Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Multi-level Monte Carlo finite volume methods for uncertainty quantification in nonlinear systems of balance laws

S. Mishra, Ch. Schwab and J. Šukys

Research Report No. 2012-08
April 2012

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

MULTI-LEVEL MONTE CARLO FINITE VOLUME METHODS FOR UNCERTAINTY QUANTIFICATION IN NONLINEAR SYSTEMS OF BALANCE LAWS

S. MISHRA, CH. SCHWAB, AND J. ŠUKYS

ABSTRACT. A mathematical formulation of conservation and of balance laws with random input data, specifically with random initial conditions, random source terms and random flux functions, is reviewed. The concept of random entropy solution is specified. For scalar conservation laws in multi-dimensions, recent results on the existence and on the uniqueness of random entropy solutions with finite variances are presented. The combination of Monte Carlo sampling with Finite Volume Method discretization in space and time for the numerical approximation of the statistics of random entropy solutions is proposed.

The finite variance of random entropy solutions is used to prove asymptotic error estimates for combined Monte Carlo Finite Volume Method discretizations of scalar conservation laws with random inputs. A Multi-Level extension of combined Monte Carlo Finite Volume Method (MC-FVM) discretizations is proposed and asymptotic error bounds are presented in the case of scalar, nonlinear hyperbolic conservation laws. Sparse tensor constructions for the computation of compressed approximations of two- and k-point space-time correlation functions of random entropy solutions are introduced.

Asymptotic error versus work estimates indicate superiority of Multi-Level versions of MC-FVM over the plain MC-FVM, under comparable assumptions on the random input data. In particular, it is shown that these compressed sparse tensor approximations converge essentially at the same rate as the MLMC-FVM estimators for the mean solutions.

Extensions of the proposed algorithms to nonlinear, hyperbolic systems of balance laws are outlined. Multiresolution discretizations of random source terms which are exactly bias-free are indicated.

Implementational aspects of these Multi-Level Monte Carlo Finite Volume methods, in particular results on large scale random number generation, scalability and resilience on emerging massively parallel computing platforms, are discussed.

Date: March 5, 2013.

1991 *Mathematics Subject Classification.* 65M12, 65M22, 65M08, 65M55, 65C05, 35L65.

Key words and phrases. Balance Laws, Shallow-Water, Compressible Euler, MHD, Magnetohydrodynamics, Uncertainty Quantification, Multi-Level Monte Carlo, Finite Volume Methods, Parallel Computing, Load Balancing.

Acknowledgement. This work is performed as part of ETH interdisciplinary research grant CH1-03 10-1. CS acknowledges partial support by the European Research Council under grant ERC AdG No. 247277 - STAHDPDE. SM acknowledges partial support by the European Research Council under grant ERC StG No. 306279 - SPARCCLE .

Contents

1. Introduction

1.1. **Weak solutions of systems of balance laws.**

Systems of balance laws are nonlinear systems of partial differential equations (PDEs) of the form:

$$\begin{aligned} &\mathbf{U}_t + \sum_{j=1}^{d} \frac{\partial}{\partial x_j}(\mathbf{F}_j(\mathbf{U})) = \mathbf{S}, \quad x = (x_1, \ldots, x_d) \in \mathbb{R}^d, t > 0, \\ &\mathbf{U}(x, 0) = \mathbf{U}_0(x), \quad x \in \mathbb{R}^d. \end{aligned} \tag{1.1}$$

Here, $\mathbf{U} : \mathbb{R}^d \mapsto \mathbb{R}^m$ is the vector of unknowns and $\mathbf{F}_j : \mathbb{R}^m \mapsto \mathbb{R}^m$, $j = 1, ..., d$ denotes the flux vector for the j-th direction with the positive integer m denoting the dimension of the state space, and $\mathbf{S} : \mathbb{R}^d \mapsto \mathbb{R}^m$ denotes the so-called *source term.* If $\mathbf{S} = \mathbf{0} \in \mathbb{R}^m$, (1.1) is termed a *conservation law.*

Examples of balance laws include the Euler equations of gas dynamics, the equations of MagnetoHydroDynamics, the shallow water equations of oceanography and the Buckley-Leverett equations modeling flow of two phases in a porous medium.

It is well known that solutions of (1.1) develop discontinuities in finite time, even when the initial data is smooth [10]. Hence, solutions of (1.1) are sought (and computed) in the weak sense: a weak solution $\mathbf{U} \in (L^1_{\mathrm{loc}}(\mathbb{R}^d \times \mathbb{R}_+))^m$ is required to satisfy the integral identity

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}^d} \left(\mathbf{U}(\varphi)_t + \sum_{j=1}^{d} \mathbf{F}_j(\mathbf{U})\varphi_{x_j} + \mathbf{S}(\mathbf{U})\varphi \right) dxdt + \int_{\mathbb{R}^d} \mathbf{U}_0(x)\varphi(x, 0)dx = 0 \ , \tag{1.2}$$

for all test functions $\varphi \in (C^1_0([0, \infty) \times \mathbb{R}^d))^m$. It is classical that weak solutions are not necessarily unique [10]. Additional admissibility criteria such as *entropy conditions* are necessary to obtain uniqueness. In space dimension $d > 1$, rigorous existence and uniqueness results for deterministic, nonlinear conservation laws and for generic initial data are available only for the scalar case, i.e., in the case $m = 1$.

1.2. **Numerical methods.**

Numerical schemes have assumed the role of being the main tools for the study of systems of balance (conservation) laws. Many efficient numerical schemes for approximating systems of conservation laws are currently available. They include the Finite Volume, conservative Finite Difference and Discontinuous Galerkin methods, see [25, 19]. For simplicity of exposition, we present the standard Finite Volume Method, following [25].

We consider here (again for the simplicity of exposition) a fixed, positive time step $\Delta t > 0$ and a triangulation $\mathcal{T}$ of the bounded spatial domain $\mathbf{D} \subset \mathbb{R}^d$ of interest. Here, the term triangulation $\mathcal{T}$ will be understood as a partition of the physical domain $\mathbf{D}$ into a finite set of disjoint open convex polyhedra $K \subset \mathbb{R}^d$ with boundary ∂K being a finite union of closed plane faces (which are, in these notes, polyhedra contained in $d-1$ dimensional hyperplanes, understood as points in the case $d = 1$). Let $\Delta x_K := \operatorname{diam} K = \sup\{|x - y| : x, y \in K\}$ and by $\Delta x(\mathcal{T}) := \max\{\Delta x_K : K \in \mathcal{T}\}$ denote the *mesh width* of $\mathcal{T}$. For any volume $K \in \mathcal{T}$, we define the *set* $\mathcal{N}(K)$ *of neighbouring volumes*

$$\mathcal{N}(K) := \{K' \in \mathcal{T} : K' \neq K \wedge \mathrm{meas}_{d-1}(\overline{K} \cap \overline{K'}) > 0\}. \tag{1.3}$$

Note that volumes $K' \in \mathcal{T}$ whose closure shares a set of $d - 1$ measure zero with $\overline{K}$ is *not* a neighboring volume. For every $K \in \mathcal{T}$ and $K' \in \mathcal{N}(K)$ denote $\nu_{K,K'}$

to be the exterior unit normal vector, i.e. pointing outward from the volume K at the face $\overline{K} \cap \overline{K'}$. We set:

$$\lambda = \Delta t / \min\{\Delta x_K : \; K \in \mathcal{T}\} \tag{1.4}$$

by assuming a uniform discretization in time with constant time step Δt. The constant λ is determined by a standard CFL condition (see [19]) based on the maximum wave speed.

Then, an explicit first-order finite volume ([19]) for approximating (1.1) is given by

$$\mathbf{U}_K^{n+1} = \mathbf{U}_K^n - \frac{\Delta t}{\mathrm{meas}(K)} \sum_{K' \in \mathcal{N}(K)} \mathbf{F}(\mathbf{U}_K^n, \mathbf{U}_{K'}^n) + \mathbf{S}_K^n, \tag{1.5}$$

where

$$\mathbf{U}_K^n \approx \frac{1}{\mathrm{meas}(K)} \int_K \mathbf{U}(\mathbf{x}, t^n) d\mathbf{x}$$

is an approximation to the cell average of the solution and $\mathbf{F}(\cdot,\cdot)$ is a numerical flux that is consistent with $\mathbf{F} \cdot \nu_{K,K'}$. Numerical fluxes are usually derived by (approximately) solving Riemann problems at each cell interface resulting in the Godunov, Roe and HLL fluxes, see e.g. [25]. The discrete source $\mathbf{S}$ in (1.5) can be a straight-forward evaluation,

$$\mathbf{S}_K^n = \frac{1}{\mathrm{meas}(K)} \int_K \mathbf{S}(\mathbf{x}, \mathbf{U}_K^n) d\mathbf{x}$$

or something more sophisticated, for instance the well-balanced version of the bottom topography source term [15] in shallow-water simulations.

Higher order spatial accuracy is obtained by reconstructing $\mathbf{U}$ from $\mathbf{U}_K^n$ in non-oscillatory piecewise polynomial functions in terms of the TVD [25], ENO [21] and WENO [40] procedures or by the Discontinuous Galerkin method (see, e.g. [9]). Higher order temporal accuracy is achieved by employing strong stability preserving Runge-Kutta methods [20]. Space-time DG-discretizations can also be employed for uniformly high-order spatio-temporal accuracy [6].

1.3. **Uncertainty Quantification (UQ).** Any numerical scheme approximating (1.1) requires the initial data $\mathbf{U}_0$, the source term $\mathbf{S}$ and the flux functions $\mathbf{F}_j$ as inputs. However, in practice, these inputs cannot be measured precisely. As a first example, consider the modeling of propagation of tsunami waves with the shallow water equations. It is not possible to measure the initial water displacement (at tsunami source) with any precision in real time. Similarly, the bottom topography is measured with sonar equipment and this data collection is prone to uncertainty. Thus, the inputs (initial data and source terms) to the underlying shallow water equations are uncertain. As a second example, consider the modeling of an oil and gas reservoir. Water flooding is modeled by the equations of two phase flow. However, the rock permeability as well as the relative permeabilities of each phase with respect to the other, need to measured. Again, the measurement process is characterized by uncertainty. Consequently, the inputs (the fluxes) to the underlying two-phase flow equations are uncertain. This uncertainty in the inputs for (1.1) results in the propagation of uncertainty into the solution. The modeling and approximation of the propagation of uncertainty in the solution due to uncertainty in inputs constitutes the theme of uncertainty quantification (UQ).

Uncertainty in inputs and solutions of PDEs is frequently modeled in a probabilistic manner. The inputs are random fields with prescribed probability laws.

Then, the solution is also realized as a random field and its law and the (deterministic!) statistical moments of the solutions like the expectation and variance are the quantities of engineering interest.

It is a non-trivial matter to develop efficient algorithms for quantifying uncertainty in solutions of balance (conservation) laws with random inputs. The biggest challenge lies in the fact that discontinuities in physical space (which inevitably arise in solutions of nonlinear hyperbolic conservation laws) may propagate into parametric representations of the probability densities (laws) of the random solutions. A robust numerical method should be able to deal with this phenomenon. Another challenge lies in dealing with the fact that the number of random sources driving the uncertainty may be very large (possibly countably infinite in the case of random field inputs parametrized by Karhunen–Loève expansions).

The design of efficient numerical schemes for quantifying uncertainty in solutions of partial differential equations has seen a lot of activity in recent years.

Among the most popular methods (particularly for elliptic and parabolic PDEs) are the stochastic Galerkin methods based on generalized Polynomial Chaos (gPC for short). An incomplete list of references on gPC methods for uncertainty quantification in hyperbolic conservation laws includes [3, 8, 27, 43, 35, 45] and other references therein. Although these *deterministic* methods show some promise, they suffer from the disadvantage that they are highly *intrusive*: existing codes for computing deterministic solutions of balance (conservation) laws need to be completely reconfigured for implementation of the gPC based stochastic Galerkin methods. An alternative class of methods for quantifying uncertainty in PDEs are the stochastic collocation methods, see [48] for a general review and [28, 47] for modifications of these methods near discontinuities. Stochastic collocation methods are non-intrusive and easier to parallelize than the gPC based stochastic Galerkin methods. However, the lack of regularity of the solution with respect to the stochastic variables (the solution can be discontinuous in the stochastic variables) impedes efficient performance of both the stochastic Galerkin as well as the stochastic collocation methods (see, however, [31]).

Another class of methods for computational uncertainty quantification in numerical solutions of PDEs are *statistical sampling methods*, most notably Monte Carlo (MC) sampling. In a MC method, the probability space is *sampled* and the underlying deterministic PDE is solved for each sample. The MC samples of numerical solutions of the PDE are combined into statistical estimates of expectation and other statistical moments of the random solution which are necessary to quantify uncertainty. In uncertainty quantification for hyperbolic scalar conservation laws with random initial data, MC type methods together with Finite Volume (FV) spatio-temporal discretizations of the PDE were proposed in a recent paper [30]. The MC-FVM methods were analyzed in the context of a scalar conservation law with random initial data and corresponding estimates of the combined discretization and statistical sampling errors were obtained. MC methods are non-intrusive; they can, therefore, be based on existing, deterministic CFD solvers. As it was shown in [30], MC methods converge at rate $1/2$ as the number M of MC samples increases with each "sample" corresponding to a full, deterministic flow simulation. The asymptotic convergence rate $M^{-1/2}$ in terms of the number M of MC samples is non-improvable by the central limit theorem. To achieve a sampling error which is of the order of the discretization error, MC Finite Volume Methods therefore require a large number of "samples", with each sample consisting of the numerical solution of (1.1) with a given draw of initial data (and/or random flux and random source term). This slow convergence entails high computational costs for MC type UQ methods in CFD. In particular, accurate quantification of uncertainty by direct MC methods combined with available solvers for hyperbolic systems of conservation or balance laws in several space dimensions becomes very costly, even with a moderately large number of random inputs.

One is therefore led to explore alternative approaches. In recent years, adaptive deterministic discretization methods of *polynomial chaos type* have received substantial attention. These methods have been, in connection with elliptic and parabolic problems, found to be able to facilitate convergence rates which are higher than the (mean square) rate $1/2$ afforded by MC sampling, under appropriate conditions on the input data. While their implementation is intrusive and therefore more involved than that of MC methods (see, e.g. [43]), potentially higher convergence rates than MC-FVM can be achieved by these methods since they approximate directly certain statistical moments of random solutions (in the form of polynomial chaos expansions of random solutions) which recently have been found to exhibit additional smoothness as compared to "pathwise" solutions [39] which, typically, feature discontinuities. In general, however, the lack of regularity of solutions in nonlinear hyperbolic conservation laws and the nonstandard nature of the strongly coupled, large hyperbolic systems (i.e. the dimension m of the state space in (1.1) is a discretization parameter) which result from the so-called "stochastic Galerkin projection" (i.e. a mean - square projection of the conservation law onto a m-term truncated polynomial chaos expansion) indicates at present for this approach only a limited range of applicability (see, however, [39] for evidence of a *mechanism for smoothing through ensemble averaging* in random entropy solutions of hyperbolic conservation laws).

In order to address the slow convergence of MC methods, we proposed in [30] a novel *Multi-Level Monte Carlo* Finite Volume (MLMC-FVM) algorithm for *scalar* conservation laws in [30]. Multi-Level MC methods were introduced by S. Heinrich for numerical quadrature [22] and developed by M. Giles to enhance the efficiency of path simulations for Itô stochastic ordinary differential equations in [17, 18]. More recently, MLMC Finite Element Methods for elliptic problems with stochastic coefficients were introduced by Barth, Schwab and Zollinger in [5]. The analysis in these references, in particular in [30, 5], reveals that the *MLMC is able to deliver converged numerical approximations to statistics of uncertain solutions of partial differential equations in computational complexity comparable to that of one numerical solve of a single "path", i.e. a single realization of the random input data*, under in a sense minimal regularity on the solution. Specifically, only finiteness of second moments of the random solution is needed, when the size of solution is measured in terms of a slightly stronger norm than the norm appearing in energy bounds.

We note that the Multi-Level Monte Carlo method is not the only way to enhance the standard MC method. For instance, variance reduction techniques (such as importance sampling, control variates, stratified sampling, correlated sampling, and conditional Monte Carlo) [14] or quasi Monte Carlo methods (see, eg., [11] and the references there for a survey) are also available. Efficient variance reduction, however, requires additional a-priori analysis and knowledge about the second moments of the a-priori unknown solution of the random PDE; such knowledge is rarely available. Quasi Monte Carlo methods on the other hand require a parametrization and smoothness assumptions of the unknown solution field which may not hold for nonlinear problems with discontinuous solutions considered here. If these assumptions are not verified, an adhoc application of QMC integration in problems with many random sources (such as the ones considered in Section 6.3 of these notes) may lead to simulation methods with a lack of robustness with respect to the curse of the stochastic dimension (i.e. cases when many random variables are involved) as compared to standard Monte Carlo methods.

1.4. **Objectives of these notes.** The present paper has several objectives. First, we will outline the concept of random entropy solutions for *scalar, multi-dimensional* conservation laws with random inputs. We present a mathematical framework of well-posedness of such problems and provide, in particular, precise statements on the existence and the uniqueness of random entropy solutions for *scalar, multi-dimensional* conservation laws with random inputs.

To this end, we recapitulate results of our recent paper [30] on random entropy solutions for scalar conservation laws with uncertain initial data. Furthermore, we outline extensions of the results on wellposedness and the existence and uniqueness of random entropy solutions for a scalar conservation law with random flux. Further details and complete mathematical developments of these results are available in [31]. The corresponding theory will provide a rigorous basis for the design and analysis of Multi-Level Monte Carlo Finite Volume Methods for the efficient computational quantification of uncertainty in a scalar, hyperbolic conservation law with random input data.

The second objective of this paper is to outline essentials on statistical sampling methods of the Monte Carlo (MC) and Multi-Level Monte Carlo (MLMC) type, with particular attention to their use in computational fluid dynamics. We summarize recent results from [30, 32, 33, 31], describe the algorithms, outline the convergence and complexity analysis and present several numerical experiments to demonstrate the efficiency of the proposed algorithms. Systems of conservation laws with uncertain initial data, uncertain source terms and uncertain flux functions are considered in our numerical examples.

The remainder of the paper is organized as follows: in Section 2, the mathematical theory of random entropy solutions of scalar conservation laws with uncertain initial data and uncertain flux functions is outlined. The MC algorithms and MLMC algorithms are presented in Sections 3 and 4 respectively. Details of implementation are provided in Section 5 and numerical experiments are presented in Section 6. Sparse tensor methods to efficiently compute higher statistical moments of the random entropy solutions are also discussed within Section 6. The paper concludes with a description and demonstration of MLMC-FVM approximation of random event probabilities in Section 7.

2. Random entropy solutions

In this section, we introduce the notion of random entropy solutions for conservation laws with random initial data and with random flux functions. We show that scalar conservation laws are well-posed in the sense that we have existence and uniqueness of random entropy solutions for scalar conservation laws with, in particular, continuous dependence of random entropy solutions on the statistical input data of the scalar conservation laws. Since, even in the deterministic case, rigorous results are available only for the scalar problem, in this section we will restrict the mathematical developments to the scalar case ($m = 1$ in (1.1)). We start with some mathematical preliminaries.

2.1. **Random Fields.** Let $(\Omega, \mathcal{F})$ be a measurable space, with Ω denoting the set of all elementary events, and $\mathcal{F}$ a σ-algebra of all possible events in our probability model. If $(E, \mathcal{G})$ denotes a second measurable space, then an *E-valued random variable* (or random variable taking values in E) is any mapping $X : \Omega \to E$ such that the set $\{\omega \in \Omega\colon\ X(\omega) \in A\} = \{X \in A\} \in \mathcal{F}$ for any $A \in \mathcal{G}$, i.e. such that X is a $\mathcal{G}$-measurable mapping from Ω into E.

Assume now that E is a metric space; with the Borel σ-field $\mathcal{B}(E)$, $(E, \mathcal{B}(E))$ is a measurable space and we shall always assume that E-valued random variables $X : \Omega \to E$ will be $(\mathcal{F}, \mathcal{B}(E))$ measurable. If E is a separable Banach space with norm $\| \circ \|_E$ and (topological) dual E^*, then $\mathcal{B}(E)$ is the smallest σ-field of subsets of E containing all sets

$$\{x \in E : \varphi(x) \le \alpha\},\ \varphi \in E^*,\ \alpha \in \mathbb{R}\,. \tag{2.1}$$

Hence if E is a separable Banach space, $X : \Omega \to E$ is an E-valued random variable if and only if for every $\varphi \in E^*$, $\omega \longmapsto \varphi(X(\omega)) \in \mathbb{R}^1$ is a $\mathbb{R}^1$-valued random variable.

The random variable $X : \Omega \to E$ is called *Bochner integrable* if, for any probability measure $\mathbb{P}$ on the measurable space $(\Omega, \mathcal{F})$,

$$\int_\Omega \|X(\omega)\|_E\, \mathbb{P}(d\omega) < \infty\,. \tag{2.2}$$

A probability measure $\mathbb{P}$ on $(\Omega, \mathcal{F})$ is any σ-additive set function from Ω into $[0, 1]$ such that $\mathbb{P}(\Omega) = 1$, and the measure space $(\Omega, \mathcal{F}, \mathbb{P})$ is called probability space. We shall always assume, unless explicitly stated, that $(\Omega, \mathcal{F}, \mathbb{P})$ is complete.

If $X : (\Omega, \mathcal{F}) \to (E, \mathcal{E})$ is a random variable, $\mathcal{L}(X)$ denotes the *law of X under* $\mathbb{P}$, i.e.

$$\mathcal{L}(X)(A) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \in A\}) \quad \forall A \in \mathcal{E}\,. \tag{2.3}$$

The image measure $\mu_X = \mathcal{L}(X)$ on $(E, \mathcal{E})$ is called law or distribution of X.

We shall require for $1 \le p \le \infty$ Bochner spaces of p-summable random variables X taking values in the Banach-space E. By $L^1(\Omega, \mathcal{F}, \mathbb{P}; E)$ we denote the set of all (equivalence classes of) integrable, E-valued random variables X. We equip it with the norm

$$\|X\|_{L^1(\Omega;E)} = \int_\Omega \|X(\omega)\|_E\, \mathbb{P}(d\omega) = \mathbb{E}(\|X\|_E)\,. \tag{2.4}$$

More generally, for $1 \le p < \infty$, we define $L^p(\Omega, \mathcal{F}, \mathbb{P}; E)$ as the set of p-summable random variables taking values E and equip it with norm

$$\|X\|_{L^p(\Omega;E)} := (\mathbb{E}(\|X\|_E^p))^{1/p},\ 1 \le p < \infty\,. \tag{2.5}$$

For $p = \infty$, we denote by $L^\infty(\Omega, \mathcal{F}, \mathbb{P}; E)$ the set of all E-valued random variables which are $\mathbb{P}$-almost surely bounded. This set is a Banach space equipped with the norm

$$\|X\|_{L^\infty(\Omega;E)} := \operatorname*{ess\,sup}_{\omega\in\Omega} \|X(\omega)\|_E\,. \tag{2.6}$$

If $T < \infty$ and $\Omega = [0,T]$, $\mathcal{F} = \mathcal{B}([0,T])$, we write $L^p([0,T];E)$. Note that for any separable Banach-space E, and for any $r \geq p \geq 1$,

$$L^r(0,T;E),\ C^0([0,T];E) \in \mathcal{B}(L^p(0,T;E))\,. \tag{2.7}$$

2.2. **k-th moments.** For $k \in \mathbb{N}$ and separable Banach space X, we denote by $X^{(k)} = \underbrace{X \otimes \cdots \otimes X}_{k \text{ times}}$ the k-fold tensor product of k copies of X. Throughout the following, we shall assume the k-fold tensor product of the Banach-space X with itself, i.e. $X^{(k)}$, to be equipped with a cross norm $\| \circ \|_{X^{(k)}}$ which satisfies

$$\|u_1 \otimes \cdots \otimes u_k\|_{X^{(k)}} = \|u_1\|_X \ldots \|u_k\|_X\,. \tag{2.8}$$

We refer to [30, Section 3.4] and to the references of [30] for more information on k-fold tensor products $X^{(k)}$ of a Banach space X and for norms on $X^{(k)}$.

In particular, for $X = L^p(\mathbb{R}^d)$, $1 \leq p < \infty$, we get from Fubini's theorem the isomorphism

$$L^p(\mathbb{R}^d)^{(k)} \cong L^p(\mathbb{R}^{kd})\,. \tag{2.9}$$

For $k \in \mathbb{N}$ and for $u \in L^k(\Omega;X)$, we consider the random field $(u)^{(k)}$ defined by $\underbrace{u(\omega) \otimes \cdots \otimes u(\omega)}_{k-\text{times}}$. Then

$$(u)^{(k)} = u \otimes \cdots \otimes u \in L^1(\Omega; X^{(k)}) \tag{2.10}$$

and, by (2.8), we have

$$\|(u)^{(k)}\|_{L^1(\Omega;X^{(k)})} = \int_\Omega \|u(\cdot,\omega)\|_X^k \,\mathbb{P}(d\omega) = \|u\|^k_{L^k(\Omega,X)} < \infty\,. \tag{2.11}$$

Therefore, $(u)^{(k)} \in L^1(\Omega, X^{(k)})$ and *the k-th moment* (or k-point correlation function of u)

$$\mathcal{M}^k(u) := \mathbb{E}[(u)^{(k)}] \in X^{(k)} \tag{2.12}$$

is well-defined as a (deterministic) element of $X^{(k)}$ for $u \in L^k(\Omega;X)$.

2.3. **Random initial data.** Equipped with the above notions, we first model uncertain initial data by assuming $(\Omega, \mathcal{F}, \mathbb{P})$ as the underlying probability space and realizing the uncertain initial data as a random field u_0, i.e. a $L^1(\mathbb{R}^d)$-valued random variable which is a $L^1(\mathbb{R}^d)$ measurable map

$$u_0 : (\Omega, \mathcal{F}) \longmapsto \big(L^1(\mathbb{R}^d),\ \mathcal{B}(L^1(\mathbb{R}^d))\big)\,. \tag{2.13}$$

We assume further that

$$u_0(\cdot,\omega) \in L^\infty(\mathbb{R}^d) \cap BV(\mathbb{R}^d) \quad \mathbb{P}\text{-a.s.}, \tag{2.14}$$

which is to say that

$$\mathbb{P}(\{\omega \in \Omega : u_0(\cdot,\omega) \in (L^\infty \cap BV)(\mathbb{R}^d)\}) = 1\,. \tag{2.15}$$

Since $L^1(\mathbb{R}^d)$ and $\mathrm{Lip}(\mathbb{R}^d;\mathbb{R}^d)$ are separable, (2.13) is well defined and we may impose for $k \in \mathbb{N}$ the *k-th moment condition*

$$\|u_0\|_{L^k(\Omega;L^1(\mathbb{R}^d))} < \infty\,, \tag{2.16}$$

where the Bochner spaces with respect to the probability measure are defined in (2.4), (2.5) above.

2.4. **Random Flux Functions for SCL.** Noting that the space $E = \mathrm{Lip}(\mathbb{R}; \mathbb{R}^d)$ is separable, we concentrate on the case of spatially homogeneous random flux functions and follow [31]. The definition of random flux for scalar conservation laws (i.e. for the case $m = 1$ in (2.1)) that we shall work with is

Definition 2.1. *A* (spatially homogeneous) random flux *for a scalar conservation law is a random field taking values in the separable Banach space* $E = \mathrm{Lip}(\mathbb{R}^1; \mathbb{R}^d)$*, i.e. a measurable mapping from* $(\Omega, \mathcal{F})$ *to* $(\mathrm{Lip}(\mathbb{R}^1; \mathbb{R}^d); \mathcal{B}(\mathrm{Lip}(\mathbb{R}; \mathbb{R}^d)))$*. A* bounded random flux *is a random flux whose* $\mathrm{Lip}(\mathbb{R}^1; \mathbb{R}^d)$*-norm is bounded* $\mathbb{P}$*-a.s., i.e.*

$$\exists 0 < B(f) < \infty : \quad \|f(\omega; \cdot)\|_{\mathrm{Lip}(\mathbb{R}^1; \mathbb{R}^d)} \le B(f) \quad \mathbb{P} - a.s. \tag{2.17}$$

We observe that a bounded random flux has finite statistical moments of any order. Of particular interest will be the *second moment of a bounded random flux* (i.e. its "two-point correlation in state-space"). The existence of such a state-space correlation function is addressed in the following lemma from [31], to which we refer for the proof.

Lemma 2.2. *Let f be a bounded random flux as in Definition 2.1 which belongs to* $L^2(\Omega, d\mathbb{P}; \mathrm{Lip}(\mathbb{R}; \mathbb{R}^d))$*. Then its covariance function, i.e. its centered second moment defined by*

$$\mathrm{Cov}[f](v, v') := \mathbb{E}\left[(f(\cdot; v) - \mathbb{E}[f(\cdot; v)]) \otimes (f(\cdot; v') - \mathbb{E}[f(\cdot; v')])\right] \tag{2.18}$$

is well-defined for all $v, v' \in \mathbb{R}$ and there holds

$$\mathrm{Cov}[f] \in \mathrm{Lip}(\mathbb{R} \times \mathbb{R}; \mathbb{R}^{d \times d}_{sym}). \tag{2.19}$$

The two point correlation function of a bounded random flux allows, as is well-known in statistics, for spectral decompositions of the random flux in terms of eigenpairs of its covariance operator, which is a compact and self-adjoint integral operator on square-integrable flux functions with kernel function $\mathrm{Cov}[f](v, v')$ defined in (2.18). We remark in passing that our assumption of Lipschitz continuity of (realizations of) random flux functions entails linear growth of such fluxes as the state variables tend to infinity, ie. as $|v| \to \infty$ which, at first sight, precludes considering the covariance operator on the space of square integrable flux functions. In [31], we circumvent these integrability issues for scalar conservation laws by truncating the state space to a bounded interval $[-R, R]$ with sufficiently large $R > 0$. By classical $L^\infty(\mathbb{R}^d)$ bounds on entropy solutions of scalar conservation laws, for sufficiently large values of the flux cutoff R, any realization of the random scalar conservation law will "see" only the flux function for states which (in absolute value) are below the threshold values R; accordingly, it suffices to consider the flux covariance operator only as integral operator on $L^2(-R, R)$ which is the view taken in [31].

As a concrete example for a random flux, we have the following representation using the Karhunen-Loeve (KL) expansion.

Example: Karhunen–Loève expansion of bounded random flux. Consider a bounded random flux $f(\omega; u)$ in the sense of Definition 2.1. By Lemma 2.2, its covariance function $\mathrm{Cov}[f]$ is well-defined; for $0 < R < \infty$ we denote by $\mathcal{C}^R_f$ the integral operator with bi-Lipschitz kernel $\mathrm{Cov}[f](u, v)$, defined on $L^2(-R, R)$ by

$$\mathcal{C}^R_f[\Phi](u) := \int_{|v| \le R} \mathrm{Cov}[f](u, v)\Phi(v)dv \; . \tag{2.20}$$

As explained above, the covariance operator $\mathcal{C}_f^R$ describes the covariance structure of the random flux on the set $[-R,R]$ of states. Given initial data $u_0 \in L^\infty(\mathbb{R}^d)$ with a-priori bound $\|u_0\|_{L^\infty(\mathbb{R}^d)} \leq R$, the unique entropy solution $S(t)u_0$ of of the *deterministic* SCL will, for all $t > 0$, take values in $[-\|u_0\|_{L^\infty(\mathbb{R}^d)}, \|u_0\|_{L^\infty(\mathbb{R}^d)}]$. For random flux and random initial data, therefore, we will continue under the assumption

$$R > \operatorname*{ess\,sup}_{\omega\in\Omega} \|u_0(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)} \tag{2.21}$$

will ensure that $\mathcal{C}_f^R$ will "capture" all possible states.

For every $0 < R < \infty$, the integral operator $\mathcal{C}_f$ is a compact, self-adjoint operator on $L^2(-R,R)$. By the spectral theorem, it admits for every fixed value $0 < R < \infty$ a sequence $(\lambda_j^R, \Phi_j^R)_{j\geq 1}$ of real eigenvalues λ_j^R (accumulate only at zero), which are assumed to be enumerated in decreasing magnitude and repeated according to multiplicity, and a corresponding set of eigenfunctions Φ_j^R; to exlude trivial degeneracies, we shall assume throughout that *the sequence* $(\Phi_j^R)_{j\geq 1}$ *is a complete, orthonormal base of* $L^2(-R,R)$.

It follows from the Lipschitz-continuity (2.19) of $\mathrm{Cov}[f]$ and from the eigenvalue equation

$$(\mathcal{C}_f^R \Phi_j^R)(u) = \lambda_j^R \Phi_j^R(u)\ , \qquad |u| \leq R\ , \tag{2.22}$$

that $\Phi_j^R \in \mathrm{Lip}([-R,R];\mathbb{R}^d)$: for $u, u' \in [-R,R]$, there holds by Lemma 2.2 and by the eigenvalue equation (2.22)

$$\begin{aligned}
|\Phi_j^R(u) - \Phi_j^R(u')| &= \frac{1}{\lambda_j^R}\left| \int_{|v|<R} \left(\mathrm{Cov}[f](u,v) - \mathrm{Cov}[f](u',v)\right)\Phi_j^R(v)dv \right| \\
&\leq \frac{8RB(f)}{\lambda_j^R}|u-u'| \sup_{|v|<R}\left(\int_\Omega \left\|f(\omega;v) - \bar{f}(v)\right\|_2^2 d\mathbb{P}(\omega)\right)^{1/2} \left\|\Phi_j^R\right\|_{L^1(-R,R)} \\
&\leq \frac{8R^{3/2}B(f)}{\lambda_j^R}|u-u'| \sup_{|v|<R}\left(\int_\Omega \left\|f(\omega;v) - \bar{f}(v)\right\|_2^2 d\mathbb{P}(\omega)\right)^{1/2}\ .
\end{aligned}$$

Any bounded random flux $f(\omega;u)$ therefore admits, for every fixed $0 < R < \infty$, a Karhunen–Loève expansion

$$f(\omega;u) = \bar{f}(u) + \sum_{j\geq 1} Y_j^R(\omega)\Psi_j^R(u), \quad |u| \leq R\ , \tag{2.23}$$

which converges in $L^2(\Omega, d\mathbb{P}; L^2(-R,R)^d)$. In (2.23), the nominal flux $\bar{f}(u) = \mathbb{E}[f(\cdot;u)]$ and the sequence $(Y_j^R)_{j\geq 1}$ is a sequence of pairwise uncorrelated random variables given by

$$\forall j \in \mathbb{N}: \quad Y_j^R(\omega) := \sqrt{\lambda_j^R}\int_{|v|<R} f(\omega;v)\Phi_j^R(v)dv\ . \tag{2.24}$$

and the principal components of the random flux are given by

$$\forall j \in \mathbb{N}: \quad \Psi_j^R(u) := \frac{1}{\sqrt{\lambda_j^R}}\Phi_j^R(u)\ .$$

We remark that under suitable smoothness conditions on two-point correlation function $\mathrm{Cov}[f]$ of the random flux the convergence of the expansion (2.23) is a) pointwise with respect to u and b) the convergence rates increase with increasing smoothness of $\mathrm{Cov}[f]$ (see, e.g. [31]).

2.5. **Random entropy solutions of scalar conservation laws.** Equipped with the above notions of random initial data and random fluxes, we consider a random scalar conservation law (RSCL):

$$\begin{aligned}\partial_t u(x,t;\omega) + \sum_{j=1}^{d} \frac{\partial}{\partial x_j} f_j(\omega; u(x,t;\omega))) = 0 \ , \\ u(x,0;\omega) = u_0(x;\omega) \ , \quad x \in \mathbb{R}^d \ .\end{aligned} \tag{2.25}$$

and define,

Definition 2.3. *A random field $u : \Omega \ni \omega \to u(x,t;\omega)$, i.e. a measurable mapping from $(\Omega, \mathcal{F})$ to $C([0,T]; L^1(\mathbb{R}^d))$, is a* random entropy solution *of the SCL* (2.25) *with random initial data u_0 satisfying* (2.13) - (2.16) *for some $k \geq 2$ and with a* spatially homogeneous *random flux $f(\omega; u)$ as in Definition 2.1 that is statistically independent of u_0, if it satisfies the following conditions:*

(i.) Weak solution: for $\mathbb{P}$-a.e $\omega \in \Omega$, $u(\cdot,\cdot;\omega)$ satisfies the following integral identity, for all test functions $\varphi \in C_0^1(\mathbb{R}^d \times [0,\infty))$:

$$\int_0^\infty \int_{\mathbb{R}^d} \left(u(x,t;\omega)\varphi_t(x,t) + \sum_{j=1}^{d} f_j(\omega; u(x,t;\omega)) \frac{\partial}{\partial x_j}\varphi(x,t) \right) dxdt + \int_{\mathbb{R}^d} u_0(x,\omega)\varphi(x,0)dx = 0 \ . \tag{2.26}$$

(ii.) Entropy condition:

For any pair of (deterministic) entropy η and (stochastic) entropy flux $Q(\omega;\cdot)$ i.e η, Q_j with $j = 1,2,\ldots,d$ are functions such that η is convex and such that $Q_j'(\omega;\cdot) = \eta' f_j'(\omega;\cdot)$ for all j, and for $\mathbb{P}$-a.e $\omega \in \Omega$, u satisfies the following integral identity,

$$\int_0^\infty \int_{\mathbb{R}^d} \left(\eta(u(x,t;\omega))\varphi_t(x,t) + \sum_{j=1}^{d} Q_j(\omega; u(x,t;\omega)) \frac{\partial}{\partial x_j}\varphi(x,t) \right) dxdt \geq 0 \ , \tag{2.27}$$

for all deterministic test functions $0 \leq \varphi \in C_0^1(\mathbb{R}^d \times (0,\infty))$, $\mathbb{P}$-a.s.

Throughout what follows, we assume that the deterministic entropy function $\eta(\cdot)$ in (2.27) is a *Kružkov entropy*, i.e. $\eta(u) = |u - k|$ for some $k \in \mathbb{R}$.

In a forthcoming paper [31], we show the following well-posedness result for random entropy solutions of (2.25).

Theorem 2.4. *Consider the RSCL* (2.25) *with spatially homogeneous, bounded random flux $f : \Omega \to \mathrm{Lip}(\mathbb{R}; \mathbb{R}^d)$ as in Definition 2.1 and with (independent of f) random initial data $u_0 : \Omega \to L^1(\mathbb{R}^d)$ satisfying* (2.14), (2.15) *and the k-th moment condition* (2.16) *for some integer $k \geq 2$. In particular, then, there exists a constant $\bar{R} < \infty$ such that*

$$\|u_0(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)} \leq \bar{R} \qquad \mathbb{P} - a.e. \ \omega \in \Omega \ . \tag{2.28}$$

Assume moreover that the random flux admits the representation (2.23) *with* (2.24) *where the Lipschitz-continuous scaled flux components $\overline{\Psi_j^R}$ have Lipschitz constants B_j^R such that $\boldsymbol{B}^R := (B_j^R)_{j\geq 1} \in \ell^1(\mathbb{N})$ with some $R \geq \bar{R}$ as in* (2.28).

Then there exists a unique random entropy solution $u : \Omega \ni \omega \to C_b(0,T;L^1(\mathbb{R}^d))$ *which is "pathwise", i.e. for* $\mathbb{P}-a.e.\omega \in \Omega$*, described in terms of a nonlinear mapping* $S(\omega;t)$ *depending only on the random flux, such that*

$$u(\cdot,t;\omega) = S(\omega;t)u_0(\cdot,\omega)\ , \quad t>0,\ \mathbb{P}-a.e.\omega \in \Omega \tag{2.29}$$

such that for every $k \geq m \geq 1$ *and for every* $0 \leq t \leq T < \infty$ *holds*

$$\|u\|_{L^k(\Omega;C(0,T;L^1(\mathbb{R}^d)))} \leq \ \|u_0\|_{L^k(\Omega;L^1(\mathbb{R}^d))}\ , \tag{2.30}$$

$$\|S(\omega;t)\, u_0(\cdot,\omega)\|_{(L^1\cap L^\infty)(\mathbb{R}^d)} \leq \ \|u_0(\cdot,\omega)\|_{(L^1\cap L^\infty)(\mathbb{R}^d)} \tag{2.31}$$

and such that we have $\mathbb{P}$*-a.s.*

$$TV(S(\omega;t)u_0(\cdot,\omega)) \leq TV(u_0(\cdot,\omega))\ . \tag{2.32}$$

and, with $\bar{R}$ *as in* (2.28)*,*

$$\sup_{0\leq t\leq T} \|u(\cdot,t;\omega)\|_{L^\infty(\mathbb{R}^d)} \leq \bar{R} \quad \mathbb{P}-a.e.\ \omega \in \Omega\ . \tag{2.33}$$

Remark 2.5. The above theorem establishes that RSCLs are well-posed (in several space dimensions) for uncertain initial data as well as for random fluxes. An extension of these definitions and results to include random source terms $S(u,\omega)$ for bounded sources as well as to spatially inhomogeneous flux functions $f(\omega;x,u)$ is possible, provided their dependence on the spatial coordinate is Lipschitz: they are measurable mappings from $(\Omega,\mathcal{F})$ into $(E,\mathcal{B}(E))$ where $E = \mathrm{Lip}(\mathbb{R}^{d+1};\mathbb{R}^d)$.

Remark 2.6. So far, we considered the Karhunen–Loève expansion only for RSCLs (2.25), i.e. for the case $m=1$ in (1.1). It is straightforward to extend the principal component representation and the notion of covariance operator to flux functions for hyperbolic systems (1.1): in this case, the covariance operator is to be interpreted as abstract, symmetric bilinear form on the space $\mathrm{Lip}(\mathbb{R}^m;\mathbb{R}^{m\times d})$ whose kernel coincides with a symmetric, fourth order tensor function on the state space $\mathbb{R}^m$. Spectral decompositions analogous to (2.23) for $\mathbb{R}^{m\times d}$ matrix-valued random flux functions which arise in (1.1) can then be defined in an analogous fashion. However, due to the lack of bounds like (2.33), the approach of [31] can not be directly applied for the mathematical investigation of random hyperbolic systems (1.1) at present. Nevertheless, the spectral expansion (2.23) for (1.1) may be a useful tool to achieve a parsimonious parametric representation of a general, given random flux also in the numerical treatment of random hyperbolic systems (1.1).

The notions of random entropy solutions for a system of balance laws (1.1) with uncertain initial data, fluxes and sources can be analogously defined. Currently, there are no global well-posedness results for systems of balance laws. Hence, we are unable to extend the well-posedness results of Theorem 2.4 to the case of systems of balance laws such as (1.1).

3. Monte Carlo Finite Volume Method

Our aim is to approximate the random balance law (1.1). The spatio-temporal discretization can be performed by any standard Finite Volume or DG scheme, for instance (1.5). The probability space will be discretized using a statistical sampling technique. The simplest sampling method is the Monte Carlo (MC) algorithm consisting of the following three steps:

1. **Sample:** We draw M independent identically distributed (i.i.d.) initial data, flux and source samples $\{\mathbf{U}_0^k, \mathbf{F}_j^k, \mathbf{S}^k\}$ with $j = 1, \dots, d$ and $k = 1, \dots, M$ from the random fields $\{\mathbf{U}_0, \mathbf{F}_j, \mathbf{S}\}$ and approximate these by piece-wise constant cell averages.
2. **Solve:** For each realization $\{\mathbf{U}_0^k, \mathbf{F}_j^k, \mathbf{S}^k\}$, the underlying balance law (1.1) is solved numerically by the Finite Volume Method (1.5). We denote the FVM solutions by $\mathbf{U}_{\mathcal{T}}^{k,n}$, i.e. by cell averages $\{\mathbf{U}_C^{k,n} : C \in \mathcal{T}\}$ at time t^n,
$$\mathbf{U}_{\mathcal{T}}^{k,n}(\mathbf{x}) = \mathbf{U}_C^{k,n}, \quad \forall \mathbf{x} \in C,\ C \in \mathcal{T},\ k = 1, ..., M.$$
3. **Estimate Statistics:** We estimate the expectation of the random solution field with the sample mean (sample average) of the approximate solution:
$$E_M[\mathbf{U}_{\mathcal{T}}^n] := \frac{1}{M} \sum_{k=1}^{M} \mathbf{U}_{\mathcal{T}}^{k,n}. \tag{3.1}$$
Higher statistical moments can be approximated analogously under suitable statistical regularity of the underlying random entropy solutions [30].

The above algorithm is, at first sight, straightforward to implement. We remark that step 1 requires a (pseudo) random number generator. Here, care must be exercised in ensuring good statistical properties in massively parallel implementations (see Section 5 for details). In step 2, any standard (high-order) finite volume or DG scheme can be used. Hence, existing ("legacy") code for FVM (or DG) can be (re)used and there is no need to rewrite FVM (or DG) code. However, in doing so, particular care must be exercised that the "forward solver" thus employed is particularly robust: MC sampling will generate equistributed samples which cover the entire "senario space" of data, and include, in particular, also data instances which might appear "unlikely" or "unphysical" to experts. Nevertheless, such extremal scenarios do contribute to the approximate ensemble averages and must be resolved by the forward solver with accuracy versus CPU that is comparable to the accuracy of so-called "benchmark problems" commonly accepted as test for code performance in various CSE communities. "Legacy codes" whose performance might have been optimized on "benchmark problems" may lack such robustness for MC-generated scenarios which fall outside sets of commonly accepted benchmarks.

Furthermore, the only (data) interaction between different samples is in step 3 when ensemble averages are computed. Thus, the MC-FVM algorithms for UQ are non-intrusive as well as easily parallelizable.

Although a rigorous error estimate for the MCFVM approximating systems of balance laws appears to be currently out of reach, we rely on the analysis for a scalar conservation law in [30, 31] and on our computational experience with MLMC-FVM solution of systems of balance laws with random inputs in [32, 33] to postulate that the following estimate holds:

$$\|\mathbb{E}[\mathbf{U}(\cdot, t^n)] - E_M[\mathbf{U}_{\mathcal{T}}^n]\|_{L^2(\Omega; L^1(\mathbb{R}^d))} \leq C_{\mathrm{stat}} M^{-\frac{1}{2}} + C_{\mathrm{st}} \Delta x^s\ . \tag{3.2}$$

The positive constants $C_{\mathrm{stat}}, C_{\mathrm{st}}$ depend only on the second moments (must be *finite*) of the random initial data and of the source term in (1.1). In the above, we have assumed that the underlying Finite Volume (or DG) scheme converges to the solutions of the deterministic balance law (1.1) at rate $s > 0$. One rationale for adopting MLMC-FVM for the numerical solution of (1.1) with random data and fluxes lies in the fact that, in practice, even second or higher order schemes are known to realize only convergence rates $0 < s < 1$, due to the lack or regularity of the exact solutions. Therefore, the use of deterministic sampling strategies which promise a convergence rate that is higher than $1/2$ will, in general, not substantially improve the work versus accuracy of FVM for these problems.

Based on the error analysis of [30], we equilibrate the discretization and the sampling errors in the a-priori estimate (3.2) and choose [30, 32]

$$M = \mathcal{O}(\Delta x^{-2s}). \tag{3.3}$$

Next, we assume that the computational work (e.g. FLOPs, i.e. the number of required floating point operations) of the FVM solver for *one* complete run is given (asymptotically) by

$$\mathrm{Work}_{\mathrm{FVM}}(\Delta x) = C_s \Delta x^{-(d+1)}, \tag{3.4}$$

where $C_s > 0$ is independent of Δx and Δt, but depends on the order s of the FVM scheme.

Then, with the choice (3.3) and definition (3.4), it is straightforward to deduce that the asymptotic error vs. work estimate of the MC estimator (3.2) is given by

$$\|\mathbb{E}[u(\cdot, t^n)] - E_M[u^n_{\mathcal{T}}]\|_{L^2(\Omega;L^1(\mathbb{R}^d))} \lesssim (\mathrm{Work})^{-s/(d+1+2s)}. \tag{3.5}$$

The above error vs. work estimate is considerably more expensive when compared to the FVM discretization error for the corresponding deterministic problem [30]:

$$\|u(\cdot, t^n) - u^n_{\mathcal{T}}\|_{L^1(\mathbb{R}^d)} \leq C_{\mathrm{Err}} C_s^{s/(d+1)} \mathrm{Work}^{-s/(d+1)} =: C_{\mathrm{FVM}} \mathrm{Work}^{-s/(d+1)}. \tag{3.6}$$

4. Multi-level Monte Carlo Finite Volume Method

The low convergence rate (3.5) of MC-FVM motivates the use Multi-Level Monte Carlo Finite Volume Method (MLMC-FVM). The key idea behind the MLMC-FVM is to simultaneously sample a *hierarchy of discretizations of the PDE with random inputs with a level-dependent number of samples.* In the present setting, this amounts to running a deterministic FV solver on a sequence of nested grids in space with correspondingly adapted time step sizes, so as to ensure the validity of a CFL condition uniformly over all space-discretizations of the hierarchy [30].

4.1. **MLMC-FVM Error analysis.**

4.1.1. *MLMC-FVM algorithm.* The Multi Level Monte Carlo Finite Volume algorithm (MLMC-FVM for short) consists of the following four steps:

1. **Hierarchy of space-time discretizations:** Consider *nested* triangulations $\{\mathcal{T}_\ell\}_{\ell=0}^\infty$ of the spatial domain $\mathbf{D}$ with corresponding mesh widths Δx_ℓ that satisfy:

$$\Delta x_\ell = \Delta x(\mathcal{T}_\ell) = \sup\{\mathrm{diam}(K) : \ K \in \mathcal{T}_\ell\} = O(2^{-\ell} \Delta x_0), \quad \ell \in \mathbb{N}_0, \tag{4.1}$$

   where Δx_0 is the mesh width for the coarsest resolution and corresponds to the lowest level $\ell = 0$.
2. **Sample:** For each level of resolution $\ell \in \mathbb{N}_0$, we draw a *level-dependent number M_ℓ* of independent, identically distributed (i.i.d) samples from the input random fields $\{\mathbf{U}_0(\omega), \mathbf{F}_j(\omega), \mathbf{S}(\omega)\}$. Importantly, these random field inputs are only sampled on $\mathcal{T}_\ell$ in spatially discrete form, i.e. as realizations of cell-averages on triangulation $\mathcal{T}_\ell$, to yield *vectors* $\{\mathbf{U}_{0,\ell}(\omega), \mathbf{F}_{j,\ell}(\omega), \mathbf{S}_\ell(\omega)\}$.

on mesh $\mathcal{T}_\ell$. We index the level-dependent number M_ℓ of samples of these vectors by k, i.e. we write for $\ell = 0, 1, ...$

$$\{\mathbf{U}_{0,\ell}^k, \mathbf{F}_\ell^k, \mathbf{S}_\ell^k\}_{k=1}^{M_\ell} = \{\mathbf{U}_{0,\ell}(\omega^k), \mathbf{F}_{j,\ell}(\omega^k), \mathbf{S}_\ell(\omega^k) : k = 1, ..., M_\ell\}. \tag{4.2}$$

3. **Solve:** For each resolution level $\ell \in \mathbb{N}_0$ and for each realization of the random input data $\{\mathbf{U}_{0,\ell}^k, \mathbf{F}_{j,\ell}^k, \mathbf{S}_\ell^k\}$, $k = 1, ..., M_\ell$, the resulting *deterministic balance law* (1.1) (for this particular realization) is solved numerically by the Finite Volume Method (1.5) with mesh width Δx_ℓ. We denote the resulting ensemble of Finite Volume solutions by $\mathbf{U}_{\mathcal{T}_\ell}^{k,n}$, $k = 1, ..., M_\ell$. These constitute vectors of approximate cell averages, i.e. $\mathbf{U}_{\mathcal{T}_\ell}^{k,n} = \{\mathbf{U}_C^{k,n} : C \in \mathcal{T}_\ell\}$ of the corresponding realization of the random balance law at time level t^n and at spatial resolution level ℓ.
4. **Estimate solution statistics:** Fix some positive integer $L < \infty$ corresponding to the highest level. We estimate the expectation of the random solution field with the following estimator:

$$E^L[\mathbf{U}(\cdot, t^n)] := \sum_{\ell=0}^{L} E_{M_\ell}[\mathbf{U}_{\mathcal{T}_\ell}^n - \mathbf{U}_{\mathcal{T}_{\ell-1}}^n], \tag{4.3}$$

with E_{M_ℓ} being the MC estimator defined in (3.1) for the mesh level ℓ.

Remark 4.1. In the present article, we assume (for ease of exposition only) the sequence of triangulations $\{\mathcal{T}_\ell\}_{\ell=0}^\infty$ to be nested. This assumption was also made in the proofs of [30]. We emphasize here that an inspection of the arguments in [30] reveals that *the nestedness assumption on the meshes is not essential for the error bounds to hold.* However, in order to execute Step 4. (estimate solution statistics), in the case that the grid hierarchy is non-nested, an efficient intergrid transfer resp. prolongation must be available. This is often the case, when *multilevel discretizations* have been employed in the deterministic solver which is used for the discrete solutions.

Remark 4.2. Higher statistical moments (2.12) of the random entropy solution can be approximated analogously (see, e.g., the sparse tensor discretization of [30]). An additional, new issue arises in the efficient numerical computation of *space-time correlation functions* due to the high-dimensionality of such correlation functions.

The MLMC-FVM is *non-intrusive* as any standard FVM (or DG) code can be used in step 3. Furthermore, MLMC-FVM is amenable to *efficient parallelization* as data from different grid resolutions and different samples only interacts in step 4. We refer to [42] and to Section 5 for details.

4.1.2. *MLMC-FVM error bounds.* Again, based on the rigorous estimate for scalar conservation laws in [30, 31] and on our experience for systems of balance laws [32, 33], we postulate the following error estimate:

$$\|\mathbb{E}[u(\cdot, t^n)] - E^L[u(\cdot, t^n)]\|_{L^2(\Omega; L^1(\mathbb{R}^d))} \leq C_1 \Delta x_L^s + C_2 \Big\{ \sum_{\ell=0}^{L} M_\ell^{-\frac{1}{2}} \Delta x_\ell^s \Big\} + C_3 M_0^{-\frac{1}{2}}. \tag{4.4}$$

Here, the parameter $s > 0$ refers to the convergence rate of the finite volume scheme for the deterministic problem and $C_{1,2,3}$ are positive constants depending only on the second moments of the initial data and the source term.

From the error estimate (4.4), we obtain that the number of samples to equilibrate the statistical and spatio-temporal discretization errors in (4.3) is given by

$$M_\ell = \mathcal{O}(2^{2(L-\ell)s}) \ , \quad \ell = 0, 1, ..., L \ . \tag{4.5}$$

Notice that (4.5) implies that the largest number of MC samples is required on the coarsest mesh level $\ell = 0$, whereas only a small fixed number $M_L \in \mathbb{N}$ of MC samples are needed on the finest discretization levels. To achieve this, we choose

$$M_\ell = M_L 2^{2(L-\ell)s}. \tag{4.6}$$

Using choice (4.5) for M_ℓ, the error bound (4.4) reduces to

$$\|\mathbb{E}[u(\cdot, t^n)] - E^L[u(\cdot, t^n)]\|_{L^2(\Omega;L^1(\mathbb{R}^d))} \leq C_{\mathrm{Err}} \Delta x_L^s (L+1), \tag{4.7}$$

with positive constant C_{Err} independent of $\Delta x, \Delta t$ and L.

MLMC-FVM work estimates. Total amount of work required for the complete MLMC-FVM estimator (4.3) can be estimated using (4.6) and (3.4):

$$\begin{aligned}
\mathrm{Work}_{\mathrm{MLMC}}(\Delta x_L, L) &= \sum_{\ell=0}^{L} M_\ell \mathrm{Work}_{\mathrm{FVM}}(\Delta x_\ell) \\
&= \sum_{\ell=0}^{L} M_L 2^{2s(L-\ell)} C_s \Delta x_\ell^{-(d+1)} \\
&= \sum_{\ell=0}^{L} M_L 2^{2s(L-\ell)} C_s 2^{-(d+1)(L-\ell)} \Delta x_L^{-(d+1)} \\
&= M_L C_s \Delta x_L^{-(d+1)} 2^{\alpha L} \sum_{\ell=0}^{L} 2^{\alpha \ell},
\end{aligned} \tag{4.8}$$

where $\alpha := 2s - (d+1)$. Next, we bound the sum in (4.8) using geometric series,

$$\sum_{\ell=0}^{L} 2^{-\alpha\ell} \leq \begin{cases} \frac{2^{-\alpha L}}{1-2^{\alpha}} & \alpha < 0, \\ L+1 & \alpha = 0, \\ \frac{1}{1-2^{-\alpha}} & \alpha > 0 \ . \end{cases} \tag{4.9}$$

MLMC-FVM error vs. work bounds. Combining error estimate (4.7) with work estimates (4.8) - (4.9), we obtain (see [32]) the error vs. work estimate for (4.3),

$$\left\|\mathbb{E}[u] - E^L[u]\right\|_{L^2(\Omega;L^1(\mathbb{R}^d))} \leq \begin{cases} C_{\mathrm{MLMC}} \cdot \mathrm{Work}^{-s/(d+1)} \cdot \log(\mathrm{Work}) & s < (d+1)/2, \\ C_{\mathrm{MLMC}} \cdot \mathrm{Work}^{-1/2} \cdot \log{(\mathrm{Work})}^{3/2} & s = (d+1)/2, \\ C_{\mathrm{MLMC}} \cdot \mathrm{Work}^{-1/2} \cdot \log(\mathrm{Work}) & s > (d+1)/2. \end{cases} \tag{4.10}$$

with constant C_{MLMC} given by (disregarding constants from the logarithm base)

$$C_{\mathrm{MLMC}} = C_{\mathrm{Err}} (M_L C_s)^{s/(d+1)} \cdot \begin{cases} (1-2^{\alpha})^{-s/(d+1)} & s < (d+1)/2, \\ 1 & s = (d+1)/2, \\ (1-2^{-\alpha})^{-s/(d+1)} & s > (d+1)/2. \end{cases} \tag{4.11}$$

For usual convergence rates, i.e. $s \geq 1/2$, $s \in \mathbb{N}/2$, constant C_{MLMC} is bounded by

$$C_{\mathrm{MLMC}} \leq (2M_L)^{s/(d+1)} \cdot C_{\mathrm{Err}} C_s^{s/(d+1)}. \tag{4.12}$$

In the numerical experiments of subsection 6.7, constants C_{Err} in (3.6) and (4.7) have almost equal numerical value. Hence constant C_{MLMC} can be bounded in terms of C_{FVM},

$$C_{\text{MLMC}} \leq (2M_L)^{s/(d+1)} \cdot C_{\text{FVM}} \leq \sqrt{2M_L} \cdot C_{\text{FVM}}, \quad \text{for } s \leq (d+1)/2. \tag{4.13}$$

The error estimates in (4.10) show that the MLMC-FVM is superior to the MC-FVM as the asymptotic computational cost for MLMC-FVM scales as $\text{Work}^{-s/(d+1)}$ (disregarding logarithmic term, see [32]); compare to $\text{Work}^{-s/(d+1+2s)}$ for the MC-FVM scheme as in (3.5). Hence, the MLMC-FVM is expected to be (asymptotically) considerably faster than the MC-FVM for the same magnitude of error.

Notice, that for $s > (d+1)/2$, the asymptotic error convergence rate in (3.5) no longer increases as it is limited by the convergence rate of Monte Carlo sampling which equals $1/2$. Moreover, the constant C_{MLMC} also *increases* with s, see (4.11). However, such high ($s > 2$) convergence rates of the *deterministic* solver can only be observed in very special cases, where the pathwise regularity of the stochastic solution is high (i.e. in the absence of shocks such as, for example, the Ringleb flow).

Furthermore, if $s \leq (d+1)/2$ then the error vs. work estimate (4.10) is almost (i.e. up to a logarithmic term) of the same order as the error vs. work of the deterministic finite volume scheme which implies that the total amount of work to achieve a certain error level ε, say, in approximation of the random entropy solution's mean field will, asymptotically, be equal to that of approximating the entropy solution of one deterministic balance law at the same level L of resolution.

In addition to *optimal* asymptotic convergence rates of the MLMC-FVM solver, bounds (4.13) provide estimates on how much more work is required for *stochastic* version of the simulation compared to its *deterministic* version. The difference in constants is comparably small, i.e. the *stochastic* simulation is *at most* $\sqrt{2M_L}$ times less accurate (disregarding logarithmic terms) compared to its *deterministic* version, where the free parameter M_L is usually chosen to be small, i.e. $\mathcal{O}(1) - \mathcal{O}(10)$.

Remark 4.3. *Notice, that the standard MC-FVM estimator* (3.1) *is monotonic, i.e. mean estimator of* positive *quantities (such as pressure, density) is also* positive*. However, the MLMC-FVM estimator* (4.3) *is not monotonic, i.e. negative undershoots might occur, which are* not *related to numerical instability, bias of the estimators, inappropriate FVM schemes or other implementation errors. The awareness of this* non-monotonicity *of the MLMC-FVM estimator is essential for the interpretation of the numerical results. More detailed explanation and examples are provided in Remark 7.1.*

4.2. **Sparse Tensor Approximations of k-point correlations.** We now consider the efficient MLMC-FVM approximation of two and of k-point correlation functions of random entropy solutions of the system (1.1). Throughout, we consider only the scalar case, i.e. $m = 1$ in (1.1), in order to simplyfy the notation. All concepts and methods have (tensor) analogues in the case of hyperbolic systems.

Throughout this section, we assume that there exists a unique random entropy solution of (1.1) which satisfies, for a given order $k \geq 1$ of correlation of interest,

$$u(\cdot, t; \omega) \in L^{2k}(\Omega; C^0([0,T]; W^{s,1}(D))) , \quad \text{for some} \quad 0 < s \leq 1 . \tag{4.14}$$

I.e., we assume that the function is integrable to the power $2k$ if $k \geq 1$ is the order of the moment of interest, and admits, as a function of the spatial variable, a fractional deriative of order s which belongs to the space $L^1(D)$ where $D \subset \mathbb{R}^d$ denotes the computational domain.

For a mesh hierarchy $\{\mathcal{T}_\ell\}_{\ell=0}^{\infty}$ in D, we define the space S_ℓ of simple, i.e. piecewise constant functions on $\mathcal{T}_\ell$, and the associated projector by

$$S_\ell := S(\mathcal{T}_\ell), \quad P_\ell := P_{\mathcal{T}_\ell} : L^1(\boldsymbol{D}) \mapsto S_\ell, \qquad \ell \geq 0\,. \tag{4.15}$$

Here, for a given triangulation $\mathcal{T}$, $P_{\mathcal{T}}$ denotes the operator which associates to a function $v \in L^1(D)$ the piecewise constant function of cell-averages $P_{\mathcal{T}}v \in S(\mathcal{T})$.

Then the MC-FVM approximations of $\mathcal{M}^k(u(\cdot,t))$ are defined as statistical estimates from the ensemble

$$\{\widehat{v}^i_{\mathcal{T}}(\cdot,t)\}_{i=1}^M \tag{4.16}$$

obtained by from samples of the RSCL: specifically, the first moment of the random solution $u(\cdot,t;\omega)$ at time $t>0$, is estimated as

$$\mathcal{M}^1(u(\cdot,t)) \approx E_M[v_{\mathcal{T}}(\cdot,t)] := \frac{1}{M}\sum_{i=1}^M \widehat{v}^i_{\mathcal{T}}(\cdot,t)\,, \tag{4.17}$$

and, for $k>1$, the kth moment (or k point correlation function) $\mathcal{M}^k(u(\cdot,t)) = \mathbb{E}[(u(\cdot,t))^{(k)}]$ defined in (2.12) is estimated by

$$E^{(k)}_M[v_{\mathcal{T}}(\cdot,t)] := \frac{1}{M}\sum_{i=1}^M \underbrace{(\widehat{v}^i_{\mathcal{T}} \otimes \cdots \otimes \widehat{v}^i_{\mathcal{T}})}_{k-\text{times}}(\cdot,t)\,. \tag{4.18}$$

More generally, for $k>1$, we consider time instances $t_1,\dots,t_k \in (0,T]$, $T<\infty$, and define the statistical FVM estimate of $\mathcal{M}^k(u)(t_1,...,t_k)$ by

$$E^{(k)}_M\left[v_{\mathcal{T}}\right](t_1,\dots,t_k) := \frac{1}{M}\sum_{i=1}^M \underbrace{(\widehat{v}^i_{\mathcal{T}}(\cdot,t_1) \otimes \cdots \otimes \widehat{v}^i_{\mathcal{T}}(\cdot,t_k))}_{k-\text{times}}\,. \tag{4.19}$$

The work to form a single tensor product in the ensemble average (4.19) over a finite spatial domain $D \subset \mathbb{R}^d$ grows as $O(\Delta x^{-kd})$ which, in general, entails a computational effort that is, for moment orders $k \geq 2$, prohibitive. To reduce the complexity of k-th moment estimation, we introduce in the following a *compressive approximation* of two- and of k-point correlation functions that is similar to the strategy for high order moment approximation in elliptic problems with random data in [44, 5].

4.2.1. *Sparse Tensorization of FV Solutions.* Since the cell-average projections $P_\ell : L^1(\mathbb{R}^d) \to S_\ell$ in (4.15) are onto, we may define the linear space of *increment* or *details* of FV functions between successive meshes in the grid hierarchy $\{\mathcal{T}_\ell\}_{\ell=0}^{\infty}$ by

$$W_\ell := (P_\ell - P_{\ell-1})S_\ell, \qquad \ell \geq 0, \tag{4.20}$$

where $P_{-1} := 0$ so that $W_0 = S_0$. Typically, this increment space is spanned by the so-called *Haar-wavelets* on $\mathcal{T}_\ell$; however, in what follows we aim at retaining the non-intrusive nature of the MLMC-FVM and therefore we will *never explicitly assume redesign of the FV solvers in terms of multiresolution analysis on the triangulations* $\mathcal{T}$. We rather propose an algorithm which is based only on the piecewise constant approximations at each timestep, and on recursively identifying the projections of the FV solution onto the increment or detail space W_ℓ in (4.20) “on the fly”, by the so-called pyramid scheme. Such schemes are available for approximations in structured meshes, but moreover can also be developed for unstructured grids (we refer to [37, Chapter 2], [38] for details).

With (4.20), for any $L \in \mathbb{N}_0$, we have the multilevel decomposition

$$S_L = W_0 \oplus ... \oplus W_L = \bigoplus_{\ell=0}^{L} W_\ell \tag{4.21}$$

and the k-point correlation functions $(v_L(\cdot,t))^{(k)}$ of the FV solutions on mesh $\mathcal{T}_L$ at time $t > 0$ take values in the tensor product space

$$(S_L)^{(k)} := \underbrace{S_L \otimes ... \otimes S_L}_{k \text{ times}} = \sum_{|\vec{\ell}|_\infty \le L} S_{\ell_1} \otimes ... \otimes S_{\ell_k} = \bigoplus_{|\vec{\ell}|_\infty \le L} \bigotimes_{j=1}^{k} W_{\ell_j} \; . \tag{4.22}$$

Then, the full tensor projections

$$P_L^{(k)} v := \underbrace{P_L \otimes ... \otimes P_L}_{k \text{ times}} : L^1(\mathbb{R}^{kd}) \to (S_L)^{(k)} \tag{4.23}$$

are bounded, linear and onto. Here, $|\vec{\ell}|_\infty := \max\{\ell_1, ..., \ell_k\}$ and the last sum in (4.22) is a direct one. Obviously, if $N_L := \dim S_L < \infty$ (as is the case when, e.g., the spaces S_ℓ are only defined on a bounded domain $D \subset \mathbb{R}^d$) then $\dim((S_L)^{(k)}) = N_L^k$ which is prohibitive. Sparse Tensor approximations of k-point correlations $(v(\cdot,t))^{(k)}$ will be approximations in tensor products of spaces of piecewise constant functions on meshes on coarser levels which are defined similar to (4.22) by

$$\widehat{(S_L)}^{(k)} := \bigoplus_{|\vec{\ell}|_1 \le L} \bigotimes_{j=1}^{k} W_{\ell_j} \tag{4.24}$$

where now $|\vec{\ell}|_1 := \ell_1 + ... + \ell_k$. If the mesh family $\{\mathcal{T}_\ell\}_{\ell=0}^\infty$ is generated by recursive dyadic refinements of the initial triangulation $\mathcal{T}_0$, when $N_L = \dim S_L < \infty$ (as is the case e.g. on bounded domains $D \subset \mathbb{R}^d$) it holds

$$\dim \widehat{(S_L)}^{(k)} = O(N_L (\log_2 N_L)^{k-1}) \; . \tag{4.25}$$

With $\widehat{(S_L)}^{(k)}$ in (4.24), we also define the *sparse tensor projection*

$$\widehat{(P_L)}^{(k)} := \bigoplus_{|\vec{\ell}|_1 \le L} \bigotimes_{j=1}^{k} (P_{\ell_j} - P_{\ell_j - 1}) : L^1(\mathbb{R}^{kd}) \to \widehat{(S_L)}^{(k)} \; . \tag{4.26}$$

The approximation properties of the sparse tensor projection are as follows (cf. the Appendix of [30]): for any function $U(x_1, ..., x_k)$ which belongs to $(W^{s,1}(\mathbb{R}^d))^{(k)}$, it holds

$$\|U - \widehat{(P_L)}^{(k)} U\|_{L^1(\mathbb{R}^{kd})} \le C(\Delta x_L)^s |\log \Delta x_L|^{k-1} \|U\|_{(W^{s,1}(\mathbb{R}^d))^{(k)}} , \tag{4.27}$$

where $C > 0$ depends only on k, d and on the shape regularity of the family $\{\mathcal{T}_\ell\}_{\ell \ge 0}$ of triangulations, but is independent of Δx.

4.2.2. *Definition of the Sparse Tensor MLMC-FVM Estimator.* With the above notions in hand, we proceed to the *definition of the sparse tensor MLMC-FVM estimator of* $\mathcal{M}^{(k)}(u(\cdot,t))$. To this end, we modify the full tensor product MLMC-FV estimator which is based on forming the k-fold tensor products of the FV solution in the spaces $S_L^{(k)}$ in (4.22), as follows (recall from (4.17) that $E_M[\cdot]$ denotes the MC estimate based on M samples): for a given sequence $\{M_\ell\}_{\ell=0}^L$ of MC samples at level ℓ, the sparse tensor MLMC estimate of $\mathcal{M}^k[u(\cdot,t)]$ is, for $0 < t < \infty$, defined by

$$\widehat{E^{L,(k)}}[u(\cdot,t)] := \sum_{\ell=0}^{L} E_{M_\ell}[\widehat{P_\ell}^{(k)}(v_\ell(\cdot,t))^{(k)} - \widehat{P_{\ell-1}}^{(k)}(v_{\ell-1}(\cdot,t))^{(k)}]\ . \tag{4.28}$$

We observe that (4.28) is identical to the full tensor product formation of the Finite Volume solution if the sparse projectors $\widehat{P_\ell^{(k)}}$ in (4.28) are replaced by the full tensor projections $P_\ell^{(k)}$, except for the sparse formation of the k-point correlation functions of the FV solutions corresponding to the initial data samples $\hat{u}_0^i$. In bounded domains, this reduces the work for the formation of the k-point correlation function from N_L^k to $O(N_L(\log_2 N_L)^{k-1})$ per sample at mesh level L. As our convergence analysis ahead will show, use of sparse rather than full tensor products will not entail any reduction in the order of convergence of the k-th moment estimates.

4.2.3. *Error and Complexity Analysis of the Sparse Tensor MLMC-FVM.* The following, basic result on the complexity of the MLMC-FVM proved in [30] shows that *MLMC-FVM estimates of two- and of k-point correlations of the random entropy solutions are possible in log-linear complexity of one single, deterministic solve on the finest mesh level L.*

Theorem 4.4. *Assume the regularity (4.14). Assume further that we are given a FVM such that* (1.4) *holds and such that the deterministic FVM scheme converges at rate* $s > 0$ *in* $L^\infty([0,\infty]; L^1(\mathbb{R}^d))$. *Then the MLMC-FVM estimate* $\widehat{E^{L,(k)}}[u(\cdot,t)]$ *defined in (4.28) satisfies, for every sequence* $\{M_\ell\}_{\ell=0}^L$ *of MC samples, the error bound*

$$\|\mathcal{M}^k u(\cdot,t) - \widehat{E^{L,(k)}}[u(\cdot,t;\omega)]\|_{L^2(\Omega;L^1(\mathbb{R}^{kd}))}$$

$$\lesssim (1\vee t)\Delta x_L^s |\log \Delta x_L|^{k-1}\left\{\|\mathrm{TV}(u_0(\cdot,\omega))\|_{L^k(\Omega;d\mathbb{P})}^k + \|u_0(\cdot\,;\omega)\|_{L^\infty(\Omega;W^{s,1}(\mathbb{R}^d))}^k\right\}$$

$$+\left\{\sum_{\ell=0}^{L}\frac{\Delta x_\ell^s |\log \Delta x_\ell|^{k-1}}{M_\ell^{1/2}}\right\}\left\{\|u_0(\cdot\,;\omega)\|_{L^{2k}(\Omega;W^{s,1}(\mathbb{R}^d))}^k + t\|\mathrm{TV}(u_0(\cdot\,;\omega))\|_{L^{2k}(\Omega;d\mathbb{P})}^k\right\}\ .$$

The total work to compute the MLMC estimates $\widehat{E^{L,(k)}}[u(\cdot\,;t)]$ *on compact domains* $D \subset \mathbb{R}^d$ *is therefore (with* $O(\cdot)$ *depending on the size of* D*)*

$$\widehat{\mathrm{Work}}_L^{MLMC} = O\left(\sum_{\ell=0}^{L} M_\ell \Delta x_\ell^{-(d+1)} |\log \Delta x|^{k-1}\right)\ . \tag{4.29}$$

Based on Theorem 4.4, we infer that the choice of sample sizes M_ℓ at level ℓ should also be used in the MLMC-FVM estimation of k-point correlation functions of order $k > 1$ of the random entropy solution, provided the order s of the underlying deterministic FVM scheme is at most 1 (see [30] for details). Due to the linear complexity of the pyramid scheme, the conversion of the FVM approximations of the draws $\hat{u}^i(\cdot\,,t;\omega)$ of the random entropy solution at time $t>0$ into a multilevel representation and the sparse tensor product formation in the MLMC estimator (4.28) increases the work bounds for the first moments only by a logarithmic factor, so that, in terms of the computational work, we have with the choices (4.5) of MC samples M_ℓ, the following error bound in terms of work in a bounded computational domain $D \subset \mathbb{R}^d$:

$$\|\mathcal{M}^k u(\cdot,t) - \widehat{E^{L,(k)}}[u(\cdot,t;\omega)]\|_{L^2(\Omega;L^1(D^k))} \leq C(\widehat{\mathrm{Work}_L}^{MLMC})^{-s'/(d+1)} \tag{4.30}$$

for any $0<s'<s$ with the constant depending on D and growing as $0<s'\to s\leq 1$.

5. Efficient Implementation of MLMC-FVM

As stated in the previous section, the MLMC-FVM algorithm has four stages. We discuss implementation issues that arise in each stage below.

5.1. Step 1: Hierarchy of nested grids.

We will solve systems of balance laws (1.1) in one and two space dimensions. In the numerical results which are reported in Section 6 ahead, in two space dimensions, we will choose Cartesian meshes for simplicity. It is relatively straightforward to choose any hierarchy of nested grids consisting of either triangles/ tetrahedra or quad/hexahedral volume in either one, two or three space dimensions.

5.2. Step 2: Sample.

In this step, we have to draw M_ℓ i.i.d. samples for the initial data and source random fields $\mathbf{U}_0, \mathbf{F}_j, \mathbf{S}$ corresponding to the underlying probability distribution. Standard random number generators (RNG) can be readily used to draw such samples. For the serial implementation, any reasonable RNG works well in practice. However, random number generation becomes a very sensitive part of Monte Carlo type algorithms on massively parallel architectures. Inconsistent seeding and insufficient period length of the RNG might cause correlations in presumably i.i.d. draws which might potentially lead to biased solutions, see [32]. We used the `WELL`-series of pseudo random number generators from [24, 23]. These generators have been designed with particular attention towards large periods and good equidistribution. To deal with the seeding issues, we *injectively* map the *unique* rank of each core (in a parallel algorithm) to some corresponding element in the hardcoded array of prime numbers (henceforth, the array of seeds), see [42] for detailed explanations. In this way statistical independence is preserved. For all numerical experiments reported here, the RNG `WELL512a` was used. We found `WELL512a` to have a sufficiently large period $2^{512}-1$ and to be reasonably efficient (33 CPU sec for 10^9 draws). We emphasize that there are plenty of alternatives to `WELL512a` with even longer periods (which, however, use more memory than `WELL512a`). To name a few: `WELL1024a` with period $2^{1024}-1$, takes 34 sec and `WELLRNG44497` with period $2^{44497}-1$ which takes 41 sec to generate 10^9 draws; yet another alternative could be "Marsenne Twister" [29] - it has the same period of $2^{19937}-1$ as its counterpart `WELLRNG19937`. Within our numerical experiments, we did not observe any *qualitative* difference between the described RNGs.

5.3. Step 3: Solve.

For each realization of the random input fields, we need to solve (1.1) for each realization of the random input with a finite volume or DG scheme.

5.3.1. *General Consideration.* As the solve step in the MLMC-FVM algorithm will be repeated for a large number of data samples on different space-time resolution levels, a robust and efficient FVM code for systems of balance laws. We recall that the MC and MLMC methods rely on *equidistribution of samples in the data space*. Therefore, in MC-FVM and MLMC-FVM, also extreme (i.e. “improbable” or physically practically impossible in the eyes of experts) data scenarios are generated, and care must be taken that the numerical solver for these scenarios is of comparable efficiency as for the more standard, “benchmark” cases. *Robustness of the numerical solver* is therefore a key issue in the development and deployment of MLMC techniques for nonlinear PDEs. In our large scale numerical experiments, we choose the code named `ALSVID` [1] that was designed by researchers at CMA, University of Oslo and SAM, ETH Zürich. Based on this platform, we developed a version called `ALSVID-UQ` which is specifically tailored to UQ for hyperbolic systems of conservation and balance laws, and the complete source code of which is publicly available for download under [2]. As `ALSVID` is extensively used in the examples of this paper, we describe it briefly below.

5.3.2. *ALSVID.* This finite volume code approximates the Shallow-water, Euler equations and MHD equations in one, two and (in the last two cases) in three space dimensions. It is based on the following ingredients:

1. **Approximate Riemann solver:** The numerical fluxes in the Finite Volume Scheme (1.5) used in `ALSVID` are based on approximate Riemann solvers of the HLL type for the Euler and MHD equations [16]. For shallow-water equations with bottom topography, the energy stable well-balanced schemes of [15] have been implemented in `ALSVID-UQ`.
2. **Divergence constraint.** The divergence constraint in the MHD equations [16] is handled in `ALSVID` by adding the Godunov-Powell source term to the MHD equations. This source term is proportional to the divergence and allows divergence errors to be swept out of the domain. Numerical stability can only be ensured by a careful *upwinding* of the source term, see [16].
3. **Non-oscillatory reconstructions.** `ALSVID` employs a variety of piecewise polynomial non-oscillatory reconstruction procedures for attaining high order of spatial accuracy. In particular, second order ENO and WENO procedures are employed, see Section 2 of [16]. However, these procedures need to be modified in order to preserve positivity of the density and pressure. Such modifications are described in Section 2 of [16].
4. **Time stepping.** High-order accurate time stepping procedures of the SSP Runge-Kutta [20] are employed in `ALSVID`.

Fluxes on the boundary of the computational domain are defined using so-called ghost cells, see, for example, Chapter 10 in [25]. The FV solver library `ALSVID` uses a modular structure in `C++` with a `Python` front end for pre- and post-processing. One and two dimensional visualizations are performed with `MatPlotLib` and three dimensional data sets are visualized using `MayaVi2`. Extensive testing of `ALSVID` has been performed and reported in [16].

A massively parallel version of `ALSVID` had already been developed for deterministic problems; we refer to [1] for further details. The parallelization paradigm for `ALSVID` is based on domain decomposition using Message Passing Interface (`MPI`) standard and its particular implementation `OpenMPI`.

5.4. **Stable computation of sample statistics.** For both MC-FVM and MLMC-FVM algorithms, we need to combine ensembles of individual realizations of numerical solutions for the statistical estimation of ensemble averages.

5.4.1. *Discussion of round-off effects.* It is straightforward to evaluate the sample mean for the MC-FVM and the estimator (4.3) for MLMC-FVM. A straightforward algorithm to compute an *unbiased* estimate of the variance for scalar $u = u(x,t)$ with fixed x, t is the following statistical estimator $\mathrm{Var}_M[u]$ of the variance $\mathrm{Var}[u] := \mathbb{E}[u^2] - \mathbb{E}[u]^2$:

$$\mathrm{Var}_M[u] := \frac{1}{M-1}\sum_{i=1}^{M}(u^i)^2 - \frac{1}{M(M-1)}\left(\sum_{i=1}^{M} u^i\right)^2 , \tag{5.1}$$

where u^i are MC-FVM samples. This way, it suffices to loop over all samples only once; unfortunately, both quantities are almost equal in regions of nearly vanishing variance. This is typically the case in smooth regions of the flow, i.e. outside of shocks and of viscous boundary layers. We observed in our numerical experiments that the straightforward use of the estimator (5.1) is prone to *subtractive cancellation* and *loss of accuracy* in finite (IEEE double) precision floating point arithmetic. This problem is well-known, and in [46], the authors propose an alternative *stable* "on-line" variance computation algorithm:

Set $\bar{u}^0 = 0$ and $\Phi^0 = 0$; then proceed iteratively:

$$\bar{u}^i \;=\; \sum_{j=1}^{i} u^j/i, \tag{5.2}$$

$$\Phi^i \;:=\; \sum_{j=1}^{i}(u^j - \bar{u}^i)^2 = \Phi^{i-1} + (u^i - \bar{u}^i)(u^i - \bar{u}^{i-1}) \;. \tag{5.3}$$

Then, unbiased mean and variance estimates are given by:

$$E_M[u] = \bar{u}^M, \quad \mathrm{Var}_M[u] = \Phi^M/(M-1). \tag{5.4}$$

Although identical in exact arithmetic, the above algorithm can deal with small cancellation errors. Note, however, that the estimators (5.1) and (5.4) will only converge under the provision that *fourth statistical moments of the random solution, evaluated pointwise in space and time, are finite*; an alternative approach which does not require pointwise finite fourth moments proceeds via estimating the space-time two-point correlation functions of the random entropy solutions, see [30]. For this approach, finite fourth moments of u are also required, albeit with values in $(C_b([0,\ T], L^1(\mathbb{R}^d)))^m$.

5.4.2. *Efficient parallelization.* The key issue in the parallel implementation of the whole algorithm (the solve steps) is to distribute computational work evenly among the cores. Without going into the details, we refer the reader to the novel static load balancing strategy on homogeneous parallel architectures (i.e. all cores are assumed to have identical CPUs and RAM per node, and equal bandwidth and latency to all other cores) in recent papers [42, 32].

6. Performance Studies of the MLMC-FVM for Conservation Laws

In this section, we will test the MLMC-FVM algorithm, presented in the previous section, and demonstrate its robustness and efficiency. We run numerical tests for five different problem sets: two of them will consider a multi-dimensional system of conservation laws with uncertain initial data, one will consider a system of balance laws with random bottom topography (source term) and the remaining three numerical experiments will address the performance of our Multi-Level MC-FVM for conservation laws with random fluxes.

Recalling that the discretization of the random conservation law involves discretizing in space-time with a standard Finite Volume Method and the discretizing the probability space with a statistical sampling method, we tabulate various combinations of methods that are to be tested:

MC	Monte Carlo with 1st order FVM scheme	$M = \mathcal{O}(\Delta x^{-1})$,
MC2	Monte Carlo with 2nd order FVM scheme	$M = \mathcal{O}(\Delta x^{-2})$,
MLMC	multilevel MC with 1st order FVM scheme	$M_\ell = M_L 2^{(L-\ell)}$,
MLMC2	multilevel MC with 2nd order FVM scheme	$M_\ell = M_L 4^{(L-\ell)}$.

Furthermore, we need to following parameters, which will be specified for every simulation in the form of a table below the corresponding figure:

Parameter	**Description**
L	number of hierarchical mesh levels
M_L	number of samples at the finest mesh level
grid size	number of cells in X and in Y directions
CFL	CFL number based on the fastest wave
cores	total number of cores used in the simulation
runtime	clock-time (serial runs) or wall-time (parallel runs); hrs:min:sec
efficiency	MPI efficiency, as defined in [42].

As we will present numerical convergence analysis results, we need to specify the following error estimator.

Error estimator. Since the solution is a random field, the discretization error is a random quantity as well. For our computational convergence analysis we therefore compute a statistical estimator by averaging estimated discretization errors from several independent runs. We will compute the error in (3.2) by approximating $L^2(\Omega; L^1(\mathbb{R}^d))$ norm with MC quadrature. Let U_{ref} denote the reference solution and $\{U_k\}_{k=1,\dots,K}$ be a sequence of independent approximate solutions obtained by running MC-FVM or MLMC-FVM solver K times corresponding to K realizations of the stochastic space. Then the $L^2(\Omega; L^1(\mathbb{R}^d))$-based relative error estimator is defined as in [30],

$$\mathcal{R}E = \sqrt{\sum_{k=1}^{K} (\mathcal{R}E_k)^2 / K}, \tag{6.1}$$

where:

$$\mathcal{R}E_k = 100 \times \frac{\|U_{\text{ref}} - U_k\|_{\ell^1(\mathcal{T})}}{\|U_{\text{ref}}\|_{\ell^1(\mathcal{T})}} . \tag{6.2}$$

The extensive analysis for the appropriate choice of K is conducted in [30]; unless indicated otherwise, we choose $K = 30$ which proved to be sufficient in our numerical experiments to remove statistical fluctuations in the convergence plots.

Equipped with the above notation and concepts, we present the following six numerical experiments.

6.1. **Euler equations with uncertain initial data.** The Euler equations of gas dynamics are

$$\begin{cases} \rho_t + \operatorname{div}(\rho\mathbf{u}) = 0, \\ (\rho\mathbf{u})_t + \operatorname{div}(\rho\mathbf{u}\otimes\mathbf{u} + p\mathbf{ID}) = 0, \\ E_t + \operatorname{div}((E+p)\mathbf{u}) = 0. \end{cases} \tag{6.3}$$

Here, ρ is the density and $\mathbf{u}$ is the velocity field. The pressure p and total energy E are related by the ideal gas equation of state:

$$E := \frac{p}{\gamma-1} + \frac{1}{2}\rho|\mathbf{u}|^2, \tag{6.4}$$

with γ being the ratio of specific heats.

The MLMC-FVM algorithm is tested on a problem with a large number of sources of uncertainty. We consider the so-called *cloud-shock interaction problem.* The computational domain is taken to be $\mathbf{D} = [0,1]\times[0,1]$. Let $Y \sim \frac{1}{25} + \mathcal{U}(0,\frac{1}{50})$ and let $Y_1,...,Y_7 \sim \mathcal{U}(0,1)$ denote i.i.d. random variables independent of Y.

The initial data consists of an initial shock with uncertain amplitude and uncertain location given by:

$$\begin{aligned} &\{\rho_0(\mathbf{x},\omega),\ \mathbf{u}_0(\mathbf{x},\omega),\ p_0(\mathbf{x},\omega)\} = \\ &= \begin{cases} \left\{3.86859 + \frac{1}{10}Y_6(\omega),\ (11.2536,\ 0)^\top,\ 167.345 + Y_7(\omega)\right\} & \text{if} \quad \mathbf{x}_1 < Y(\omega), \\ \{1,\ (0,\ 0)^\top,\ 1\} & \text{if} \quad \mathbf{x}_1 > Y(\omega). \end{cases} \end{aligned} \tag{6.5}$$

Furthermore, a high density cloud or bubble with uncertain amplitude and uncertain shape of the form

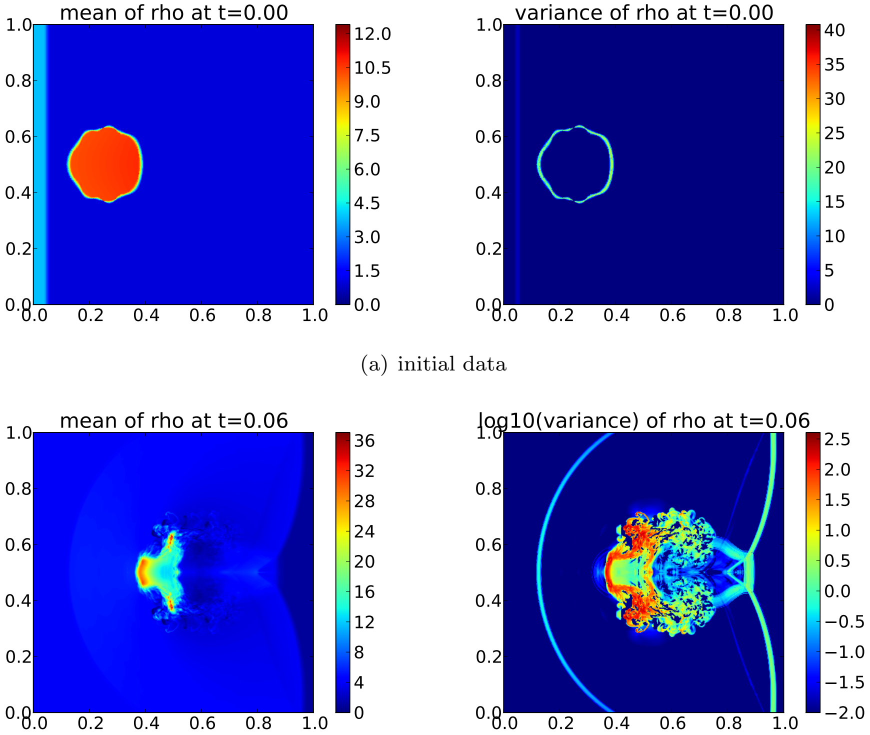
$$\begin{aligned} \rho_{\mathbf{0}}(\mathbf{x},\omega) = 10 + \frac{1}{2}Y_1(\omega) + Y_2(\omega)\sin(4(\mathbf{x}_1 - \frac{1}{4})) + \frac{1}{2}Y_3(\omega)\cos(8(\mathbf{x}_2 - \frac{1}{2})) \\ \text{if} \quad r \le 0.13 + \frac{1}{50}Y_4(\omega)\sin\theta + \frac{1}{100}Y_5(\omega)\sin(10\theta), \end{aligned} \tag{6.6}$$

where

$$r = \sqrt{(\mathbf{x}_1 - 0.25)^2 + (\mathbf{x}_2 - 0.5)^2}, \quad \theta = \frac{\mathbf{x}_1 - 0.25}{r}, \tag{6.7}$$

lies to the right of the shock. The mean and the variance of the initial data are depicted in Figure 1(a). Note that there are 8 sources of uncertainty in the above problem. A parametric representation of the initial data results in a 11 dimensional problem consisting of two space, one time and eight stochastic dimensions. The mean and variance of the solution at time $t = 0.06$ is shown in Figure 1(b). The results are from a MLMC-WENO run with 10 nested levels of resolution ($L = 9$) and the finest resolution is set to 4096×4096 mesh. The number M_L of MC samples at the finest resolution is 8 and number of cores for this run is 1023.

The physics of the flow in this case consists of the supersonic initial shock moving to the right, interacting with the high density bubble and leading to a complex flow pattern that consists of a leading bow shock, trailing tail shocks and a very complex region (near the center) possessing sharp gradients as well as turbulent like smooth features. The mean flow (for the density) consists of the bow shock, tail shocks and a complex region with sharp gradients as well as smooth regions. The variance is concentrated in the smooth region at the center; it is significantly smaller at the tail shocks and almost vanishing at the bow shock. The initial uncertainty in the shape of the bubble seems to lead to a more complex distribution of the variance.



(a) initial data

(b) solution at $t = 0.06$

L	M_L	grid size	CFL	cores	runtime	efficiency
9	8	4096x4096	0.4	1023	5:38:17	96.9%

FIGURE 1. Cloud shock at $t = 0$ and $t = 0.06$ using MLMC-FVM

6.2. **MHD equations of plasma physics.** Next, we consider the MHD equations:

$$\left\{\begin{aligned} \rho_t + \operatorname{div}(\rho\mathbf{u}) &= 0, \\ (\rho\mathbf{u})_t + \operatorname{div}(\rho\mathbf{u}\otimes\mathbf{u} + (p + \frac{1}{2}|\mathbf{B}|^2)I - \mathbf{B}\otimes\mathbf{B}) &= 0, \\ \mathbf{B}_t + \operatorname{div}(\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u}) &= 0, \\ E_t + \operatorname{div}((E + p + \frac{1}{2}|\mathbf{B}|^2)\mathbf{u} - (\mathbf{u}\cdot\mathbf{B})\mathbf{B}) &= 0, \\ \operatorname{div}(\mathbf{B}) &= 0, \end{aligned}\right\} \tag{6.8}$$

Here, $\mathbf{B}$ denotes the magnetic field and the total energy is given by the equation of state (6.4). In this example, the random initial data is a parametric version of the celebrated *Orszag-Tang* vortex which is *randomly* perturbed in two different ways:

(1) **2 sources of uncertainty.** Let $Y_1, Y_2 \sim \mathcal{U}(0,1)$. The phases of the velocities are uncertain and depend on the scaled random variables Y_1, Y_2:

$$\begin{aligned}
&\{\rho_0(\mathbf{x},\omega),\ p_0(\mathbf{x},\omega)\} = \{\gamma^2,\ \gamma\},\\
&\mathbf{u}_0(\mathbf{x},\omega) = \left(-\sin\left(\pi\mathbf{x}_2 + \frac{1}{20}Y_1(\omega)\right),\ \sin\left(\pi\mathbf{x}_1 + \frac{1}{10}Y_2(\omega)\right)\right)^\top,\\
&\mathbf{B}_0(\mathbf{x},\omega) = (-\sin(\pi\mathbf{x}_2),\ \sin(2\pi\mathbf{x}_1))^\top.
\end{aligned} \tag{6.9}$$

(2) **8 sources of uncertainty.** Let $Y_i \sim \mathcal{U}(-1,1),\ i = 1,\dots,8$. The amplitudes of the initial density and pressure are uncertain

$$\begin{aligned}
\rho_0(\mathbf{x},\omega) &= \gamma^2\left(1 + \frac{1}{20}Y_1(\omega)\right),\\
p_0(\mathbf{x},\omega) &= \gamma\left(1 + \frac{1}{20}Y_4(\omega)\right),
\end{aligned} \tag{6.10}$$

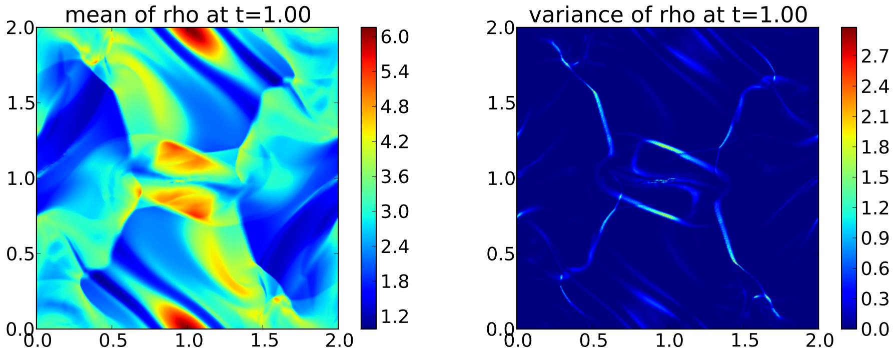
and, additionally, the phases of the initial velocities and the phases with the amplitudes of the initial magnetic fields are also uncertain,

$$\begin{aligned}
\mathbf{u}_0(\mathbf{x},\omega) &= \left(-\sin\left(\pi\mathbf{x}_2 + \frac{1}{20}Y_2(\omega)\right),\ \sin\left(\pi\mathbf{x}_1 + \frac{1}{10}Y_3(\omega)\right)\right)^\top,\\
\mathbf{B}_1(\mathbf{x},\omega) &= -\left(1 + \frac{1}{20}Y_6(\omega)\right)\sin\left(\pi\mathbf{x}_2 + \frac{1}{25}Y_5(\omega)\right),\\
\mathbf{B}_2(\mathbf{x},\omega) &= \left(1 + \frac{1}{20}Y_8(\omega)\right)\sin\left(2\pi\mathbf{x}_1 + \frac{1}{20}Y_7(\omega)\right).
\end{aligned} \tag{6.11}$$

Here, as in the setup for cloud-shock interaction problem in Figure 1(a), a parametric representation of the initial data results in a 11 dimensional problem consisting of two space, one time and eight stochastic dimensions.

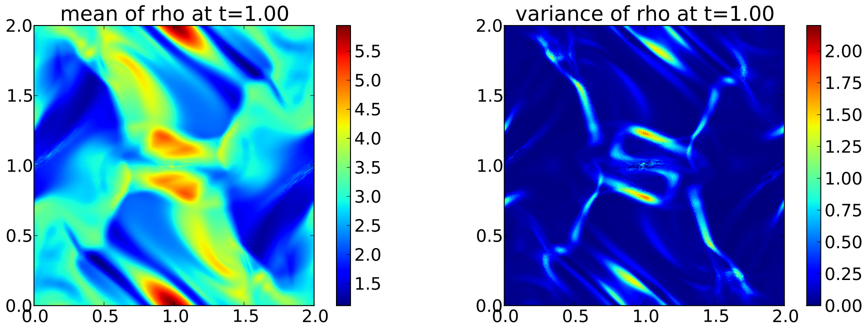
The MLMC-FVM solution is then considered for both versions of the initial data, i.e. with 2 sources (6.9) and with 8 sources (6.10) of uncertainty. The mean field and the variance (for the plasma density) of the solutions are shown in Figures 2 and 3, respectively.

The computation is performed using the MLMC-FVM scheme with second-order WENO reconstruction, and with the HLL three wave solver of [16]. The code uses an upwind discretization of the Godunov-Powell source term. The results shown in these figures are from a computation with 8 levels of refinement ($L = 7$) and the finest mesh resolutions of 2048×2048 mesh cells and 4096×4096 mesh cells for the problem with two sources of uncertainty (6.9) and with eight sources of uncertainty (6.10), respectively. The number of MC samples at the finest resolution for both problems is 4. The problems have more than 10^9 degrees of freedom per time step and the total number of time steps is about 10^4 amounting to an overall computational cost of this simulation between 10^{12} and 10^{13} FLOPS. These numbers show that the simulations are extremely challenging and requires massively parallel architectures. In fact, the problem with 2 sources of uncertainty took about 6 hours (wall-clock) on 128 cores (simulated on ETH's parallel cluster Brutus [49]) and the problem with 8 sources of uncertainty took about 3.5 hours (wall-clock) on 2040 cores (simulated on Palu, CSCS [50]). We also observe that the variance for the problem with eight sources of uncertainty is more diffused than the variance for the problem with two sources of uncertainty.



L	M_L	grid size	CFL	cores	runtime	efficiency
7	4	2048x2048	0.475	128	5:02:14	98.4%

FIGURE 2. Uncertain Orszag-Tang vortex solution at $t = 1.0$ using MLMC-FVM (2 sources of uncertainty). Variance is very large near discontinuities of the path-wise solutions.



L	M_L	grid size	CFL	cores	runtime	efficiency
8	4	4096x4096	0.475	2044	3:17:18	97.0%

FIGURE 3. Uncertain Orszag-Tang vortex solution at $t = 1.0$ using MLMC-FVM (8 sources of uncertainty). Again, the largest variances appear near discontinuities of the path-wise solutions.

It is well known (see [16]) that stable computation of numerical solutions of the Orszag-Tang problem on highly refined meshes (which, by the CFL condition (1.4) entails a correspondingly large number of timesteps) is quite challenging. Since our spatial resolution at mesh level $L = 7$ is very fine, we need an *extremely* robust code like ALSVID for the solve step in MLMC-FVM in order to resolve this problem.

The mean density is quite complicated with shocks along the diagonals of the domain as well a (smooth) current sheet at the center of the domain. The solution consists of discontinuities interspersed within interesting smooth features. Our simulations show that the variance is concentrated at shocks as well as at the current sheets and other interesting smooth regions. From this problem as well as the results of the previous section, we observe that the variance is a very good indicator of where the discontinuities and sharp gradients of the solution are concentrated and would serve as a good *a posteriori* error indicator for adaptive mesh refinement.

6.2.1. *Numerical convergence analysis.* We analyze these particular two dimensional numerical experiments (Orszag-Tang vortex with 2 and 8 sources of uncertainty) in greater detail. Again, we use the high-resolution MLMC-FVM simulations from Figures 2 and 3 as the reference solutions, respectively. We investigate convergence of error vs. work in Figure 4 and Figure 6 for 2 sources of uncertainty and in Figure 5 and Figure 7 for 8 sources of uncertainty. The error in the mean field converges at expected rates. At comparable numerical resolution and accuracy, the MLMC(2) is about two orders of magnitude faster than the MC(2) method for both problems. We observe a slight deterioration in the estimated convergence rates for the variance. This could well be a pre-asymptotic effect. As seen in Figures 6 and 7, the curves are steepening which seems to indicate better rates with further refinement. Again, the MLMC(2) appears considerably faster than the corresponding MC(2) method in delivering variance estimates of comparable numerical accuracy.

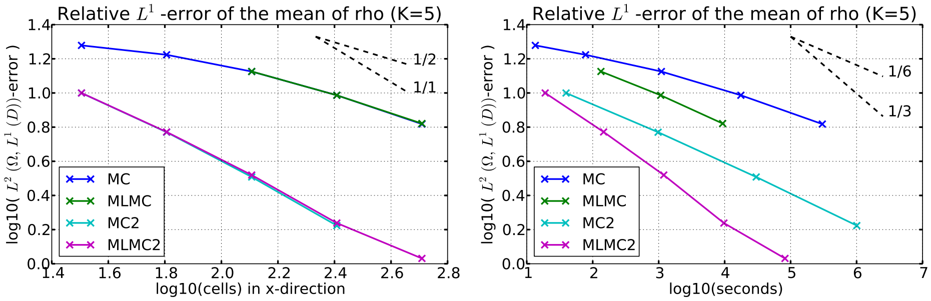


FIGURE 4. Convergence of mean in the uncertain Orszag-Tang vortex simulation (2 sources of uncertainty).

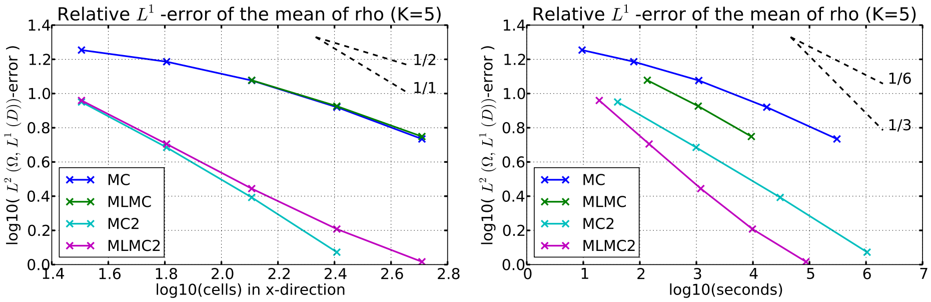


FIGURE 5. Convergence of mean in the uncertain Orszag-Tang vortex simulation (8 sources of uncertainty).

Remark 6.1. Our aim in computing the Orszag-Tang vortex with two and with eight sources of uncertainty in the initial data is to compare the robustness of the MLMC method with respect of an increase in the number of sources of uncertainty.

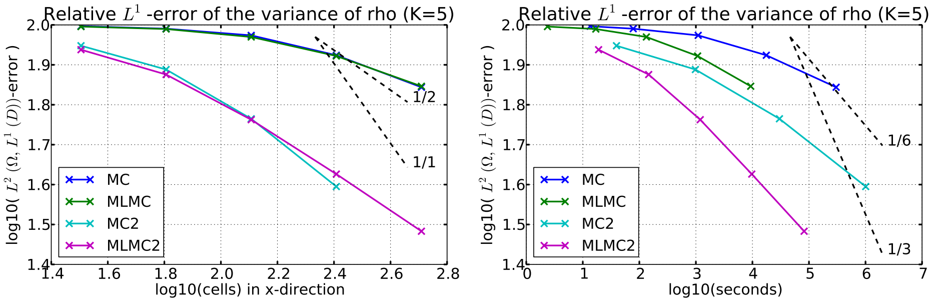


FIGURE 6. Convergence of variance in the uncertain Orszag-Tang vortex simulation (2 sources of uncertainty).

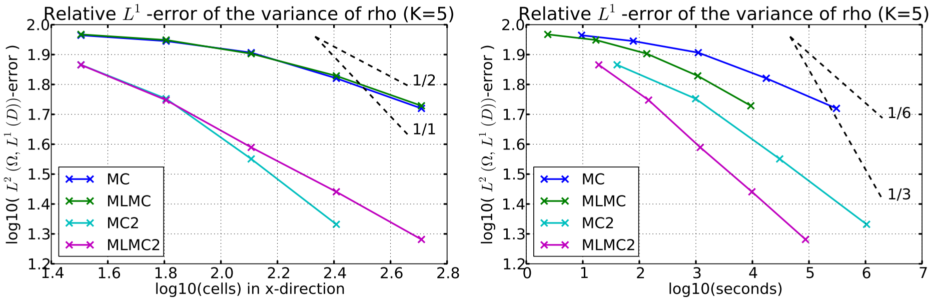


FIGURE 7. Convergence of variance in the uncertain Orszag-Tang vortex simulation (8 sources of uncertainty).

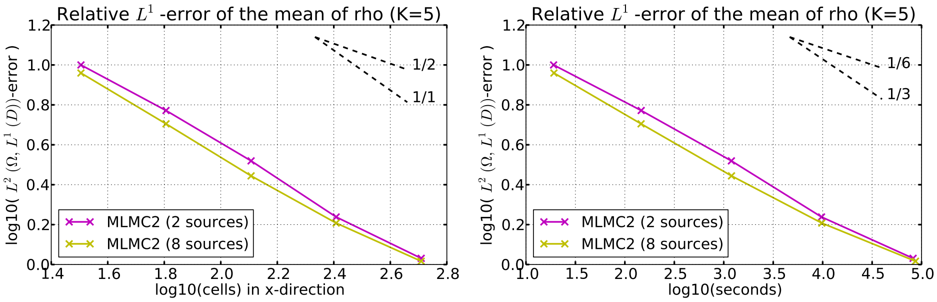


FIGURE 8. Error convergence sensitivity to the number of sources of uncertainty. For both 2 and 8 sources of uncertainty, the convergence of error vs. computational work is almost identical.

To this end, we plot the error vs. resolution and the error vs. runtime for the MLMC(2) FVM with both two and with eight sources of uncertainty in Figure 8. The results in this figure show that the runtime for a fixed level of error is nearly identical whether there are two or eight sources of uncertainty in the initial data. This shows that the MLMC method is quite robust with respect to large number of sources of uncertainty in the data: an increase in the number of sources of uncertainty does not appear to lead to a deterioration in the computational efficiency. Thus, the MLMC method can be used for computing uncertainty in problems with very large number of sources of randomness.

6.2.2. *Efficiency of parallelization.* We test the efficiency of static load balancing for the parallelization procedure described in [42] in this two-dimensional example. Here, the parallelization efficiency is defined as

$$\text{efficiency} := \frac{(\text{cumulative wall-time}) - (\text{cumulative wall-time of MPI calls})}{\text{cumulative wall-time}}. \tag{6.12}$$

It separates the amount of time spent in computing from the amount of time spent in communicating (the latter is indicated with *dashed* lines in runtime plots). In Figure 9, we show the parallelization efficiency of the MLMC-FVM and see that the algorithm is quite efficient and most of the time is spent computing rather than communicating or waiting.

The strong scaling (fixed discretization and sampling parameters while increasing #cores) for this problem is shown in Figure 10. We see that the algorithm scales linearly up to around 4000 cores. Similarly, Figure 11 shows a weak scaling (problem size is equivalent to #cores) up to a similar number of processors. We have not tested the algorithm for a larger number of processors since we are limited by the size of the machine [50], but we expect it to scale upto a much larger number of cores. The results in both one and two space dimensions indicate that our static load balancing algorithm is quite efficient.

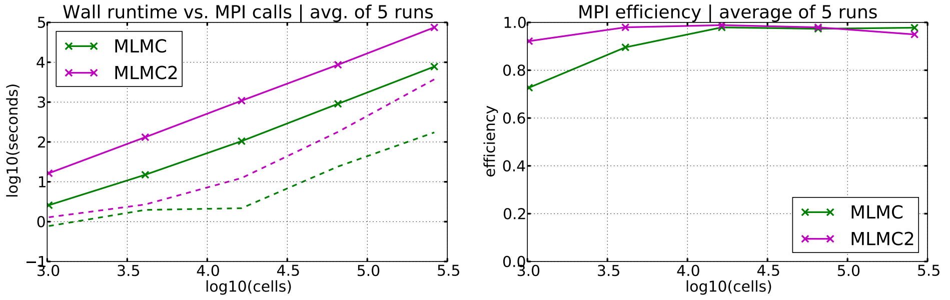


FIGURE 9. MPI overhead.

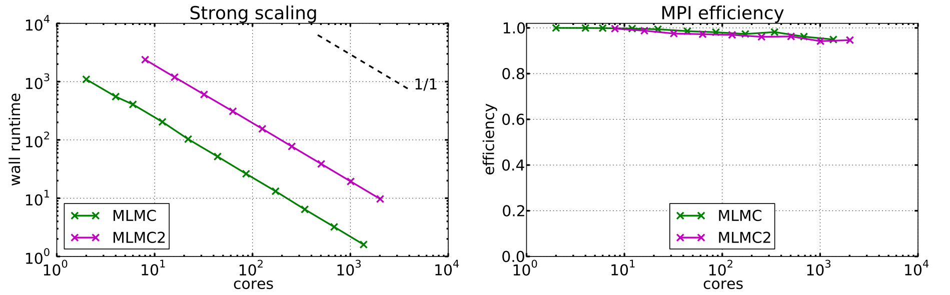


FIGURE 10. Strong scaling. The inferior scalability of DDM has no significant influence on the overall strong scaling for $d = 2$.

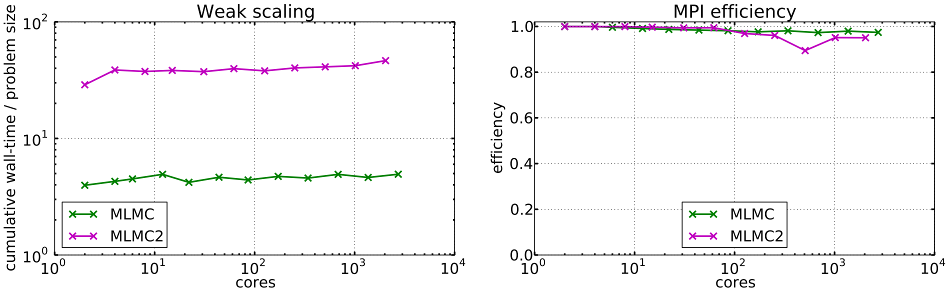


FIGURE 11. Weak scaling. The inferior scalability of DDM has no significant influence on the overall weak scaling for $d = 2$.

6.3. **Shallow-water equations with uncertain bottom topography.** In this section, we consider the shallow-water equations (in two space dimensions):

$$\begin{cases} h_t + (hu)_x + (hv)_y = 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x + (huv)_y = -ghb_x, \\ (hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2}gh^2\right)_y = -ghb_y. \end{cases} \tag{6.13}$$

Here, h is the height of the fluid column above the bottom topography $b = b(x, y)$ over which the fluid flows and (u, v) is the vertically averaged horizontal fluid velocity field. The constant g denotes the acceleration due to gravity.

6.3.1. *Multi-level representation of the bottom topography.* An approximation to the exact bottom topography $b(\mathbf{x}) \in W^{1,\infty}(\mathbf{D})$ is often obtained from the measurements. For instance [7, 13], in the two-dimensional case, nodal measurements $b_{i+\frac{1}{2},j+\frac{1}{2}} := b(\mathbf{x}_{i+\frac{1}{2},j+\frac{1}{2}})$ are obtained at locations $\mathbf{x}_{i+\frac{1}{2},j+\frac{1}{2}} = (x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$, i.e. at vertices of an axiparallel quadrilateral *topography* mesh $\bar{\mathcal{T}}$ (possibly different from FVM mesh $\mathcal{T}$) on the rectangular two-dimensional domain $\mathbf{D}$. Since each measurement $b_{i+\frac{1}{2},j+\frac{1}{2}}$ is prone to uncertainty [13], all measured values are treated as random variables with some prescribed probability distribution; we choose

$$b_{i+\frac{1}{2},j+\frac{1}{2}}(\omega) := b(x_{i+\frac{1}{2},j+\frac{1}{2}}) + Y_{i,j}(\omega), \quad Y_{i,j} \sim \mathcal{U}(-\varepsilon_{i,j}, \varepsilon_{i,j}),\ \varepsilon_{i,j} > 0, \tag{6.14}$$

i.e. $b_{i+\frac{1}{2},j+\frac{1}{2}}(\omega) \in L^2(\Omega, \mathbb{R})$ are random variables (*not* necessarily independent), which deviate from the measurements $b_{i+\frac{1}{2},j+\frac{1}{2}}$ by $\pm\varepsilon_{i+\frac{1}{2},j+\frac{1}{2}}$. Thus, (6.14) provides an approximation to the *uncertain* topography $b(\mathbf{x}, \omega) \in L^2(\Omega, W^{1,\infty}(\mathbf{D}))$.

It is shown in a recent paper [33] that the uncertain bottom topography needs to be represented in a multi-level framework in-order to accelerate the MLMC-FVM algorithm. To introduce the multi-level topography representation, we recall some notation: levels $\ell = 0, \ldots, L$ enumerate nested grids $\mathcal{T}_0, \ldots, \mathcal{T}_L$ that are used in the MLMC-FVM *solver*. Apart from $\mathcal{T}_0, \ldots, \mathcal{T}_L$, we consider an additional hierarchical structure, that will be used in the multi-level representation of the *bottom topography*. More precisely, assume a nested sequence $\{\bar{\mathcal{T}}_{\bar{\ell}} = \bar{\mathcal{T}}^1_{\bar{\ell}} \times \cdots \times \bar{\mathcal{T}}^d_{\bar{\ell}},\ \bar{\ell} = 0, \ldots, \bar{L}\}$ of isotropic regular d-dimensional axiparallel quadrilateral meshes for the physical *bounded* domain $\mathbf{D} = I_1 \times \cdots \times I_d \subset \mathbb{R}^d,\ I_r \subset \mathbb{R},\ d = 1, 2$, each of them obtained by $\bar{\ell}$ uniform refinements of some initial, regular mesh $\bar{\mathcal{T}}_0$ (of domain $\mathbf{D}$) consisting of the cells C_k^0, $k = 1, \ldots, \#\bar{\mathcal{T}}_0$. Note, that a-priori we do *not* assume any relation between $\bar{L}$ and L. However, for the sake of consistency, we assume

$$\bar{\mathcal{T}}_{\bar{\ell}} = \mathcal{T}_{\ell}, \qquad \text{provided } \bar{\ell} = \ell.$$

For $p \in \mathbb{N}_0$, define $\mathcal{Q}^p(\mathbf{D}, \bar{\mathcal{T}})$ to be the space of piece-wise multivariate tensor product polynomials of degree p on a mesh $\bar{\mathcal{T}}$ of a bounded domain $\mathbf{D}$ having essentially bounded weak derivatives up to order p, i.e.

$$\mathcal{Q}^p(\mathbf{D}, \bar{\mathcal{T}}) := \{f \in W^{p,\infty}(\mathbf{D}) : f|_C \in \mathcal{Q}_p(C),\ \forall C = C_1 \times \cdots \times C_d \in \bar{\mathcal{T}}\}, \tag{6.15}$$

where $\mathcal{Q}_p(C)$ is the space of multivariate tensor product polynomials on cell C,

$$\mathcal{Q}_p(C) := \{\mathbf{x} \mapsto p_1(\mathbf{x}_1) \cdot \cdots \cdot p_d(\mathbf{x}_d) : p_r \in \mathcal{P}_d(C_r),\ \forall r = 1, \ldots, d\}.$$

We assume that *uncertain* measurements $b_{i+\frac{1}{2},j+\frac{1}{2}}(\omega) := b(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, \omega)$ of the exact bottom topography $b(x,y)$ are available, as in (6.14). Then $b_{i+\frac{1}{2},j+\frac{1}{2}}(\omega)$ are treated as nodal values and are linearly interpolated in each dimension using the bilinear *hierarchical* interpolation operator,

$$\begin{aligned}\mathcal{I}^{\bar{L}} b(x,y,\omega) &= \sum_{\bar{\ell}=0}^{\bar{L}} \sum_{\bar{\ell}'=0}^{\bar{L}} b^{\bar{\ell},\bar{\ell}'}(x,y,\omega),\\ b^{\bar{\ell},\bar{\ell}'} &:= \mathcal{I}_{\bar{\ell},\bar{\ell}'} b - \mathcal{I}_{\bar{\ell}-1,\bar{\ell}'} b - \mathcal{I}_{\bar{\ell},\bar{\ell}'-1} b, \quad \mathcal{I}_{-1,\cdot} \equiv \mathcal{I}_{\cdot,-1} \equiv 0,\end{aligned} \tag{6.16}$$

where $\mathcal{I}_{\bar{\ell},\bar{\ell}'}$ denotes bilinear *nodal* interpolation operator on the quadrilateral mesh $\bar{\mathcal{T}}^1_{\bar{\ell}} \times \bar{\mathcal{T}}^2_{\bar{\ell}'}$.

Each $b^{\bar{\ell},\bar{\ell}'}(x,y,\omega) \in L^2(\Omega, \mathcal{Q}^1(I_1 \times I_2, \bar{\mathcal{T}}^1_{\bar{\ell}} \times \bar{\mathcal{T}}^2_{\bar{\ell}'}))$ is a linear combination of the multivariate tensor products of two *hierarchical* "hat" ("Schauder") basis functions,

$$b^{\bar{\ell},\bar{\ell}'}(x,y,\omega) = \sum_{k=1}^{\hat{N}_{\bar{\ell}}} \sum_{k'=1}^{\hat{N}_{\bar{\ell}'}} b^{\bar{\ell},\bar{\ell}'}_{k,k'}(\omega) \varphi^{\bar{\ell}}_k(x) \varphi^{\bar{\ell}'}_{k'}(y), \quad b^{\bar{\ell},\bar{\ell}'}_{k,k'} \in L^2(\Omega, \mathbb{R}). \tag{6.17}$$

The interpolated bottom topography belongs to the space

$$\mathcal{I}_{\bar{L}} b(\mathbf{x}, \omega) \in L^2(\Omega, \mathcal{Q}^1(I_1 \times I_2, \bar{\mathcal{T}}_{\bar{L}})).$$

6.3.2. *2-D numerical experiments: Random perturbation of lake at rest.* We consider the shallow-water equations in the computational domain $\mathbf{D} = [0,2] \times [0,2]$, and investigate the evolution of a random perturbation of the lake at rest coupled with outflow boundary conditions.

The uncertain bottom topography $b(x,\omega)$ is represented in terms of the nodal, bivariate hierarchical basis (6.16) - (6.17) with random amplitudes. Notice that, formally, this bilinear basis can be obtained by tensorizing the univariate Schauder basis of $C^0([0,2])$. Notice also that we used in the present study only *isotropically supported* product functions. The bottom topography was resolved to 6 levels (i.e. $\bar{L} = 5$, $\bar{\ell}, \bar{\ell}' = 0, \ldots, 5$) where coefficients $b^{\bar{\ell},\bar{\ell}'}_{k,k'}(\omega)$ are given by mean values $\bar{b}^{\bar{\ell},\bar{\ell}'}_{k,k'}$ that are perturbed by independent uniformly distributed centered random variables with decaying variances,

$$b^{\bar{\ell},\bar{\ell}'}_{k,k'}(\omega) = \bar{b}^{\bar{\ell},\bar{\ell}'}_{k,k'} + Y^{\bar{\ell},\bar{\ell}'}_{k,k'}(\omega) \sim \frac{2}{5}\mathcal{U}(-\varepsilon_{\bar{\ell},\bar{\ell}'}, \varepsilon_{\bar{\ell},\bar{\ell}'}), \tag{6.18}$$

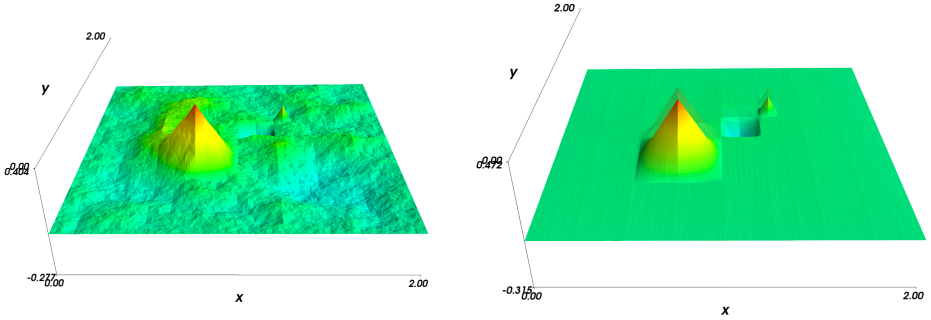
where all coefficients $\bar{b}^{\bar{\ell},\bar{\ell}'}_{k,k'}$ are zero except

$$\bar{b}^{3,3}_{2,2} = 0.4, \quad \bar{b}^{4,4}_{6,6} = -0.32, \quad \bar{b}^{5,5}_{11,11} = 0.12, \tag{6.19}$$

and

$$\varepsilon_{0,\cdot} = \varepsilon_{\cdot,0} = 0, \quad \varepsilon_{\bar{\ell},\bar{\ell}'} = 2^{-\max\{\bar{\ell},\bar{\ell}'\}}, \ \forall \bar{\ell} \geq 1. \tag{6.20}$$

A realization of the uncertain bottom topography and the corresponding mean and variance are shown in Figure 12.



(a) one realization for some **fixed** $\omega \in \Omega$

(b) mean and variance of $b(\mathbf{x},\omega)$

FIGURE 12. Uncertain bottom topography (6.18) with 9 hierarchical levels ($\bar{L} = 8$).

Next, we consider the initial data $\mathbf{U}_0$ to be a *random* perturbation of a lake-at-rest. Let $Y \sim \frac{1}{50} + \frac{1}{100}\mathcal{U}(-1,1)$ be a random variable independent of $\{Y^{\bar{\ell},\bar{\ell}'}_{k,k'}\}$. An initial perturbation around $\mathbf{x}_0 = (x_0, y_0) = (1.0, 0.7)$ with a radius $r = \frac{1}{10}$ reads

$$h_0(x,y,\omega) = \begin{cases} 1.0 + Y(\omega) - b(x,y,\omega) & \text{if} \quad |\mathbf{x} - \mathbf{x}_0| < r, \\ 1.0 - b(x,y,\omega) & \text{if} \quad |\mathbf{x} - \mathbf{x}_0| > r, \end{cases} \tag{6.21}$$

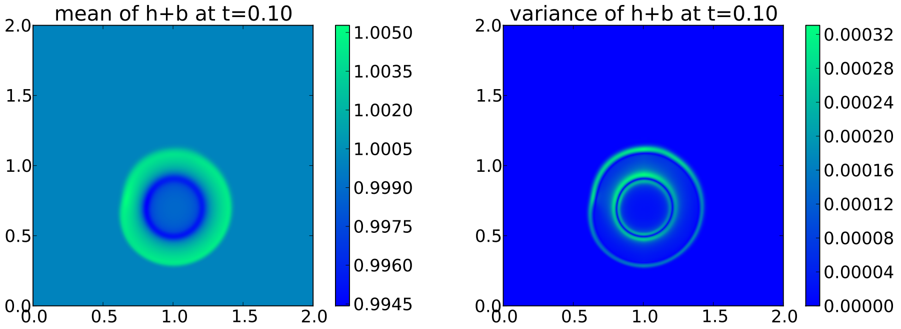
with $b(x,\omega)$ as defined in (6.18) and the initial layer velocities set to zero, i.e.

$$\{u_0(x,y,\omega),\ v_0(x,y,\omega)\} = \{0.0,\ 0.0\}. \tag{6.22}$$

Note, that here we have a very large number of sources of uncertainty ($(2^5-1)^2-1 = 962$).

Reference solutions, computed with the second-order entropy stable scheme [15] at time $T = 0.1$ is depicted in Figure 13. The results are computed on 9 nested levels of resolution ($L = 8$) with the finest resolution being on a 4096×4096 mesh and with time steps reduced accordingly in order to maintain the same CFL constant over all discretization levels. The simulation is run on 2044 cores and 16 samples are taken for the finest mesh resolution.

The above problem is quite involved due to large number of sources of uncertainty as well as the underlying difficulty of simulating small perturbations of steady states. The reference solution show that the wave (in mean) spreads out of the initial source. The variance is distributed in a non-linear and complicated manner with large amount of variance corresponding to the uncertainties in the bottom topography.



L	M_L	grid size	CFL	cores	runtime	efficiency
8	16	4096x4096	0.45	2044	2:13:51	97.5%

FIGURE 13. Reference solution for perturbed steadystate (6.21) using MLMC-FVM . Initial perturbation evolves into asymmetric ribbon wave with uncertain amplitude.

6.3.3. *Numerical convergence analysis.* We investigate convergence of error vs. work in Figure 14 and Figure 15. Here we use the MLMC-FVM simulation from Figure 13 with 9 levels of resolution with the finest resolution being on a 2048 × 2048 mesh as the reference solution $\mathbf{U}_{\text{ref}}$. The error in the mean field converges at expected rates. At comparable numerical resolution and accuracy, the MLMC2 is about two orders of magnitude faster than the MC2 method for this problem. We observe a slight deterioration in the estimated convergence rates for the variance. This could well be a pre-asymptotic effect. Again, the MLMC2 appears to be slightly faster than the corresponding MC2 method in delivering variance estimates of comparable numerical accuracy.

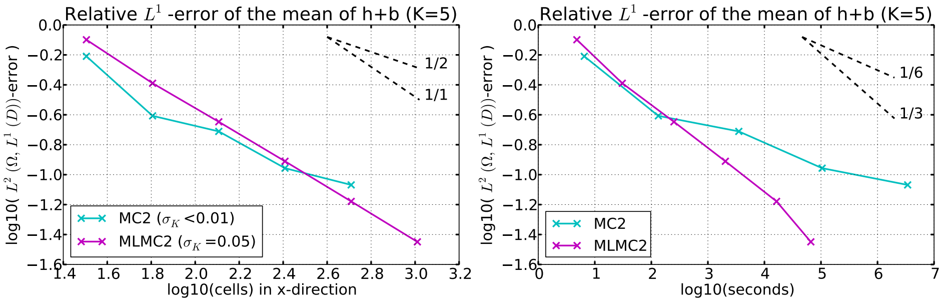


FIGURE 14. Convergence of estimated mean in the 2-D simulation (6.21). MLMC methods are 3 orders of magnitude faster than MC.

6.3.4. *Speed up due to hierarchical topology representation.* We test the gain in efficiency due to the multi-level hierarchical representation of the uncertain bottom topography (6.16) by comparing with a simulation that uses the standard MLMC algorithm. In other words, the MLMC2 (full) simulation uses the underlying bottom topography (at the resolution of the underlying topography mesh) for all shallow water samples. In particular, simulations at the coarsest level of the FVM mesh use

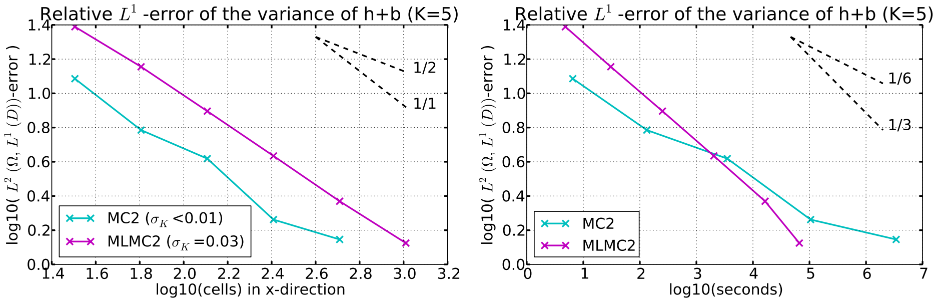


FIGURE 15. Convergence of estimated variance in the 2-D simulation (6.21). MLMC methods are asymptotically faster than MC.

the topography at the finest level of the underlying topography mesh. We compare MLMC2 (full) with MLMC2 (truncated) which uses the representation (6.16) on the perturbations of lake at rest steady state problem in Figure 16. As suggested by the theory of [33], the two methods should lead to an identical order of the error for a given space-time resolution. We verify this in Figure 16. On the other hand, the MLMC2 (truncated) is at least an order of magnitude faster than the MLMC2 (full) showing that the multi-level representation of the uncertain bottom topography really provides a significant gain in efficiency.

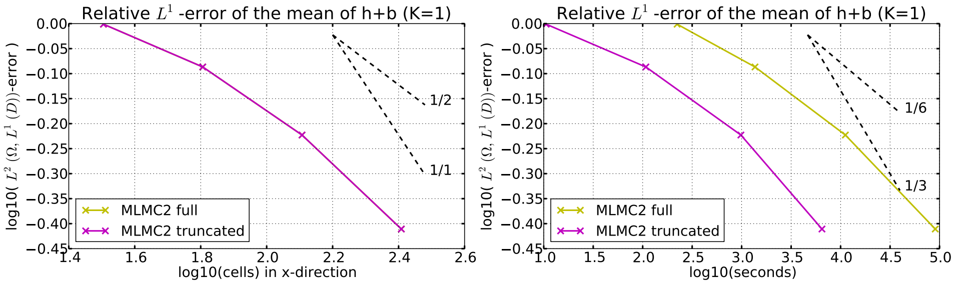


FIGURE 16. Convergence of estimated mean for 2-D steadystate (6.21) with full ($\bar{L} = 8$) and truncated ($\ell + 1$) number of levels in the hierarchical representation (6.18) of bottom topography. For a given mesh resolution, both estimators coincide. The implementation with the truncated number of levels is more than 10 times faster on a mesh of 256 × 256 cells.

6.4. **Burgers' equation with random flux.** The deterministic Burgers' equation is the simplest example of the *non-linear* scalar conservation law. It is given by

$$u_t + f(u)_x = 0, \qquad f(u) = \frac{u^2}{2}. \tag{6.23}$$

The solutions to (6.23) are well posed provided initial data $u_0(\cdot) = u(\cdot, 0)$ is given. We consider deterministic initial data of the form

$$u_0(x) = \sin(\pi x). \tag{6.24}$$

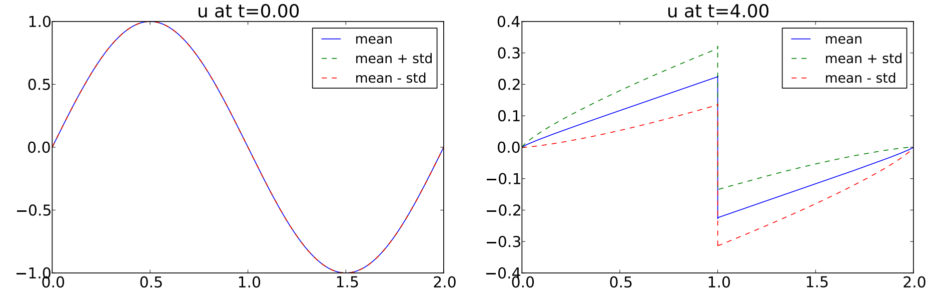
Notice that $\|u_0\|_{L^\infty(\mathbb{R})} = 1$, hence one can choose $\bar{R} = 1$ in (2.28) of Theorem 2.4.

6.4.1. *Uniformly perturbed flux.* We consider a *random* version of Burgers' equation (6.23) with the *random* flux

$$f(u,\omega) = \frac{u^{p(\omega)}}{p(\omega)}, \qquad p \sim \mathcal{U}(1.5, 2.5). \tag{6.25}$$

It is straightforward to verify that the random flux f defined above is a *bounded random flux* with each realization $f(\omega) \in \mathrm{Lip}(-R, R)$ with $R = \bar{R}$.

The initial data (6.24) and the reference solution (obtaind by MLMC-FVM) at time $t = 4$ are depicted in Figure 17. There are 13 levels ($L = 12$) of FVM mesh resolution with the finest resolution (at the finest level $\ell = L$) being 32768 cells. The Rusanov numerical flux with a second order accurate WENO reconstruction was used. At every point $x \in [0, 2]$ the solid line represents the mean and the dashed lines represent the mean ± standard deviation of the (random) solution. For each *sample* (realization) of the *random* flux (6.25), the smooth initial data evolves into (as expected) discontinuity in the physical space and a shock forms at $\mathbf{x}_1 = 1.0$. Given the fact that the flux function is random, the variance is high over the entire physical domain and is not just concentrated at the discontinuity.



L	M_L	grid size	CFL	cores	runtime	efficiency
12	16	32768	0.475	104	6:31:48	99.3%

FIGURE 17. MLMC-FVM solution of Burgers' equation with uniformly perturbed flux (6.25).

Next, we use this high-resolution MLMC-FVM simulation from Figure 17 as the reference solution. We investigate the convergence of error vs. work in Figure 18. The error in the mean field converges at expected rates; furthermore, the MLMC2 method is almost two orders of magnitude faster than the MC2 method (for the same numerical accuracy).

6.5. **Two phase flows in a porous medium with uncertain permeabilities.** Many interesting phenomena (such as water flooding) in an oil and gas reservoir can be modeled by using two phase flows in a porous medium. For simplicity, we consider the flow of two phases (oil and water) in a one dimensional reservoir [4]. The model reduces to a one dimensional scalar conservation law (with the Buckley-Leverett flux):

$$\begin{aligned} S_t + f(S)_x &= 0, \\ f(S) &= \frac{qK\lambda_o(S)}{\lambda_w(S) + \lambda_o(S)}. \end{aligned} \tag{6.26}$$

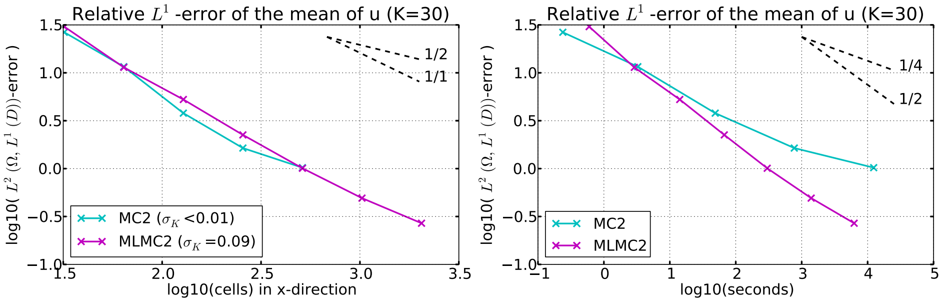


FIGURE 18. Convergence of mean for MLMC-FVM solution of Burgers' equation with uniformly perturbed flux (6.25).

Here, the variable S represents the saturation of oil, q is total flow rate, K the rock permeability and $\lambda_w, \lambda_o : [0,1] \mapsto \mathbb{R}$ are the relative permeabilities of the water and oil phases, respectively. In practice, the rock permeability needs to be measured and is prone to uncertainty. Similarly, the relative permeabilities are measured in laboratory experiments and are characterized by uncertainty. In this example, we focus on the case of uncertain relative permeabilities, which are frequently taken to be of the form

$$\lambda^o = S^2, \qquad \lambda^w(S) = (1-S)^2.$$

We add *random perturbations* to λ^o and λ^w, i.e.

$$\begin{aligned} \lambda^o(S) &= S^2 + \varepsilon_o Y_o(\omega) S^2(1-S), \\ \lambda^w(S) &= (1-S)^2 + \varepsilon_w Y_w(\omega)(1-S)^2 S, \end{aligned} \tag{6.27}$$

with

$$\varepsilon_o = 0.3, \qquad \varepsilon_w = 0.2, \qquad Y_o, Y_w \sim \mathcal{U}[-1,1].$$

Uncertain relative permeabilities defined in (6.27) are depicted in Figure 19.

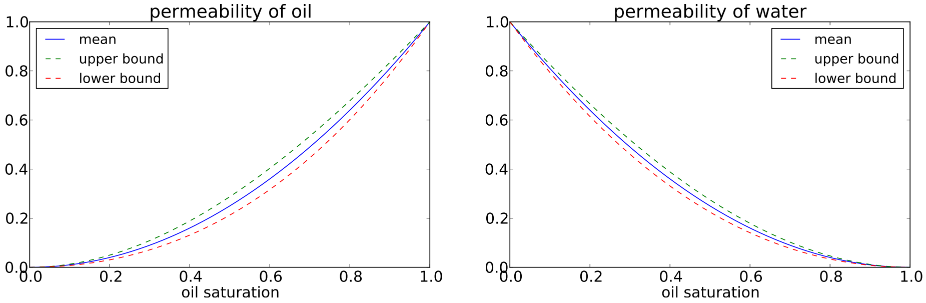


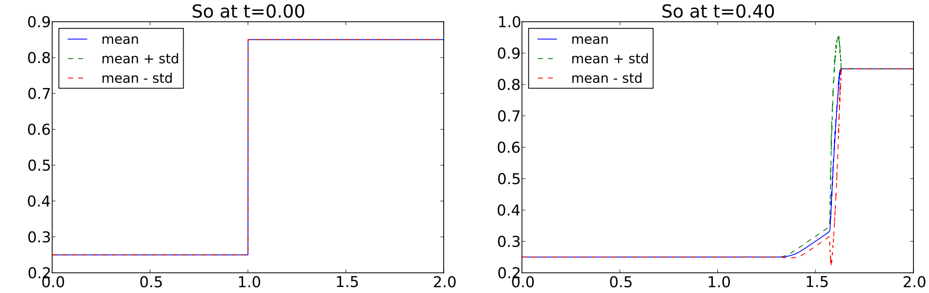
FIGURE 19. The mean and the upper/lower bounds for the *uncertain* relative oil and water permeabilities as defined in (6.27).

6.5.1. *1-D numerical experiment: deterministic initial shock.* We set $K = 1.0$ and $q = 1.0$. The initial data is given by a *deterministic* shock,

$$S^o = \begin{cases} 0.25 & \text{if} \quad \mathbf{x}_1 < 1.0, \\ 0.85 & \text{if} \quad \mathbf{x}_1 > 1.0. \end{cases} \tag{6.28}$$

Notice that $\|u_0\|_{L^\infty(\mathbb{R})} = 0.85$, hence one can choose $\bar{R} = 0.85$ in (2.28) of Theorem 2.4. Furthermore, the random flux defined in (6.26) with random permeabilities (6.27) is a *bounded random flux* with each realization $f(\omega) \in \mathrm{Lip}(-R, R)$ with $R = \bar{R}$.

The initial data (6.28) and the reference solution at time $t = 0.4$ are depicted in Figure 20. There are 13 levels ($L = 12$) of FVM mesh resolution with the finest resolution (at the finest level $\ell = L$) being 32768 cells. At every point $x \in [0, 2]$, the solid line represents the mean and the dashed lines represent the mean ± standard deviation of the (random) solution. For each *sample* of the *random* permeabilities λ^o, λ^w, the initial shock splits into a compound shock, consisting of right going rarefaction that is immediately followed by a right moving shock wave. Notice the improvement of the regularity in the stochastic solution: deterministic pathwise solutions for each sample are discontinuous due to formation of the shock; nevertheless, the mean of the solution appears to be continuous. Furthermore, the uncertainty seems to be concentrated on the compound shock in this case.



L	M_L	grid size	CFL	cores	runtime	efficiency
12	16	32768	0.475	104	1:12:27	99.0%

FIGURE 20. MLMC-FVM solution of the Buckley-Leverett equation (6.26) with uncertain permeabilities (6.27) and deterministic initial shock (6.28).

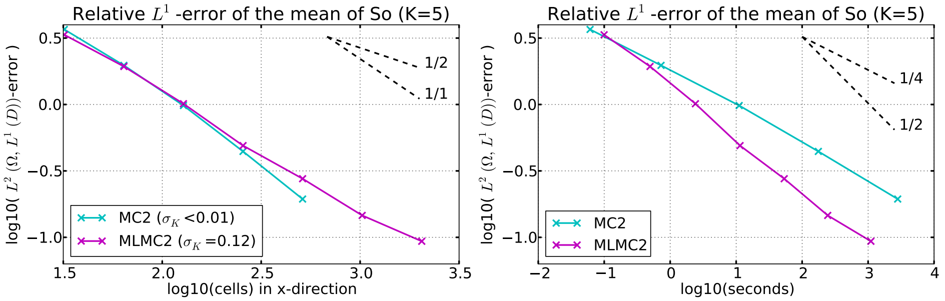


FIGURE 21. Convergence of mean in the Buckley-Leverett equation (6.26) with uncertain permeabilities (6.27).

Next, we use the high-resolution MLMC-FVM simulation from Figure 20 as the reference solution. We investigate convergence of error vs. work in Figure 21 and

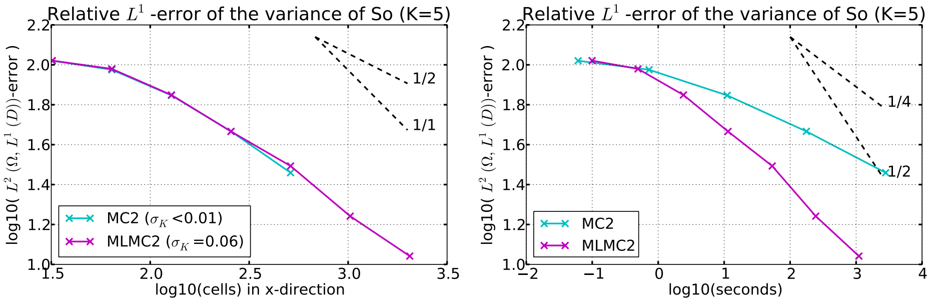


FIGURE 22. Convergence of variance in the Buckley-Leverett equation (6.26) with uncertain permeabilities (6.27).

Figure 22. The error in the mean field converges at expected rates. If we compare the MC2 method (for the same numerical accuracy) with the MLMC2, the latter is approximately an order of magnitude faster for the approximation of the mean field and approximately two orders of magnitude faster for the approximation of variance.

6.5.2. *Exposure of the non-monotonicity of the MLMC-FVM estimator.* In this section we provide examples of MLMC-FVM estimated mean and variance fields with much smaller discretization parameters (number of mesh cells and mesh levels) in order to expose the non-monotonicity of the MLMC-FVM estimator mentioned in Remark 4.3. In particular, we consider the same problem as in subsubsection 6.5.1, only with substantially smaller space and time step size.

Two cases are considered: mesh size of 128 with 5 levels and mesh size 1024 with 7 levels. The results are presented in Figures 23 - 24, respectively.

For the first case, where the resolution is *coarse*, small overshoots in the approximated mean and a significant negative overshoot to the left of the peak in the approximated variance appear. These phenomena are, indeed, discretization artifacts which are due to the MLMC-FVM estimation. For a detailed explanation, we refer to Remark 7.1.

For the second case, where the resolution is *fine*, the approximations of the mean and the variance have almost converged, i.e. the numerical overshoots have substantially smaller amplitude as compared to results presented in Figure 23. In fact, these overshoots are not significant at all and are expected to eventually vanish due to the convergence of the MLMC-FVM.

6.6. **Euler equations with uncertain equation of state.** Next we consider a *random* version of the Euler equations (6.3) from subsection 6.1 with the *random* constant of specific heats in (6.4)

$$\gamma = \gamma(\omega), \qquad \gamma \sim \mathcal{U}(5/3 - \varepsilon, 5/3 + \varepsilon), \quad \varepsilon = 0.1. \tag{6.29}$$

6.6.1. *Uniformly perturbed specific heats in 1d: Sod shock tube.* We consider one dimensional version of the Euler equations (6.3) with (6.29) in the domain $\mathbf{D} = [0, 2]$. The initial data consists of the shock at $x = 1$:

$$U_0(x, \omega) = \{\rho_0(x, \omega), u_0(x, \omega), p_0(x, \omega)\} = \begin{cases} \{3.0, 0.0, 3.0\} & \text{if} \quad x < 1, \\ \{1.0, 0.0, 1.0\} & \text{if} \quad x > 1. \end{cases} \tag{6.30}$$

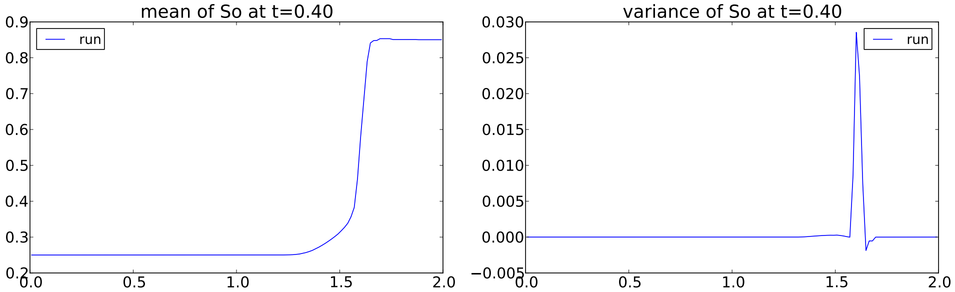


FIGURE 23. MLMC-FVM solution as in Figure 20, with only 128 cells and 5 levels ($L = 4$). Notice the small overshoot in the approximated mean and a significant negative overshoot to the left of the peak in the approximated variance. Such overshoots are explained by the non-monotonicity of the MLMC-FVM estimator.

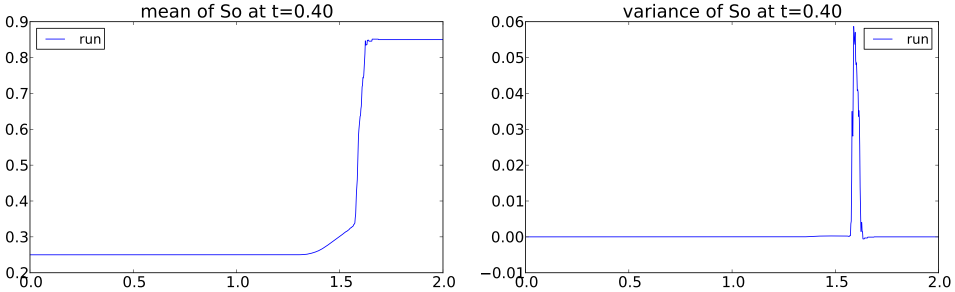
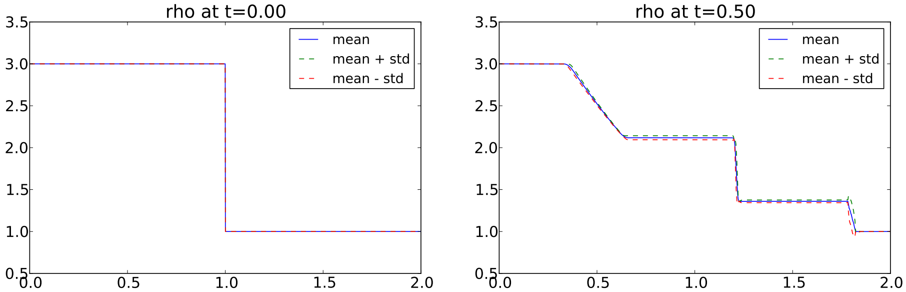


FIGURE 24. MLMC-FVM solution as in Figure 20, with only 1024 cells and 7 levels ($L = 6$). The approximations of the mean and the variance have almost converged, i.e. the overshoots are *much* smaller compared to results presented in Figure 23. In fact, these overshoots are not significant at all and eventually vanish due to the convergence of the MLMC-FVM method.

The initial data (6.30) and the reference solution at time $t = 0.5$ are depicted in Figure 25. There are 9 levels ($L = 8$) of FVM mesh resolution with the finest resolution (at the finest level $\ell = L$) being 2048 cells. Rusanov flux with second order accurate positivity preserving WENO reconstruction was used. At every point $x \in [0, 2]$ the solid line represents the mean and the dashed lines represent the mean ± standard deviation of the (random) solution. For each *sample* the *random* constant γ of specific heats, the initial shock splits into three waves: a left going rarefaction wave, a right going contact discontinuity and a right going shock wave. The variances are relatively small (compared to the mean) with a concentration of the variance near the shock.

6.6.2. *Uniformly perturbed specific heats in 2d: cloud-shock.* The MLMC-FVM algorithm is tested on a problem that is analogous to the one presented in subsection 6.1. The computational domain is again taken to be $\mathbf{D} = [0, 1] \times [0, 1]$. Compared to subsection 6.1, we assume a *random* constant of specific heats (6.29)



L	M_L	grid size	CFL	cores	runtime	efficiency
8	16	2048	0.475	36	0:00:39	99.1%

FIGURE 25. MLMC-FVM solution of the Sod shock tube (6.30) with *random* specific heats constant (6.29).

and we consider the *deterministic* initial data for cloud shock problem:

$$\begin{aligned}&\{\rho_0(\mathbf{x},\omega),\ \mathbf{u}_0(\mathbf{x},\omega),\ p_0(\mathbf{x},\omega)\} = \\ &= \begin{cases}\left\{3.86859,\ (11.2536,\ 0)^\top,\ 167.345\right\} & \text{if} \quad \mathbf{x}_1 < 0.05,\\ \{1,\ (0,\ 0)^\top,\ 1\} & \text{if} \quad \mathbf{x}_1 > 0.05,\end{cases}\end{aligned} \tag{6.31}$$

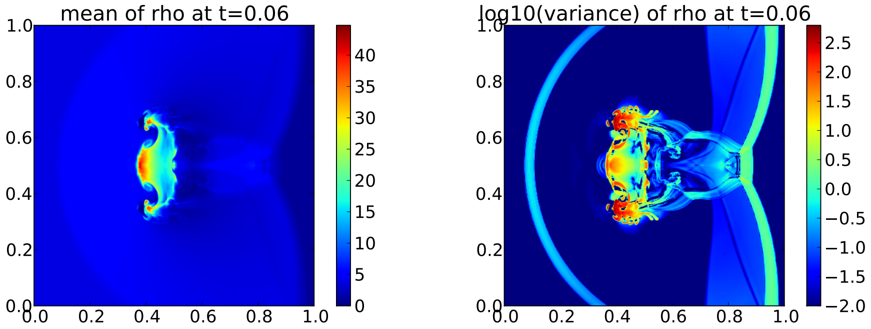
with a high density cloud (or bubble) lying to the right of the shock

$$\rho_0(\mathbf{x},\omega) = 10, \quad \text{if} \quad \sqrt{(\mathbf{x}_1 - 0.25)^2 + (\mathbf{x}_2 - 0.5)^2} \leq 0.15. \tag{6.32}$$

The mean and variance of the solution at time $t = 0.06$ obtained by the numerical simulation using MLMC-FVM are given in Figure 26. The results are from a MLMC-WENO run with 9 nested levels of resolution ($L = 8$) and the finest resolution is set to 2048×2048 mesh. Rusanov flux with second order accurate positivity preserving WENO reconstruction was used. The number M_L of MC samples at the finest resolution is 8 and number of cores for this run is 128. Although the flow and uncertainty appear to be similar to the one discussed in subsection 6.1, there are important differences. In particular, the variance near the bow and tail shocks appears to be spread over a larger region compared to the case of random initial data. Furthermore, the smooth regions after the bow shock have a very different distribution of uncertainty in this case.

6.7. **Verification of the derived constants in the asymptotic error bounds.** In this section, we would like to verify the results derived in subsection 4.1, i.e. that there is only a *small* (of order $\sqrt{2M_L}$) difference in constants of the asymptotic error convergence bounds (3.6) and (4.10). To this end, consider deterministic and stochastic one-dimensional *Sod shock tube* [32] problems modeled by Euler equations of gas dynamics (6.3). Four different configurations are considered:

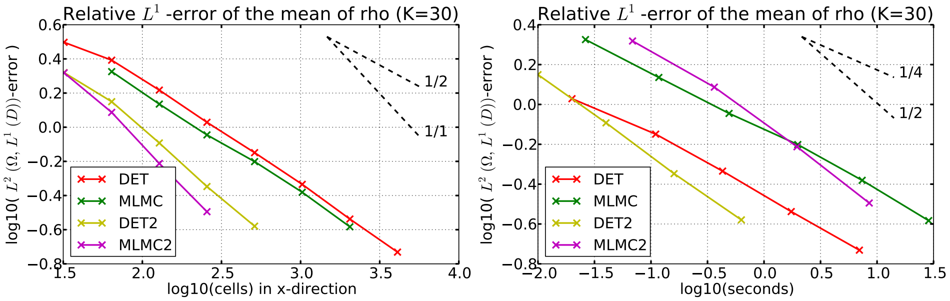
	Initial data	Solver	Reconstruction	CFL
DET	deterministic	HLL 3-wave	-	0.9
MLMC	stochastic	MLMC + 3-wave HLL	-	0.9
DET2	deterministic	HLLC 3-wave	WENO	0.475
MLMC2	stochastic	MLMC + HLL 3-wave	WENO	0.475



L	M_L	grid size	CFL	cores	runtime	efficiency
8	8	2048x2048	0.475	128	6:06:48	86.3%

FIGURE 26. MLMC-FVM solution of the cloud-shock (6.31) - (6.32) with *random* specific heats constant (6.29).

The convergence results, i.e. error vs. number of cells and error vs. computational runtime are depicted in Figure 27. In the table below the figure, the approximated constants C_{FVM} and C_{MLMC} from asymptotic error bounds (3.6) and (4.10) are provided. The values of the constants were obtained by the least squares fitting of the linear polynomial to the obtained logarithmic error convergence data. The measured differences in the constants are *small* (approximately 1.4 and 1.75 times) and hence support the derived estimate (4.13) presented in subsection 4.1.



	C_{FVM}	C_{MLMC}	Quotient
DET, MLMC	0.63	0.88	1.40
DET2, MLMC2	0.52	0.91	1.75

FIGURE 27. Error convergence for deterministic and stochastic *Sod shock tube* problem. For some *fixed* computational time, errors of the *stochastic* (MLMC(2)) runs are only approximately 1.4−1.75 times larger than errors of the *deterministic* (DET(2)) runs.

6.8. **Sparse Tensor MLMC estimation of two-point correlations.** So far, we presented numerical experiments which addressed the MLMC-FVM estimation of mean fields and of variances of random entropy solutions as functions of space and time. In Section 4.2, we also addressed theoretically the efficient computation of two- and of k-point correlation functions. Theorem 4.4 stated that in parallel to any MLMC-FVM simulation, MLMC estimates of two- and k-point correlation functions in the entire domain can be obtained by sparse tensorization of Finite Volume solutions in complexity which equals, up to logarithmic terms, that of the MLMC-FVM estimate of the mean field. The computational realization of the sparse tensor product projects is based on a *multilevel splitting* of the space of piecewise constant functions (and, therefore, of a multilevel decomposition of the Finite Volume solutions) on the mesh hierarchy $\{\mathcal{T}_\ell\}_{\ell=0}^{\infty}$.

6.8.1. *Multiresolution Basis.* For the efficient numerical realization of the sparse projectors $\widehat{P}_L^{(k)}$ defined in (4.26), a *Multiresolution* basis of the spaces S_L containing the FV approximations of the (pathwise) entropy solutions must be chosen. We assume here, for simplicity of exposition, that we are given a FV solver which is based on *cell averages*, so that the spaces S_ℓ are the space of simple (or step) functions on the triangulation $\mathcal{T}_\ell$.

In the following numerical experiments in one space dimension, which are based on the cartesian grid FV solver ALSVID (see [1], [2]), this will always be the following:

Definition 6.2. *The hierarchical basis*

$$\hat{\psi}_k^l(x) = \begin{cases} 1, & x \in \left[2k \cdot 2^{-l}, (2k+1)2^{-l}\right) \\ -1, & x \in \left[(2k+1)2^{-l}, (2k+2)2^{-l}\right) \\ 0, & \textit{otherwise} \end{cases} \tag{6.33}$$

is called Haar-wavelet basis*.*

We remark that Haar (multi-) wavelets are also available on two- and three-dimensional meshes consisting either of triangles/ tetrahedra or of quadrilaterals / hexahedra (see, e.g. [5] for constructions of any polynomial degree on triangles in two space dimensions). The MLMC and MC require that sparse tensors can be added and multiplied by scalars. By writing a sparse tensor as a linear combination of its basis

$$\begin{aligned} \widehat{u^{(k)}} &= \sum_{|\vec{l}|_1 \le L} \sum_{\vec{j}} \alpha_{\vec{j}}^{\vec{l}} (\psi_{j_1}^{l_1} \otimes \cdots \otimes \psi_{j_k}^{l_k}) \\ &= \sum_{l=0}^{L} \sum_{j} \psi_j^l \otimes \left(\sum_{|\vec{l'}|_1 \le L-l} \sum_{\vec{j'}} \psi_{j'_1}^{l'_1} \otimes \cdots \otimes \alpha_{\vec{j'}}^{\vec{l'}} \psi_{j'_{k-1}}^{l'_{k-1}} \right) \\ &=: \sum_{l=0}^{L} \sum_{j} \psi_j^l \otimes u_j^l \end{aligned}$$

it becomes clear that one can store a sparse tensor by mimicking the hierarchical structure, i.e. `u[l][k] =` u_k^l, with the 'pointwise' operations

```
    (u+v)[l][k] = u[l][k] + v[l][k]
(alpha*u)[l][k] = alpha*u[l][k].
```

The conversion of a numerical solution based on the (piecewise constant) cell averages to the hierarchic, multiresolution representation can be effected on triangulation $\mathcal{T}_\ell$ in $O(\#\mathcal{T}_\ell)$ operations by the so-called pyramid scheme (see, e.g., [] for which efficient implementations are available in image compression, for example.

We emphasize that these concepts are *not* restricted to cartesian grids: for unstructured, triangular or tetrahedral meshes in complex geometries, subspace splittings (4.21) and $L^2(D)$-stable bases for the detail spaces W_ℓ in (4.21) also exist but are, naturally, triangulation dependent and must be generated be recursive agglomeration of cells and recursive SVD's of the corresponding mass matrices. We refer to [38] and [37, Chap. 2] for details.

6.8.2. *Numerical Experiments.* The number of samples M_l and size of the mesh N_l on level l is given by M_0, N_0 and the maximal level L through

$$N_l N_0\, 2^l\ , \quad M_l M_0\, 2^{L-l} \qquad l = 1, \ldots, L\ \ . \tag{6.34}$$

If the MC-Estimator is described in these terms this means $M = N = N_L$. In the following numerical experiments $M_0 = N_0 = 8, L = 6$ and the PDE is evolved until time $T = 0.4$.

All PDEs are solved with the first order FVM using a Rusanov flux and explicit timestepping with the CFL condition $|f'| \leq C\frac{\Delta x}{\Delta t}$ with CFL number $C = 0.45$.

A *sparse tensor product with reduced sparsity* defined by

$$\widetilde{S_L^{(k)}} = \bigoplus_{|\vec{l}|_1 \leq L+3} \bigotimes_{j=1}^{k} W_{l_j} \qquad (l_j \in \{0, \ldots, L\}) \tag{6.35}$$

is also studied along with the full and sparse tensor products defined through (4.22), (4.24), respectively. With $N_0 = 2^3$ the reduced sparsity tensor product is, in fact, a full tensor on the coarsest level and realizes the sparse tensor product construction on a higher level of mesh refinement. Evidently, however, its formation out of Finite Volume solutions is of the same asymptotic complexity as forming the standard sparse tensor product.

The same numerical experiments are performed for three different equations, the few differing parameters will be mentioned accordingly. The first experiment compares MC- and MLMC-Estimates of $E[u^{(2)}], E[\widehat{u}^{(2)}]$ and $E[\widetilde{u}^{(2)}]$. These are maps from $\mathbb{R}^2$ to $\mathbb{R}$. and their graph is plotted as a heat map (i.e. individual values contained in an array are represented as colors).

The second map compares different methods for computing the variance. Four distinct methods are presented. The first is direct estimation. The other three are obtained by observing that $E[u^{(2)}](x,x) - E[u]^2$ is an estimate of the variance. Analogous results hold for sparse and reduced sparsity tensor products instead of a full tensor product.

6.8.3. *Burgers' Equation: Initial Data with Random Amplitude.* The stochastic PDE is Burgers' equation (6.23) with sinusoidal initial conditions where the amplitude is uncertain. More precisely

$$u_0(x,\omega) = X(\omega)\sin(2\pi(x-\alpha)), \quad X \sim \mathcal{U}(0,1) \tag{6.36}$$

where α is some fixed parameter. As seen in Figure 28–30, the quality of the sparse second moment depends on α. In the two cases where the sparse approximation is acceptable the discontinuity in the second moment coincides with a cell boundary. This is not the case for $\alpha = 0.078$, where one can observe some artefacts.

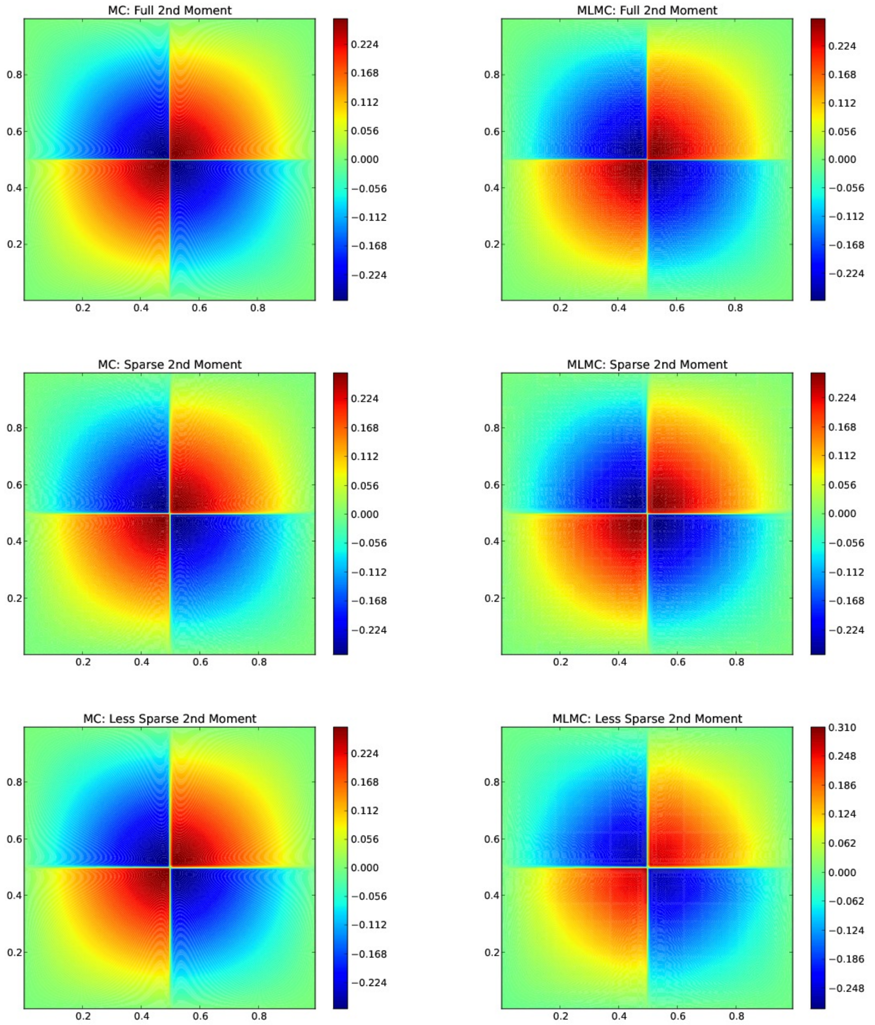


FIGURE 28. Estimated second moment of Burgers' equation (6.36) for $\alpha = 0$.

6.8.4. *Burgers' Equation: Initial Data with Random Phase.* Next we study Burgers' equation with sinusoidal initial data which has uncertain initial phase.

$$u_0(x,\omega) = \sin(2\pi(x - 0.1 \cdot X(\omega))), \quad X \sim \mathcal{U}(0,1) \tag{6.37}$$

The corresponding results for the two-point correlation function with $\alpha = 0.125$ are shown in Figure 32.

6.8.5. *Sparse Tensor Bubble Test.* The next experiment will show that the sparse approximation of a deterministic bubble function

$$f(x) = \max(0, -500(x-\alpha)^2 + 1), \quad \alpha \in [0.3, 0.7]$$

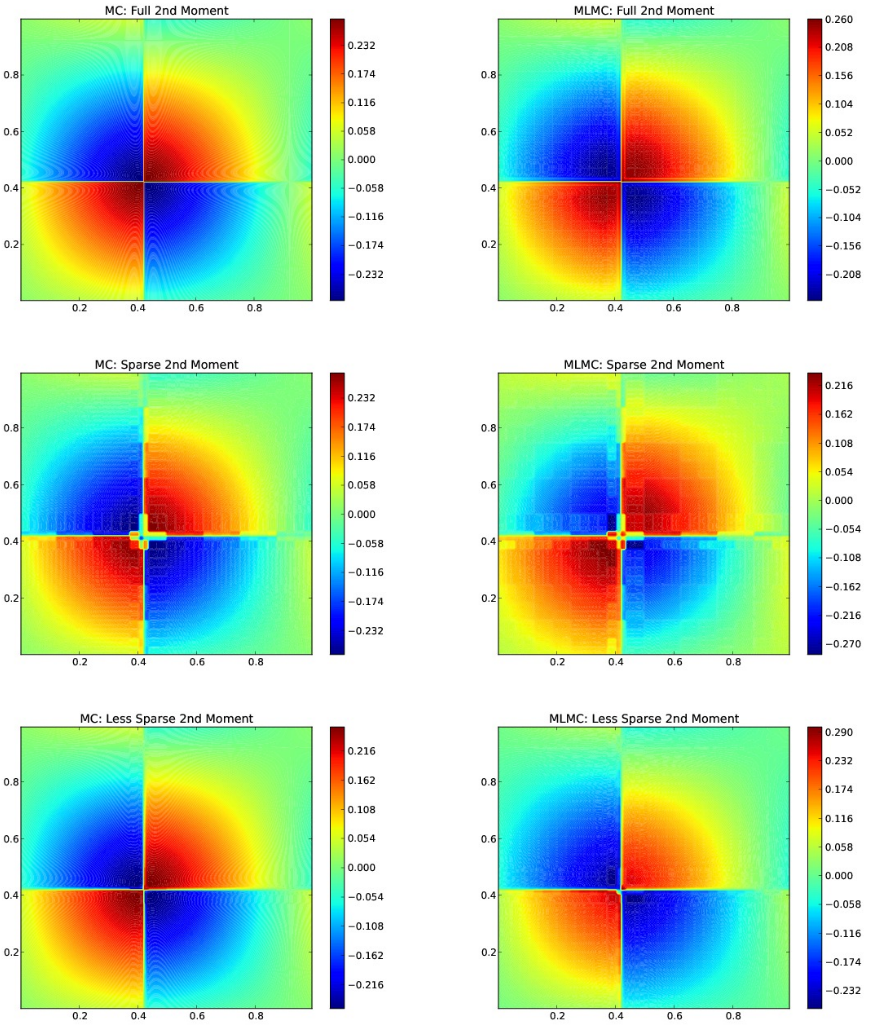


FIGURE 29. Estimated second moment of Burgers' equation 6.36) for $\alpha = 0.078$.

has similar errors as the sparse variance estimate of Burgers' equation with random phase. In particular, therefore, we conclude that *the errors in sparse tensor estimates of the two point correlation functions shown in Figures 28 - 30 are not due to amplifications of Finite Volume discretization errors in the sparse tensor product construction, but occur generically*: they are, indeed, due to the compression of data in the sparse tensor product formation itself.

Let u be the piecewise constant map obtained from f by taking cell averages on a uniform mesh with N cells, $\widehat{u}^{(2)}$ and $\widetilde{u}^{(2)}$ the approximations based on sparse tensor product and reduced sparsity tensor products, as defined above. The cell averages are computed using the midpoint rule on each cell. Figure 34 shows u^2, $\widehat{u}^{(2)}(x,x)$ and $\widetilde{u}^{(2)}(x,x)$ in a single plot. For all mesh sizes $N = 2^l, l = 5, \ldots, 10$ one can

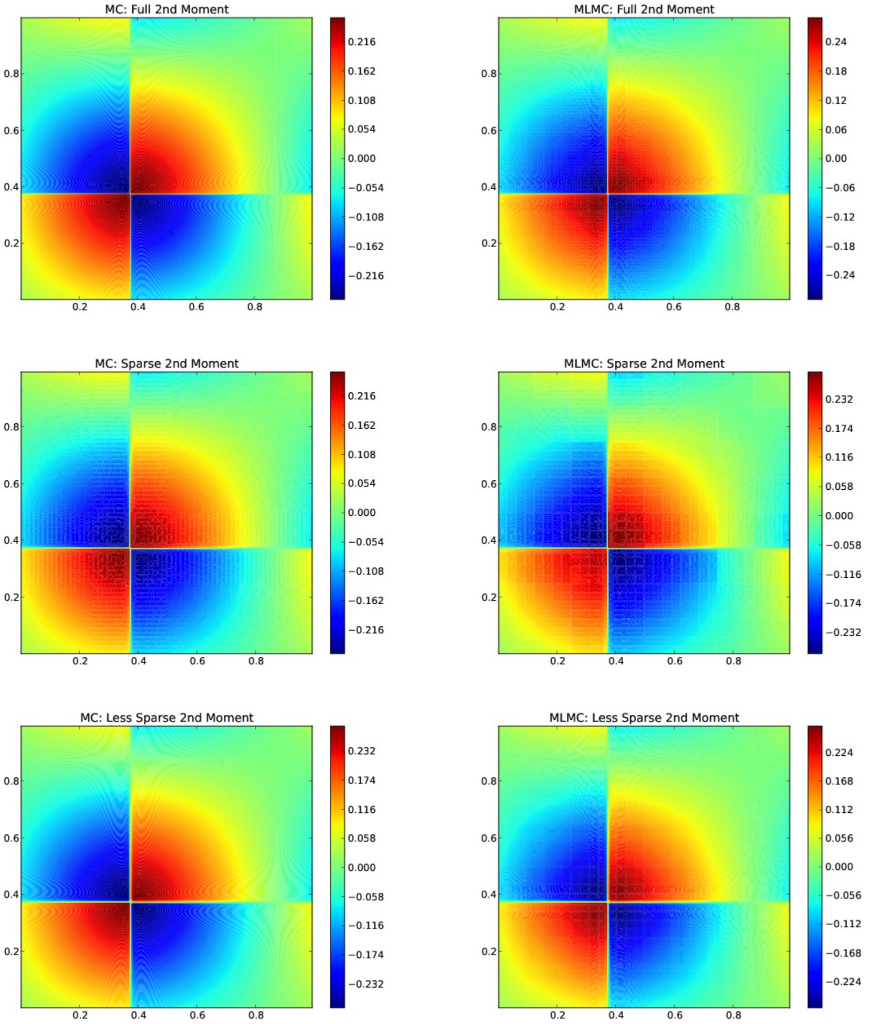


Figure 30. Estimated second moment of Burgers' equation (6.36) for $\alpha = 0.125$.

see negative spikes in both sparse second moment approximations, furthermore the positive peak in the sparse approximation can be way too weak. While the reduced sparsity tensor product approximation also has negative spikes, it is considerably more accurate than the regular sparse tensor product approximation, while still preserving the asymptotic complexity scaling of the sparse tensor product estimate. We also observe in Figure 34 that the use of sparse tensor approximations introduces significant errors which, in the specific example of variance computation, even lead to meaningless outputs such as negative values of the estimated variance.

7. MLMC approximation of probabilities

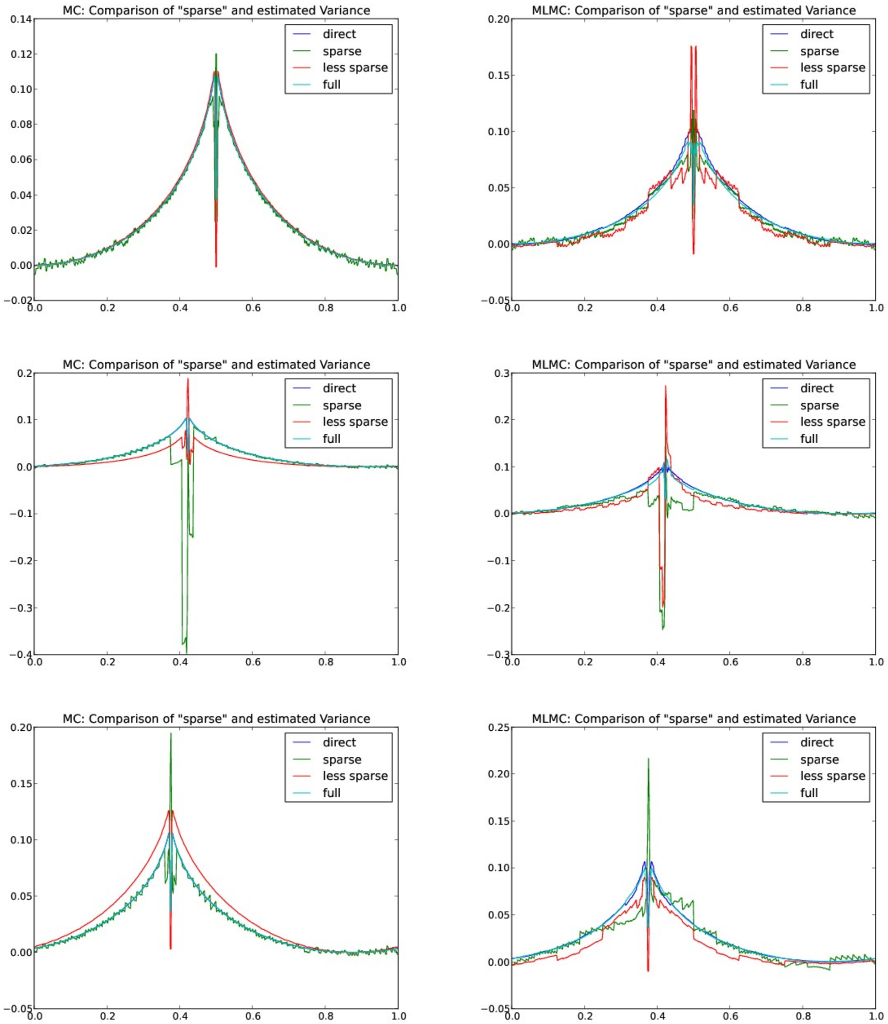


FIGURE 31. Comparison of different variance estimators for Burgers' equation (6.36) : sparse, reduced sparsity, full and direct computation.

7.1. **MLMC Estimation of Probabilities.** So far, in these notes, we addressed the analysis and implementation of MLMC-FVM solvers for the efficient solution of conservation laws with random inputs, and we focused on the efficient computation of first and higher order statistical moments of the random solution(s), such as mean field and variances.

Often, in applications, one is interested in probabilities of certain extremal events $\boldsymbol{E} \in \mathcal{F}$, rather than statistical moments, conditioned on the given random input data. Denoting by $\chi_{\boldsymbol{E}}(\omega)$ the indicator function of $\boldsymbol{E} \in \mathcal{F}$, the probability of

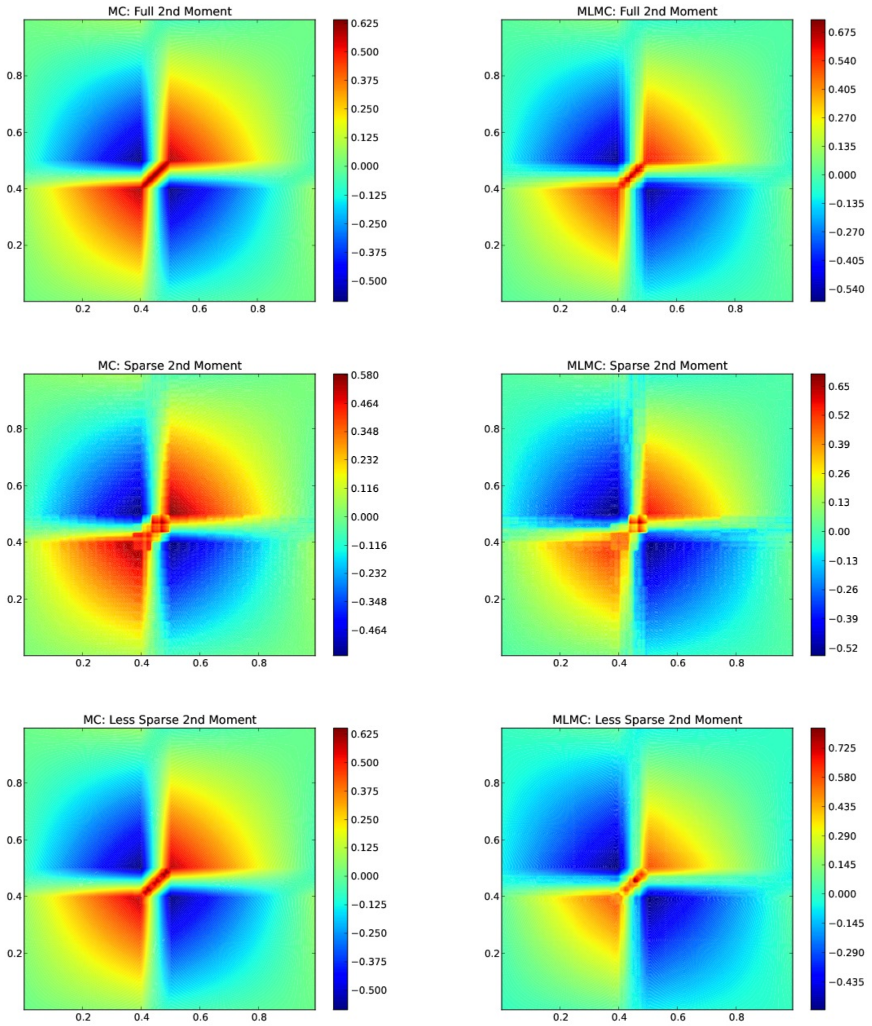


Figure 32. Estimated second moment of Burgers' equation (6.37) for $\alpha = 0.125$.

interest is

$$\mathbb{P}(\boldsymbol{E}) = \int_{\omega\in\Omega} \chi_{\boldsymbol{E}}(\omega) d\mathbb{P}(\omega) \ . \tag{7.1}$$

One of the problems of interest (in order to assess the risk) would be the following: given a fixed sub-domain $\mathbf{C} \subset \mathbf{D} \subset \mathbb{R}^d$, for a fixed time $t \geq 0$, find the *probability* $p(\mathbf{U}) \in [0,1]$ that a certain event $\boldsymbol{E} \in \mathcal{F}$ will occur. Here, we assume that the event $\boldsymbol{E}$ takes the generic form $\boldsymbol{E} = \{\omega \in \Omega : \chi(\mathbf{U}(\cdot, t; \omega)) = 1\}$, where $\chi(\cdot) : \mathbb{R}^m \to \{0,1\}$ is a measurable function. Then, the probability of interest can be expressed as

$$p(\mathbf{U}) := \mathbb{P}\left(\{\chi_{\boldsymbol{E}} = 1\}\right) = \mathbb{E}[\chi(\mathbf{U}(\cdot, t, \omega))] \ . \tag{7.2}$$

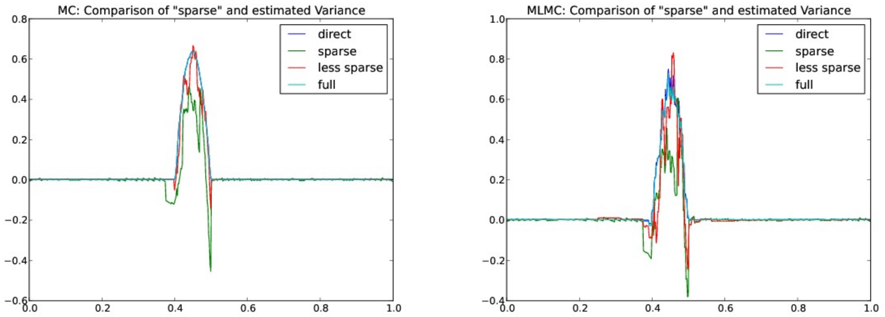


FIGURE 33. Comparison of different variance estimators: sparse, reduced sparsity, full and direct computation.

Rather than developing here a general theory for the multi-level MC computation of such probabilities, we exemplify the main ideas for the Shallow Water Equations with uncertain bottom topography, i.e. (6.13). Here, one is often interested in the event "average water level $\frac{1}{|\boldsymbol{C}|}\int_{x\in\boldsymbol{C}} h(x,t,\omega)$ at time t in subdomain $\mathbf{C}\subset\mathbf{D}$ (e.g. a neighborhood of the shoreline) *exceeds* some given *threshold* $h_{\max}$". Then

$$\chi_{\boldsymbol{E}}(\mathbf{U}(\cdot,t,\omega)) := \chi_{(h_{\max},\infty)}\left(\frac{1}{|\boldsymbol{C}|}\int_{x\in\boldsymbol{C}} h(x,t,\omega)dx\right)$$

and $\mathbb{P}(\boldsymbol{E})$ is given by

$$\mathbb{P}(\boldsymbol{E}) = \mathbb{E}[\chi(\mathbf{U}(\cdot,t,\omega))] = \int_\Omega \chi_{(h_{\max},\infty)}\left(\frac{1}{|\boldsymbol{C}|}\int_{x\in\boldsymbol{C}} h(x,t,\omega)dx\right)d\mathbb{P}(\omega)\ . \tag{7.3}$$

The probability of interest $\mathbb{P}(\boldsymbol{E})$ in (7.3) is an integral w.r.t. the probability measure $\mathbb{P}$. Hence, the integral in (7.3) could be approximated numerically by a Monte-Carlo FVM estimator, i.e. by Monte-Carlo integration of an approximate Finite Volume solution at mesh level ℓ, denoted by $\mathbf{U}_\ell$. The single-level Monte-Carlo Finite Volume estimator for $\mathbb{P}(\boldsymbol{E})$ with M i.i.d. input data samples based on the FVM on mesh level ℓ is given by

$$p_M(\mathbf{U}_\ell) := \frac{1}{M}\sum_{i=1}^{M}\chi(\mathbf{U}_\ell^i(\cdot,t)) = \frac{1}{M}\sum_{i=1}^{M}\chi_{(h_{\max},\infty)}\left(\frac{1}{|\boldsymbol{C}|}\int_{x\in\boldsymbol{C}} h_\ell^i(x,t)dx\right), \tag{7.4}$$

where $h_\ell^i(\cdot,t),\ i=1,\dots,M$ are the MC samples approximated using the FVM scheme at mesh level ℓ. The MLMC-FVM estimator combines MC-FVM estimators (7.4) on the nested family $\{\mathcal{T}_\ell\}_{\ell\geq 0}$ of FVM meshes, as before, and is given by:

$$p^L(\mathbf{U}_L) := p_{M_0}(\mathbf{U}_0) + \sum_{\ell=1}^{L}\Big(p_{M_\ell}(\mathbf{U}_\ell) - p_{M_\ell}(\mathbf{U}_{\ell-1})\Big)\ . \tag{7.5}$$

7.2. **Shallow Water equation in 2d: perturbation of a steady-state.** We consider the setup as in subsubsection 6.3.2, i.e. we are interested in the two-dimensional shallow water equations where the uncertain initial perturbation (6.21) of the water surface is propagating over the uncertain bottom topography described in (6.16) - (6.17) and in (6.18). Our aim is to numerically approximate the probabilities as in (7.3) that the cell averaged water level $h+b$ will exceed the preset

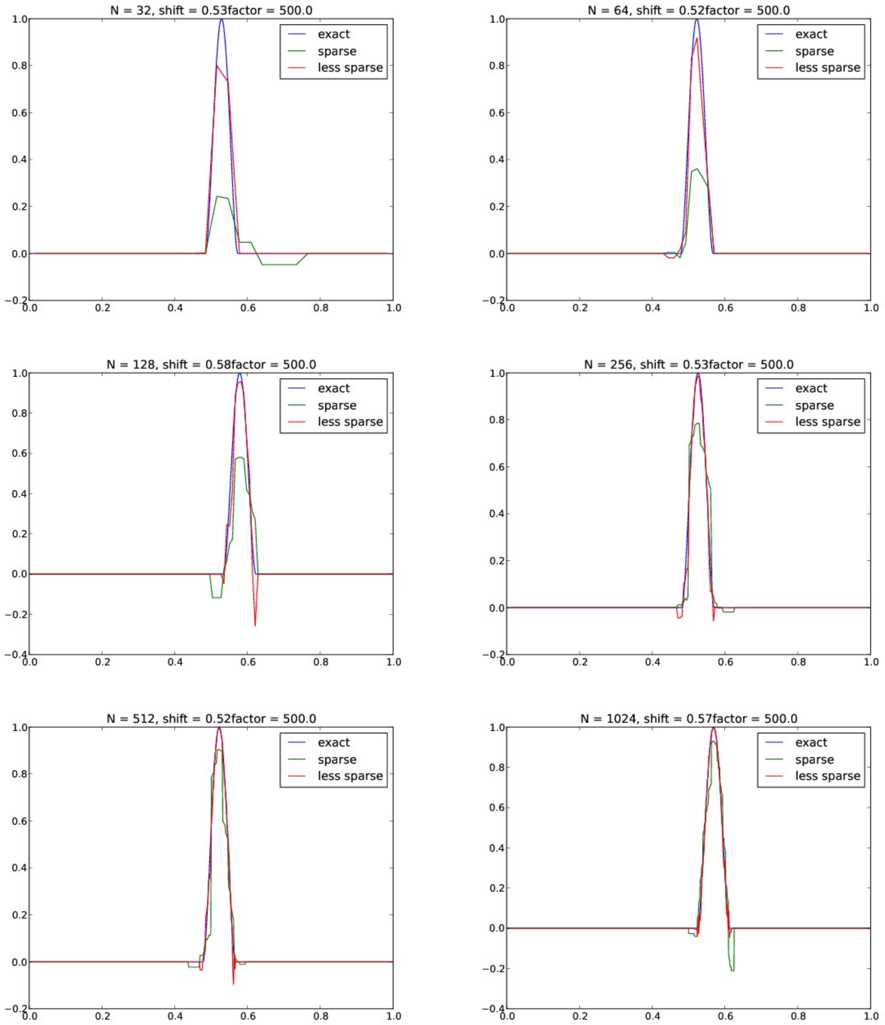
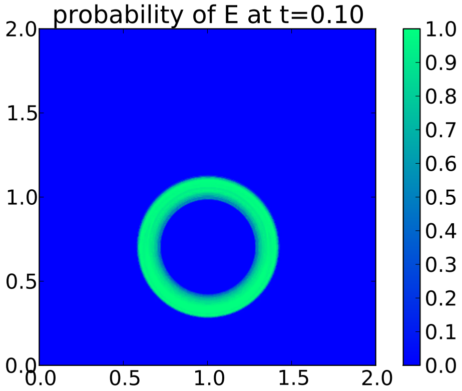


FIGURE 34. Diagonal of the sparse and reduced sparsity tensor product estimates along with the exact value, for different meshes and values for α.

threshold $h_{\max} = 1.002$ in a subdomain $\boldsymbol{C}$ where $\boldsymbol{C} \in \mathcal{T}_L$ denotes a Finite Volume cell.

The results of the numerical simulation using the MLMC-FVM estimator (7.5) for the probability integral (7.3) are given in Figure 35. There are 9 levels ($L = 8$) of FVM mesh resolution with the finest resolution (at the finest level $\ell = L$) being 2048 cells. Rusanov flux with second order accurate well-balanced TECNO [15] reconstruction was used.

Remark 7.1. *Firstly, we observe that $\chi_{\boldsymbol{E}}(\mathbf{U}^i_\ell(\cdot, t))$ can only attain values in $\{0, 1\}$. Then, since the MC-FVM approximation $p_M(\mathbf{U}_\ell)$ of $p(\mathbf{U}_\ell)$ from (7.4) is a convex combination (with equal weights) of values from $\{0, 1\}$, $p_M(\mathbf{U}_\ell)$ attains values in*



L	M_L	grid size	CFL	cores	runtime	efficiency
8	16	2048x2048	0.45	128	1:42:33	94.7%

FIGURE 35. MLMC-FVM approximation of the probabilities (7.3) that the random water level $h(\cdot,t,\omega)+b(\cdot,\omega)$ will exceed (at time $t=0.1$) the preset maximal threshold $h_{\max}=1.002$.

the interval $[0,1]$*, which is consistent with the fact that* $p_M(\mathbf{U}_\ell)$ *is approximation of the exact* probability*. However, such bounded range of values is no longer valid for the* Multi-Level *MC-FVM approximation* $p^L(\mathbf{U}_L)$ *in* (7.5)*.*

As a counter-example to prove this claim, consider only two levels, i.e. $L=1$*, one sample on the level* $\ell=0$*, i.e.* $M_0=1$*, and two samples on the finer level* $\ell=1$*, i.e.* $M_1=2$*. The coarsest (one-dimensional) mesh level is assumed to have only one cell, i.e.* $\mathcal{T}_0=\{C_1^0\}$*, and the finer mesh level is assumed to have two equal cells, i.e.* $\mathcal{T}_1=\{C_1^1,C_2^1\}$*. Then, assume we have random samples* $\omega_1^0,\omega_1^1,\omega_2^1$ *for random input data. Furthermore, we assume that the event values for these samples on different mesh resolutions are given by:*

$$\begin{aligned}
&\chi_{\boldsymbol{E}}(\mathbf{U}_0(C_1^0,\omega_1^0))=1, && \\
&\chi_{\boldsymbol{E}}(\mathbf{U}_1(C_1^1,\omega_1^1))=1, && \chi_{\boldsymbol{E}}(\mathbf{U}_1(C_1^1,\omega_2^1))=1,\\
&\chi_{\boldsymbol{E}}(\mathbf{U}_1(C_2^1,\omega_1^1))=1, && \chi_{\boldsymbol{E}}(\mathbf{U}_1(C_2^1,\omega_2^1))=0,\\
&\chi_{\boldsymbol{E}}(\mathbf{U}_0(C_2^1,\omega_1^1))=1, && \chi_{\boldsymbol{E}}(\mathbf{U}_0(C_2^1,\omega_2^1))=0.
\end{aligned}$$

Then, the corresponding MC-FVM estimates $p_{M_0}(\mathbf{U}_0)$*,* $p_{M_1}(\mathbf{U}_1)$ *and* $p_{M_1}(\mathbf{U}_0)$ *are*

$$p_{M_0}(\mathbf{U}_0)=1,\quad p_{M_1}(\mathbf{U}_0(C_1^1))=1,\quad p_{M_1}(\mathbf{U}_0(C_2^1))=\frac{1}{2},\quad p_{M_1}(\mathbf{U}_0)=\frac{1}{2}1=\frac{1}{2},$$

which lead to the following MLMC-FVM estimate (7.5) *for each cell* C_1^1,C_2^1

$$p^L(\mathbf{U}(C_1^1))=1+1-\frac{1}{2}=1\frac{1}{2},\qquad p^L(\mathbf{U}(C_2^1))=1+\frac{1}{2}-\frac{1}{2}=1. \tag{7.6}$$

Clearly $p^L(\mathbf{U}(C_1^1))>1$*. The counter-example resulting in* $p^L(\mathbf{U}(C_1^1))<0$ *can be obtained analogously. Notice, that even if we* reuse *samples from the coarsest mesh level* $\ell=0$ *in the finer mesh level* $\ell=1$*, i.e. we set* $\omega_1^1=\omega_1^0$*, the counter-example is still consistent, since* $\omega_1^1\neq\omega_2^1$*.*

The above considerations do not mean that the MLMC-FVM approximation of $p(\mathbf{U}_\ell)$ is wrong or inconsistent with the exact value; as $L \to \infty$ the MLMC-FVM approximation $p^L(\mathbf{U}_L)$ still converges *to the exact value $p(\mathbf{U}_L)$. In order to avoid confusion, however, in Figure 35 the approximated values of $p^L(\mathbf{U}_L)$ were clipped to the interval* $[0, 1]$. *Analogously, in variance plots throughout this manuscript, the variance was clipped to* $[0, \infty)$. *The consequences of such clipping are numerically* negligible*; however, the awareness of this* non-monotonicity *of the MLMC-FVM estimator is essential for the interpretation of the numerical results. In particular; it is important to realize that in MLMC-FVM such undershoots might occur, which are* not *related to numerical instability, bias of the estimators, inappropriate FVM schemes or other implementational errors.*

8. Conclusion

The issue of uncertainty quantification for physical and engineering applications that are modeled by hyperbolic systems of balance laws, has gained increasing attention in recent years. The inputs to hyperbolic systems of balance laws such as initial data, boundary conditions, source terms as well as fluxes are in general, prone to measurement errors and are therefore in principle uncertain. This data uncertainty propagates into the solution, making the design of efficient and robust methods for the efficient numerical quantification of uncertainty, a task that is of utmost significance in CSE.

In these notes, we have presented recent results on the design, analysis and implementation of efficient statistical sampling methods of the Monte Carlo (MC) and Multi-Level Monte Carlo (MLMC) type for quantifying uncertainty in the solutions of systems of random balance laws with random input data. The main conclusions from the results in these notes are as follows:

- Uncertain inputs such as random initial data, sources and fluxes are modeled within the classical probabilistic framework of Kolmogorov. The corresponding notion of *random entropy solutions* was introduced and shown to be wellposed for scalar multi-dimensional conservation laws with random initial data and with random fluxes. Furthermore, statistical regularity of the random entropy solutions was mathematically described.
- The MC-FVM and MLMC-FVM algorithms for systems of balance laws were presented. Efficient high-resolution finite volume schemes were used for the spatio-temporal discretizations of the balance laws and were combined with the MC and MLMC algorithms. The convergence and complexity analysis of the resulting schemes (in scalar case) were presented. In particular, we showed that the MLMC Finite Volume Method had the same asymptotic complexity (up to a logarithmic term) as a single deterministic finite volume solve.
- Details of implementation of the MC and MLMC Finite Volume Methods for systems of balance laws were described briefly. A static load balancing algorithm was presented. This static load balancing algorithm was demonstrated to scale the MLMC Finite Volume Methods with near optimal parallel efficiency to several thousands of parallel processing units.
- A large number of numerical experiments were presented to illustrate the efficiency of the MLMC-FVM method. These experiments included Euler and MHD equations with uncertain initial data (initial data as well as fluxes),

shallow water equations with uncertain initial data and uncertain bottom topography, Buckley-Leverett equations with uncertain relative permeabilities and the compressible Euler equations with uncertain equations of state.

- A sparse tensor discretization framework was introduced to efficiently compute k-point statistical correlation functions in several space dimensions for the random entropy solutions. Preliminary numerical results illustrating this method were presented, and the errors introduced by sparse tensorization (which is essential for computational efficiency) were discussed.
- We introduced a novel non-intrusive technique to use the MLMC algorithm for computing approximate probabilities of statistical events of engineering interest, such as probabilities of extremal events in engineering risk and failure analyses based on random entropy solutions.

The main purpose of these notes was to demonstrate that the MLMC statistical sampling method is a powerful tool in the context of computational uncertainty quantification for hyperbolic systems of balance laws with random input data. Its advantages include:

- The method is completely non-intrusive. It can be readily used in conjunction with any spatio-temporal hierarchical discretization of the underlying conservation (balance) laws. In the present notes, we combined the MLMC method with high-resolution Finite Volume Methods. We hasten to add, however, that DG (discontinuous Galerkin) discretizations can also be used as spatio-temporal solvers.
- The MLMC-FVM approach is very flexible and can be used for different types of uncertain inputs such as random initial data, source terms or flux functions.
- The method is robust with respect to very low regularity (presence of discontinuities) of the underlying random entropy solutions.
- The method can deal with a very large number of sources of uncertainty. For instance, the computation for shallow water equations with uncertain bottom topography involved approximately 1000 sources of uncertainty. To the best of our knowledge, no other method (particularly deterministic methods such as stochastic Galerkin or stochastic collocation) can handle efficiently this many sources of uncertainty (ie., high "stochastic dimensions") with solutions of low regularity and possibly nonsmooth dependence on random input parameters.
- Plain MC simulations based on *sample averages* which are convex combinations of numerical solutions. Therefore, plain MC-FVM estimates naturally inherit non-oscillation, discrete maximum and monotonicity properties of the underlying FV discretization schemes. Since MLMC-FVM estimators are not based on convex combinations of numerical FVM solutions, in these (desirable) properties of discretizations schemes are lost in these estimators. The efficiency gain afforded by the multi-level methodology, however, is so substantial that it outweighs these shortcomings, but awareness of these effects is warranted in the interpretation of simulation results.
- All properties of the MLMC methods mentioned above are directly inherited from the standard Monte Carlo methods; however, our analysis for scalar balance laws indicates that MLMC methods offer a better asymptotic rate of convergence. In particular, in all our numerical experiments we

found that already in the engineering range of accuracy MLMC-FVM methods are several orders of magnitude more efficient than plain MC methods.

- When coupled with efficient implementation on massively parallel hardware architectures, the MLMC-FVM methods can handle complex flow problems with random input data.
- MLMC-FVM is also amenable to *algorithmic fault-tolerant* parallelization, where several *lost MC samples* (due to node or network failures) could be completely *ignored*. In [34], assuming that the *expected* number of failures is not too large, the MLMC-FVM estimate is proven to converge (i.e., the *expected* MLMC-FVM error converges) even though only the remaining batch of MC "survived" samples are used to assemble the MLMC-FVM estimator.

Given these advantages, the MLMC Finite Volume Method appears as a powerful general purpose technique for quantifying uncertainty in solutions of random systems of balance laws in engineering practice.

9. Acknowledgement

The authors wish to express their gratitude to Mr. Luc Grosheintz, a student in the ETH Zürich MSc Applied Mathematics curriculum for performing the numerics for the sparse two-point correlation computations reported in Section 6.8.

The authors thank the systems support at ETH Zürich parallel Compute Cluster BRUTUS [49] for their support in the production runs for the present paper, and the staff at the Swiss National Supercomputing center (CSCS) [50] at Lugano for their assistance in the large scale Euler and MHD simulations.

References

[1] ALSVID. Available from `http://folk.uio.no/mcmurry/amhd`.

[2] ALSVID-UQ. Available from `http://www.sam.math.ethz.ch/alsvid-uq`.

[3] R. Abgrall. *A simple, flexible and generic deterministic approach to uncertainty quantification in non-linear problems.* Rapport de Recherche, INRIA, 2007.

[4] K. Aziz and A. Settari. *Fundamentals of petroleum reservoir simulation.* Applied Science Publishers, London, 1979.

[5] A. Barth, Ch. Schwab and N. Zollinger. *Multilevel MC Method for Elliptic PDEs with Stochastic Coefficients.* Numerische Mathematik Volume 119, Number 1, pages 123-161 (2011).

[6] T. J. Barth. *Numerical methods for gas-dynamics systems on unstructured meshes.* An Introduction to Recent Developments in Theory and Numerics of Conservation Laws, pp. 195–285. Lecture Notes in Computational Science and Engineering volume 5, Springer, Berlin. Eds: D. Kroner, M. Ohlberger, and Rohde, C., 1999.

[7] P. D. Bates, S. N. Lane and R. I. Ferguson. *Parametrization, Validation and Uncertainty analysis of CFD models of fluvial and flood hydraulics in natural environments.* Computational Fluid Dynamics: Applications in environmental hydraulics, John Wiley and sons, 2005, 193-212.

[8] Q. Y. Chen, D. Gottlieb and J. S. Hesthaven. *Uncertainty analysis for steady flow in a dual throat nozzle.* J. Comput. Phys, **204**:378-398, 2005.

[9] B. Cockburn and C-W. Shu. *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework.* Math. Comput., 52, 1989, 411–435.

[10] Constantine M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics (2nd Ed.).* Springer Verlag (2005).

[11] Josef Dick, Franes Y. Kuo and Ian H. Sloan. *High dimensional integration: the Quasi Monte-Carlo way.* Acta Numerica (2013) (to appear).

[12] R. Eymard, T. Gallouët, and R. Herbin. *Finite volume methods*, in Handbook of numerical analysis, Vol. **VII**, pp. 713–1020, North-Holland, Amsterdam, (2000).

[13] P. F. Fisher and N. J. Tate. *Causes and consequences of error in digital elevation models.* Prog. in Phy. Geography, 30 (4), 2006, 467-489.

[14] G. Fishman. *Monte Carlo.* Springer, 1996.

[15] U.S. Fjordholm, S. Mishra, and E. Tadmor. *Well-balanced, energy stable schemes for the shallow water equations with varying topology.* J. Comput. Phys., 230, 5587-5609, 2011.

[16] F. Fuchs, A. D. McMurry, S. Mishra, N. H. Risebro and K. Waagan. *Approximate Riemann solver based high-order finite volume schemes for the Godunov-Powell form of ideal MHD equations in multi-dimensions.* Comm. Comput. Phys., **9**:324-362, 2011.

[17] M. Giles. *Improved multilevel Monte Carlo convergence using the Milstein scheme.* Preprint NA-06/22, Oxford computing lab, Oxford, U.K, 2006.

[18] M. Giles. *Multilevel Monte Carlo path simulation.* Oper. Res., **56**:607-617, 2008.

[19] E. Godlewski and P.A. Raviart. *Hyperbolic Systems of Conservation Laws.* Mathematiques et Applications, Ellipses Publ., Paris (1991).

[20] S. Gottlieb, C. W. Shu and E. Tadmor. *High order time discretizations with strong stability property.* SIAM. Review, **43**:89 - 112, 2001

[21] A. Harten, B. Engquist, S. Osher and S. R. Chakravarty. *Uniformly high order accurate essentially non-oscillatory schemes.* J. Comput. Phys., 1987, 231-303.

[22] S. Heinrich. *Multilevel Monte Carlo methods.* Large-scale scientific computing, Third international conference LSSC 2001, Sozopol, Bulgaria, 2001, Lecture Notes in Computer Science, Vol **2170**, Springer Verlag (2001), pp. 58-67.

[23] P. L'Ecuyer and F. Panneton. *Fast Random Number Generators Based on Linear Recurrences Modulo 2: Overview and Comparison.* Proceedings of the 2005 Winter Simulation Conference, 110–119, IEEE press, 2005.

[24] P. L'Ecuyer and F. Panneton. *Fast Random Number Generators Based on Linear Recurrences Modulo 2.* ACM Trans. Math. Software, **32**:1-16, 2006.

[25] R.A. LeVeque. *Numerical Solution of Hyperbolic Conservation Laws.* Cambridge Univ. Press 2005.

[26] R. LeVeque, D. George and M. Berger. *Tsunami modeling with adaptively refined finite volume methods.* Acta Numerica, 20, 211-289, 2011.

[27] G. Lin, C.H. Su and G. E. Karniadakis. *The stochastic piston problem.* PNAS **101**:15840-15845, 2004.

[28] X. Ma and N. Zabaras. *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations.* J. Comp. Phys, 228, 3084-3113, 2009.

[29] M. Matsumoto and T. Nishimura. *Mersenne Twister: a 623-dimensionally equidistributed uniform pseudorandom number generator.* ACM Trans. Modeling and Computer Simulation 8. (Jan. 1998) 3–30.

[30] S. Mishra and Ch. Schwab. *Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data.* Math. Comp. **280(81):**1979–2018, 2012.

[31] S. Mishra, N.H. Risebro, Ch. Schwab and S. Tokareva. *Stochastic Finite Volume Method for scalar conservation Laws with random flux.* Technical Report, SAM, 2012.

[32] S. Mishra, Ch. Schwab and J. Šukys. *Multi-level Monte Carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions.* J. Comp. Phys., **231(8):**3365–3388, 2012.

[33] S. Mishra, Ch. Schwab, and J. Šukys. *Multi-level Monte Carlo Finite Volume methods for shallow water equations with uncertain topography in multi-dimensions.* SIAM J. Sci. Comput., **34(6)**, pp. B761–B784, 2012.

[34] S. Pauli and P. Arbenz and Ch. Schwab. *Intrinsic fault tolerance of multi level Monte Carlo methods.* Report 2012-24, Seminar für Angewandte Mathematik ETH Zürich, 2012. Also available from `http://www.sam.math.ethz.ch/sam_reports/index.php?id=2012-24`.

[35] G. Poette, B . Després and D. Lucor. *Uncertainty quantification for systems of conservation laws.* J. Comput. Phys. **228**:2443-2467, 2009.

[36] G.D. Prato and J. Zabcyk, *Stochastic Equations in infinite dimensions*, Cambridge Univ. Press (1991).

[37] G. Schmidlin. *Fast solution algorithms for integral equations in* $\mathbb{R}^3$ PhD dissertation ETH Zürich No. 15016 (2003).

[38] G. Schmidlin and Ch. Schwab. *Wavelet Galerkin BEM on unstructured meshes by aggregation.* LNCSE **20** (2002) 369-278, Springer Lecture Notes in CSE, Springer Verlag, Berlin Heidelberg New York (2002).

[39] Ch. Schwab and S. Tokareva. *High order approximation of probabilistic shock profiles in hyperbolic conservation laws with uncertain initial data.* Report 2011-53, Seminar für Angewandte Mathematik ETH Zürich (to appear in Math. Mod. Numer. Analysis 2013). Also available from `http://www.sam.math.ethz.ch/reports/2011/53`.

[40] C. W. Shu and S. Osher. *Efficient implementation of essentially non-oscillatory schemes - II.* J. Comput. Phys., 83, 1989, 32 - 78.

[41] C. W. Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws.* ICASE Technical report, NASA, 1997.

[42] J. Šukys, S. Mishra, and Ch. Schwab. *Static load balancing for multi-level Monte Carlo finite volume solvers.* PPAM 2011, Part I, LNCS **7203**, pp. 245–254. Springer, Heidelberg (2012).

[43] J. Tryoen, O. Le Maitre, M. Ndjinga and A. Ern. *Intrusive projection methods with upwinding for uncertain non-linear hyperbolic systems.* Preprint, 2010.

[44] T. von Petersdorff and Ch. Schwab. *Sparse Finite Element Methods for Operator Equations with Stochastic Data*, Applications of Mathematics **51**(2) (2006) 145-180.

[45] X. Wan and G. E. Karniadakis. *Long-term behaviour of polynomial chaos in stochastic flow simulations.* Comput. Meth. Appl. Mech. Engg. **195**:5582-5596, 2006.

[46] B. P. Welford. *Note on a Method for Calculating Corrected Sums of Squares and Products.* Technometrics, **4**:419-420, 1962.

[47] J. A. S. Witteveen, A. Loeven, H. Bijl *An adaptive stochastic finite element approach based on Newton-Cotes quadrature in simplex elements.* Comput. Fluids, 38, 1270-1288, 2009.

[48] D. Xiu and J. S. Hesthaven. *High-order collocation methods for differential equations with random inputs.* SIAM J. Sci. Comput., 27, 1118-1139, 2005.

[49] *Brutus,* ETH Zürich, `de.wikipedia.org/wiki/Brutus_(Cluster)`.

[50] *Cray XE6,* Swiss National Supercomputing Center (CSCS), Lugano, `www.cscs.ch`.

[51] *MPI: A Message-Passing Interface Standard.* Version 2.2, 2009, available from: `http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf`.

[52] *Open MPI: Open Source High Performance Computing.* Available from `http://www.open-mpi.org/`.

(Siddhartha Mishra)
Seminar for Applied Mathematics
ETH
HG G. 57.2,
Rämistrasse 101, Zürich, Switzerland.
E-mail address: `smishra@sam.math.ethz.ch`

(Christoph Schwab)
Seminar for Applied Mathematics
ETH
HG G. 57.1,
Rämistrasse 101, Zürich, Switzerland.
E-mail address: `christoph.schwab@sam.math.ethz.ch`

(Jonas Šukys)
Seminar for Applied Mathematics
ETH
HG E 62.1,
Rämistrasse 101, Zürich, Switzerland.
E-mail address: `jonas.sukys@sam.math.ethz.ch`

Research Reports

No.	Authors/Title
12-08	*S. Mishra, Ch. Schwab and J. Šukys* Monte Carlo and multi-level Monte Carlo finite volume methods for uncertainty quantification in nonlinear systems of balance laws
12-07	*A. Hiltebrand and S. Mishra* Entropy stable shock capturing space-time discontinuous Galerkin schemes for systems of conservation laws
12-06	*R. Hiptmair, A. Moiola and I. Perugia* Trefftz discontinuous Galerkin methods for acoustic scattering on locally refined meshes
12-05	*C. Winteler, R. Käppeli, A. Perego, A. Arcones, N. Vasset, N. Nishimura, M. Liebendörfer and F.-K. Thielemann* Magneto-rotationally driven Supernovae as the origin of early galaxy r-process elements?
12-04	*P. Grohs* Intrinsic localization of anisotropic frames
12-03	*P. Grohs* Geometric multiscale decompositions of dynamic low-rank matrices
12-02	*D. Kressner and C. Tobler* htucker - A Matlab toolbox for tensors in hierarchical Tucker format
12-01	*F.Y. Kuo, Ch. Schwab and I.H. Sloan* Quasi-Monte Carlo methods for high dimensional integration - the standard (weighted Hilbert space) setting and beyond
11-72	*P. Arbenz, A. Hiltebrand and D. Obrist* A parallel space-time finite difference solver for periodic solutions of the shallow-water equation
11-71	*M.H. Gutknecht* Spectral deflation in Krylov solvers: A theory of coordinate space based methods
11-70	*S. Mishra, Ch. Schwab and J. Šukys* Multi-level Monte Carlo finite volume methods for shallow water equations with uncertain topography in multi-dimensions
11-69	*Ch. Schwab and E. Süli* Adaptive Galerkin approximation algorithms for partial differential equations in infinite dimensions
11-68	*A. Barth and A. Lang* Multilevel Monte Carlo method with applications to stochastic partial differential equations