LTS or standard LF to compute MLMC estimate $\widehat{Q}_h^{\text{ML}}$ for RMS error tolerance $\varepsilon = 5 \cdot 10^{-5} \approx 0.020 \cdot \|\widehat{Q}_h^{\text{ML}}\|_{L^2(-0.5, 0.5)}$.

Level	variance V_ℓ		number of samples N_ℓ		costs per sample
	LF-LTS	standard LF	LF-LTS	standard LF	$C_\ell^{\text{LF}}/C_\ell^{\text{LF-LTS}}$
$\ell = 0$	$1.21 \cdot 10^{-6}$	$1.22 \cdot 10^{-6}$	1266	1455	5.89
$\ell = 1$	$6.98 \cdot 10^{-9}$	$2.04 \cdot 10^{-8}$	51	86	7.82
$\ell = 2$	$3.21 \cdot 10^{-10}$	$8.87 \cdot 10^{-10}$	10	11	4.54
$\ell = 3$	$7.96 \cdot 10^{-11}$	$2.66 \cdot 10^{-10}$	2	3	2.51

as a point source in the left rectangle. Figure 4.6 shows an overlay of all realizations of the quantity of interest $Q_\ell(\omega^{(i, \ell)})$ on levels $\ell = 1, 2$ and the MLMC estimate $\widehat{Q}_h^{\text{ML}}$ generated by Algorithm 2.1 for a RMS error tolerance $\varepsilon = 5 \cdot 10^{-5}$, which corresponds to a 2% relative L^2 error.

Next, we compare the cost of computing the same QoI either with or without LTS on each level. As shown in Table 4.1, the variances V_ℓ with LTS are smaller than with standard LF, which results in a smaller number of samples N_ℓ on each level. Moreover, the speed-up per sample $C_\ell^{\text{LF}}/C_\ell^{\text{LF-LTS}}$ is maximal on the coarsest levels $\ell = 0, 1$, where the difference between H_ℓ and h^ℓ is greatest. Here, the optimal values N_ℓ are determined by Algorithm 2.1 according to (2.7) while V_ℓ is estimated from (2.8). To estimate C_ℓ , we use the cost models (3.8) and (3.11) from Sections 3.2 and 3.3, respectively. For the total cost $\mathcal{C} = \sum_\ell N_\ell C_\ell$, the MLMC method with LTS is about 6.8 times faster than MLMC without LTS for the same error tolerance.

Finally, we compare the cost of the MLMC approach with that using standard MC on any given level. For the MLMC method with LTS, Table 4.2 provides not only the number of samples N_ℓ and the variance V_ℓ with respect to the differences ΔQ_ℓ , but also lists the numerical bias on each level, estimated by (2.9), and the variance V_ℓ^{MC} with respect to the

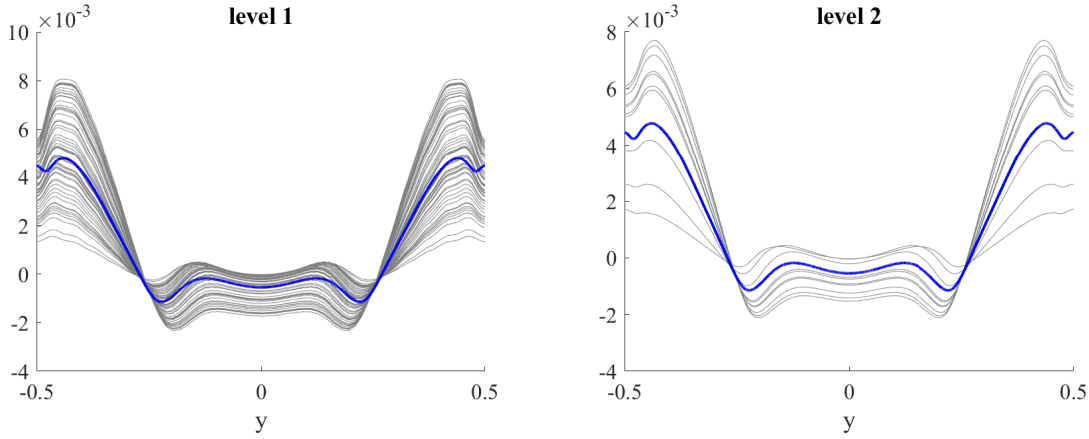


Figure 4.6. Two-dimensional narrow channel. Overlay of solutions $Q_\ell^{(i,\ell)} = u_\ell(x, y, T, \omega^{(i,\ell)})$ along the vertical line $x = -0.4$ at fixed time $T = 1$ computed by the MLMC Algorithm 2.1 with LF-LTS on levels $\ell = 1, 2$ (grey) together with the final estimate $\widehat{Q}_h^{\text{ML}}$ of $\mathbb{E}[Q]$ (blue).

quantities Q_ℓ itself:

$$(4.4) \quad \begin{aligned} V_\ell^{\text{MC}} &= \mathbb{E} \left[\|Q_\ell - \mathbb{E}[Q_\ell]\|_V^2 \right] = \mathbb{E} \left[\|Q_\ell\|_V^2 \right] - \|\mathbb{E}[Q_\ell]\|_V^2 \\ &\approx \frac{1}{N_\ell} \left(\sum_{i=1}^{N_\ell} \|Q_\ell^{(i,\ell)}\|_V^2 - \frac{1}{N_\ell} \left\| \sum_{i=1}^{N_\ell} Q_\ell^{(i,\ell)} \right\|_V^2 \right). \end{aligned}$$

Note that the MLMC Algorithm 2.1, the number of levels is determined adaptively such that the numerical bias is bounded by $\|\mathbb{E}[Q - Q_\ell]\|_V^2 < \varepsilon^2/2 = 1.25 \cdot 10^{-9}$. To bound on level ℓ the mean square error for a standard MC estimator, $\widehat{Q}_\ell^{\text{MC}}$,

$$e \left(\widehat{Q}_\ell^{\text{MC}} \right) = (N_\ell^{\text{MC}})^{-1} V_\ell^{\text{MC}} + \|\mathbb{E}[Q - Q_\ell]\|_V^2,$$

by the same error tolerance $\varepsilon^2 = 2.5 \cdot 10^{-9}$ would require $N_\ell^{\text{MC}} = 5221$ or $N_\ell^{\text{MC}} = 9779$ samples on levels $\ell = 3$ or $\ell = 2$, respectively, resulting in a 350 or 80-fold increase in computational cost over MLMC. On level $\ell = 1$, that error tolerance cannot even be reached, as the bias is already larger than ε^2 . In summary, even for numerical wave propagation where the coarsest level always ought to resolve the dominating wave-length, the multilevel strategy clearly outperforms by far a standard (single-level) MC method on any given grid.

5. Concluding remarks. To overcome the increasingly stringent bottleneck in MLMC on coarser levels due to locally refined meshes when using explicit time integration, we have introduced on each level local time-stepping (LTS), which adapts the time-step to the local CFL stability constraint. The combined LTS-MLMC algorithm thus extends the well-known robust and efficient classical MLMC algorithm for uncertainty quantification to wave propagation in complex geometry without sacrificing explicitness or inherent parallelism.

In our cost comparison of the MLMC algorithm using either standard or local time-stepping, we distinguish between two typical situations where mesh coarsening cannot occur

Table 4.2

Two-dimensional narrow channel. The variance and bias for LF-LTS applied to MLMC estimate $\widehat{Q}_h^{\text{ML}}$ with RMS error tolerance $\varepsilon = 5 \cdot 10^{-5} \approx 0.020 \cdot \|\widehat{Q}_h^{\text{ML}}\|_{L^2(-0.5,0.5)}$.

Level	number of samples	variance		numerical bias
	N_ℓ	V_ℓ (cf. (2.4))	V_ℓ^{MC} (cf. (4.4))	$\ \mathbb{E}[Q - Q_\ell]\ _V^2$
$\ell = 0$	1266	$1.21 \cdot 10^{-6}$	$1.20 \cdot 10^{-6}$	
$\ell = 1$	51	$6.98 \cdot 10^{-9}$	$7.24 \cdot 10^{-6}$	$6.45 \cdot 10^{-9}$
$\ell = 2$	10	$3.21 \cdot 10^{-10}$	$1.01 \cdot 10^{-5}$	$1.47 \cdot 10^{-9}$
$\ell = 3$	2	$7.96 \cdot 10^{-11}$	$1.25 \cdot 10^{-5}$	$9.82 \cdot 10^{-11}$

uniformly across all levels. First, when local refinement occurs inside a small fixed region of the computational domain, we have proved in Proposition 3.3 that the asymptotic complexity of the computational effort as ε^{-2} remains unchanged up to a constant factor as the desired accuracy $\varepsilon \rightarrow 0$. Depending on parameter values, however, the combined LTS-MLMC easily achieves a significant (but constant) speed-up both in theory and in our numerical examples, in fact even more so in higher dimensions. In our one-dimensional computations, for instance, we observe a 30-fold speed-up over MLMC with standard time-stepping. Second, we have considered graded meshes towards a reentrant corner, which restore the optimal convergence rates of FEM in the presence of singularity. In particular, for L -shaped domains, we have proved in Corollary 3.5 that LTS can even improve the overall asymptotic complexity of MLMC as $\varepsilon \rightarrow 0$. In other words, the smaller the desired error ε , the larger the speed-up of the LTS-MLMC algorithm over standard time integration – see also Remark 3.4.

In our analysis and numerical experiments, we have concentrated on standard continuous piecewise polynomial finite elements (with mass lumping) for the spatial discretization and on the popular leapfrog method for time discretization. Our cost estimates nonetheless hold for other spatial discretizations such as finite difference or discontinuous Galerkin methods, too. Thus, we expect a similar speed-up when replacing other explicit time integrators, such as Adams-Bashforth or Runge-Kutta methods, by their explicit LTS counterparts [21, 22, 18].

Although we have only considered pre-defined sequences of meshes featuring local refinement, the methodology and analysis presented here is also relevant for MLMC combined with adaptive mesh refinement based on a posteriori error estimators [28, 14, 17]. The combined LTS-MLMC approach will also prove useful for parabolic problems, if the LTS counterpart of explicit RK-Chebyshev methods [1] is used for time integration.

REFERENCES

- [1] A. Abdulle, M. J. Grote, and G. Rosilho de Souza, *Explicit stabilized multirate method for stiff differential equations*, arXiv:2006.00744 [math.NA], preprint.
- [2] A. Abdulle, A. Barth, and Ch. Schwab, *Multilevel Monte Carlo methods for stochastic elliptic multiscale PDEs*, SIAM Multiscale Modelling & Simulation **11** (2013), pp. 1033–1070.
- [3] I. Babuška, R. G. Kellogg, and J. Pitkäranta, *Direct and inverse error estimates for finite elements with mesh refinements*, Num. Math. **33** (1979), pp. 447–471.
- [4] M. Balesio, J. Beck, A. Pandey, L. Parisi, E. von Schwerin, and R. Tempone, *Multilevel Monte Carlo acceleration of seismic wave propagation under uncertainty*, Internat. J. Geomathematics **10** (2019),

- pp. 1–43.
- [5] A. Barth, A. Lang, and Ch. Schwab, *Multilevel Monte Carlo method for parabolic stochastic partial differential equations*, BIT **53** (2013), pp. 3–27.
 - [6] A. Barth, Ch. Schwab, and N. Zollinger, *Multi-level Monte Carlo Finite Element method for elliptic PDEs with stochastic coefficients*, Numer. Math. **119** (2011), pp. 123–161.
 - [7] C. Bierig and A. Chernov, *Convergence analysis of multilevel Monte Carlo variance estimators and application for random obstacle problems*, Numer. Math. **130** (2015), pp. 579–613.
 - [8] C. Carle and M. Hochbruck, private communication.
 - [9] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Visualization Sci. **14** (2011), pp. 3–15.
 - [10] G. Cohen, P. Joly, J. E. Roberts, and N. Tordjman, *Higher order triangular finite elements with mass lumping for the wave equation*, SIAM J. Numer. Anal. **38** (2001), no. 6, 2047–2078.
 - [11] G. Cohen, *Higher Order Numerical Methods for Transient Wave Equations*, Springer Verlag, 2002.
 - [12] J. Diaz and M. J. Grote, *Energy conserving explicit local time-stepping for second-order wave equations*, SIAM J. Sci. Comp. **31** (2009), pp. 1985–2014.
 - [13] ———, *Multilevel explicit local time-stepping for second-order wave equations*, Comp. Meth. Appl. Mech. Engin. **291** (2015), 240–265.
 - [14] M. Eigel, Ch. Merdon, and J. Neumann, *An adaptive multilevel Monte Carlo method with stochastic bounds for quantities of interest with uncertain data*, SIAM/ASA Journal on Uncertainty Quantification **4** (2016), pp. 1219–1245.
 - [15] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York, 1996.
 - [16] M. B. Giles, *Multilevel Monte Carlo path simulation*, Operations Research **56** (2008), pp. 607–617.
 - [17] M. B. Giles, C. Lester, and J. Whittle, *Non-nested adaptive timesteps in multilevel Monte Carlo computations*, Monte Carlo and Quasi-Monte Carlo Methods, Springer, 2015, pp. 303–314.
 - [18] M. J. Grote, M. Mehlin, and T. Mitkova, *Runge-Kutta based explicit local time-stepping methods for wave propagation*, SIAM J. Sci. Comp. **37** (2015), no. 2, pp. A747–A775.
 - [19] M. J. Grote, M. Mehlin, and S. A. Sauter, *Convergence analysis of energy conserving explicit local time-stepping methods for the wave equation*, SIAM J. Numer. Anal. **56** (2018), no. 2, 994–1021.
 - [20] M. J. Grote, S. Michel, and S. A. Sauter, *Stabilized leapfrog based local time-stepping method for the wave equation*, Math. Comp. **90** (2021), pp. 2603–2643.
 - [21] M. J. Grote and T. Mitkova, *Explicit local time-stepping methods for Maxwell’s equations*, J. Comput. Appl. Math. **234** (2010), pp. 3283–3302.
 - [22] ———, *Explicit local time-stepping methods for time-dependent wave propagation*, Direct and Inverse Problems in Wave Propagation and Applications, De Gruyter, Berlin, 2013, pp. 187–218.
 - [23] M. J. Grote, A. Schneebeli, and D. Schötzau, *Discontinuous Galerkin finite element methods for the wave equation*, SIAM J. Numer. Anal. **44** (2006), pp. 2408–2431.
 - [24] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, Springer-Verlag, Berlin, 2002.
 - [25] A. L. Haji-Ali, F. Nobile, E. von Schwerin, and R. Tempone, *Optimization of mesh hierarchies in multilevel Monte Carlo samplers*, Stoch PDE: Anal Comp **4** (2016), pp. 76–112.
 - [26] S. Heinrich, *Monte Carlo complexity of global solution of integral equations*, Journal of Complexity, **14** (1998), pp. 151–175.
 - [27] S. Heinrich, and E. Sindambiwe, *Monte Carlo complexity of parametric integration*, Journal of Complexity, **15** (1999), pp. 317–341.
 - [28] H. Hoel, E. von Schwerin, A. Szepessy, and R. Tempone, *Implementation and analysis of an adaptive multilevel Monte Carlo algorithm*, Monte Carlo methods and applications, **20** (2014), pp. 1–41.
 - [29] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer Series in Computational Mathematics, vol. 33, Springer-Verlag, Berlin, 2003.
 - [30] J. Lions and E. Magenes, *Non-homogeneous Boundary Value Problems and Applications*, Vol. I, Springer-Verlag, New York, 1972.
 - [31] S. Minisini, E. Zhebel, A. Kononov, and W. A. Mulder, *Local time stepping with the discontinuous Galerkin method for wave propagation in 3D heterogeneous media*, Geophysics **78** (2013), T67–T77.
 - [32] S. Mishra and Ch. Schwab, *Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data*, Math. Comp. **81** (2012), pp. 1979–2018.
 - [33] S. Mishra, Ch. Schwab, and J. Sukys, *Multi-level Monte Carlo finite volume methods for nonlinear systems*

- of conservation laws in multi-dimensions*, J. Comput. Phys. **231** (2012), pp. 3365–3388.
- [34] ———, *Multi-level Monte Carlo finite volume methods for uncertainty quantification of acoustic wave propagation in random heterogeneous layered medium*, J. Comput. Phys. **312** (2016), pp. 192–217.
- [35] M. Motamed and D. Appelö, *A multi-order discontinuous Galerkin Monte Carlo method for hyperbolic problems with stochastic parameters*, SIAM J. Numer. Anal. **56** (2018), pp. 448–468.
- [36] M. Motamed, F. Nobile, and R. Tempone, *A stochastic collocation method for the second order wave equation with a discontinuous random speed*, Numer. Math, **123** (2013), pp. 493–536.
- [37] ———, *Analysis and computation of the elastic wave equation with random coefficients*, Comput. & Math. with Appl., **70** (2015), pp. 2454–2473.
- [38] F. Müller and Ch. Schwab, *Finite elements with mesh refinement for wave equations in polygons*, J. Comput. Appl. Math. **283** (2015), pp. 163–181.
- [39] ———, *Finite elements with mesh refinement for elastic wave propagation in polygons*, Math. Meth. Appl. Sci. **39** (2016), pp. 5027–5042.
- [40] M. Pisaroni, F. Nobile, and P. Leyland, *A Continuation Multi Level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible inviscid aerodynamics*, Comput. Methods Appl. Mech. Engrg. **326** (2017), pp. 20–50.
- [41] M. Rietmann, M. J. Grote, D. Peter, and O. Schenk, *Newmark local time stepping on high-performance computing architectures*, J. Comput. Phys. **334** (2017), 308–326.
- [42] T. J. Rivlin, *The Chebyshev Polynomials*, Wiley, New York, 1974.
- [43] Ch. Schwab, *p - and hp - Finite Element Methods*, Oxford Univ. Press, New York, 1998.
- [44] M. Stanglmeier, N. C. Nguyen, J. Peraire, and B. Cockburn, *An explicit hybridizable discontinuous Galerkin method for the acoustic wave equation*, Comput. Methods Appl. Mech. Engrg. **300** (2016), pp. 748–769.
- [45] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Num. Math., **125** (2013), pp. 569–600.
- [46] J. Tryoen, O. Le Maître, M. Ndjinga, and A. Ern, *Intrusive Galerkin methods with upwinding for uncertain nonlinear hyperbolic systems*, J. Comput. Phys. **229** (2010), pp. 6485–6511.