Data-driven Multilevel Monte Carlo methods
UQ via Forecasting

University of Stuttgart

Computational Methods for Uncertainty Qunatification
Institut für Angewandte Analysis und Numerische Simulation
Allmandring 5b, D-70569 Stuttgart, Germany
lukas.brencher@ians.uni-stuttgart.de

Lukas Brencher

Motivation

The input data to numerous phenomena in the natural sciences, that can be described by partial differential equations (PDEs), is not known precisely or is not deterministic. Thus, an extension to random PDEs (rPDEs) is necessary to capture the stochastic nature of the data.
We consider the following rPDE, equipped with Dirichlet boundary conditions, for a spatial domain $\mathbb{X} = [0,1]^2$ and the time interval $\mathbb{T} = [0, 0.5]$:

$$\frac{\mathrm{d}}{\mathrm{d}t} u(\mathbf{x}, t, \omega) - \nabla_x \cdot (a(\mathbf{x}, \omega) \nabla_x u(\mathbf{x}, t, \omega)) = g(\mathbf{x}) \qquad \text{for all } (\mathbf{x}, t) \in \mathbb{X}_{\mathbb{T}}$$

Here, $a(\mathbf{x}, \omega)$ is a random diffusion coefficient given by

$$a(\mathbf{x}, \omega) = \bar{a}(\mathbf{x}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \phi_i(\mathbf{x}) \gamma_i(\omega) \ ,$$

where $((\lambda_i, \phi_i), i \in \mathbb{N})$ is a sequence of eigenpairs of a given covariance operator and $(\gamma_i, i \in \mathbb{N})$ is a sequence of independent uniformly distributed random variables. The solution u is a distribution over $\mathbb{X}_{\mathbb{T}}$. We are interested in the computation of relevant statistics of the solution, i.e., moments of u.

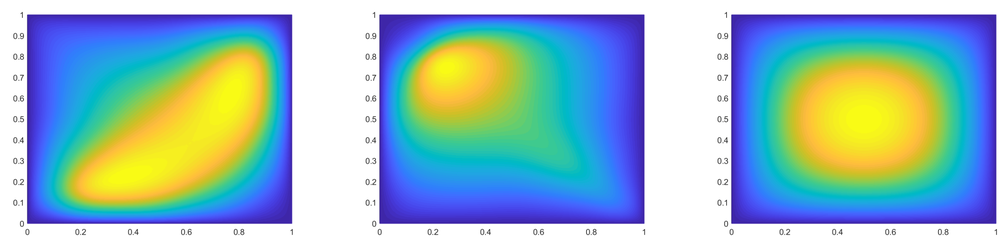


Figure 1: Parameter dependency of the solution to the rPDE.

Multilevel Monte Carlo (MLMC) estimation

Let $(u_l, l \in \mathbb{N})$ be a sequence of discretizations converging to the exact solution. For given discretization $l \in \mathbb{N}$ and number of samples $N_l \in \mathbb{N}$ the singlelevel Monte Carlo estimator is given by

$$E_{N_l}(u_l) = \frac{1}{N_l} \sum_{i=1}^{N_l} u_l^{(i)} \ .$$

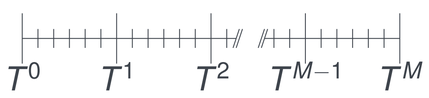
We aim to approximate the stochastic moments of the solution to the rPDE via the Multilevel Monte Carlo estimator of $\mathbb{E}(u_L)$

$$E^L(u_L) = E_{N_0}(u_0) + \sum_{l=1}^{L} E_{N_l}(u_l - u_{l-1}) \ .$$

Here, $N_0 > ... > N_L$ are the number of samples computed on each level. The general idea of multilevel Monte Carlo methods is to compute many samples on a coarse discretization (N_0) and only few samples on an accurate discretization (N_L). The overall computational costs of such a computation is dominated by the computation of few samples on the most accurate discretization.

Data-driven time parallelism via Forecasting

In time parallelism, the temporal domain is divided into subdomains, for which initial values are introduced. This yields subproblems, which can be solved in parallel.



The Parareal framework [4] is defined by two time propagators acting on different time discretizations:

- $\mathcal{G}(\mathbf{u}^i, T^i, T^j)$ providing a coarse approximation at T^j for $0 \leq i \leq j \leq M$ (serial computation)
- $\mathcal{F}(\mathbf{u}^i, T^i, T^j)$ providing a fine approximation at T^j for $0 \leq i \leq j \leq M$ (parallel computation)

where u^i denotes the numerical solution at coarse time step T^i. While the parallel computation leads to *jumps*, these are corrected iteratively until convergence for parareal iteration k and subinterval m:

$$\mathbf{u}_{k+1}^{m+1} = \mathcal{F}(\mathbf{u}_k^m, T^m, T^{m+1}) + \underbrace{\mathcal{G}(\mathbf{u}_{k+1}^m, T^m, T^{m+1}) - \mathcal{G}(\mathbf{u}_k^m, T^m, T^{m+1})}_{\text{jump}}$$

The data-driven time parallelism introduced in [2] employs the forecasting method from [3] as a coarse propagator. This forecasting method aims to predict the solution at future time steps. Assuming a *time-evolution basis* is available, we can compute the forecast by solving a least-squares problem based on previously computed time steps.
The data-driven time parallelism is defined by employing a **local forecast** as a coarse propagator and a **global version** as the parareal initialization, which computes the first initial values for the subproblems. As the fine temporal propagator the **backward Euler** time integration scheme is applied.

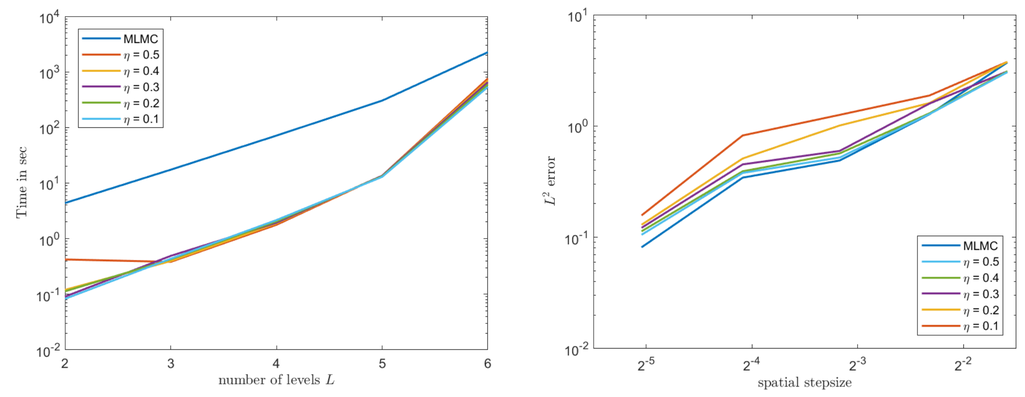
Data-driven multilevel Monte Carlo

We aim to decrease the computational costs of the MLMC estimation by computing the samples with data-driven time parallelism. Therefore, we define a *splitting parameter* $\eta \in [0, 1]$, which defines a set of $N_l^\eta := \lceil \eta N_l \rceil$ training samples to compute the time evolution bases for the forecasting procedure via a singular value decomposition. Afterwards, the remaining $N_{DD,l}^\eta = N_l - N_l^\eta$ samples can be computed via data-driven time parallelism. With these two types of samples, we can define the data-driven singlelevel Monte Carlo estimator as

$$E_N^\eta(u_l) := \frac{1}{N_l^\eta} \sum_{i=1}^{N_l^\eta} u_l^{(i)} + \frac{1}{N_{DD,l}^\eta} \sum_{i=1}^{N_{DD,l}^\eta} u_{DD,l}^{(i)}$$

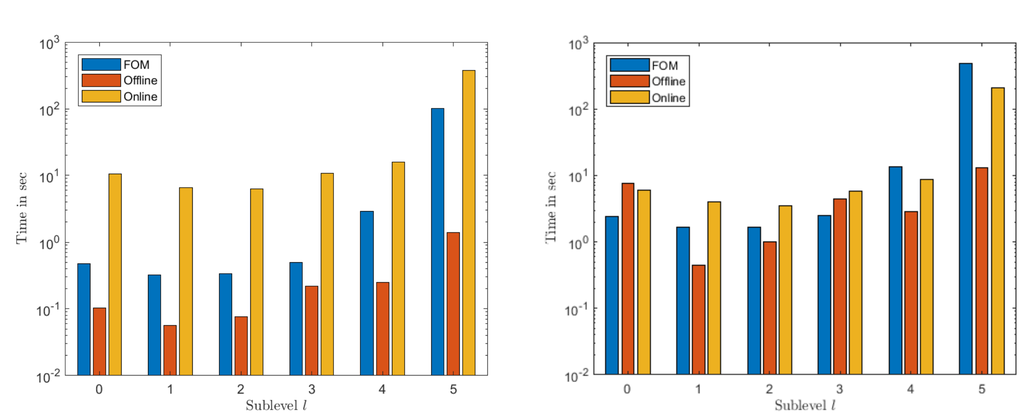
and the corresponding data-driven multilevel Monte Carlo estimator as

$$E^{L,\eta}(u_L) = E_{N_0}^\eta(u_0) + \sum_{l=1}^{L} E_{N_l}^\eta(u_l - u_{l-1}) \ .$$



(a) Walltime. **(b)** L^2-error.

Figure 2: Walltime and L^2-error of the data-driven MLMC and standard MLMC method.



(a) $\eta = 0.1$ **(b)** $\eta = 0.5$

Figure 3: Composition of the walltime of the data-driven MLMC computation, depending on the splitting parameter η.

Conclusions

The data-driven MLMC method is able to improve the computation time compared to the standard MLMC method. It consists of

- **Offline**: Compute Monte Carlo samples via some standard time integration scheme
- **Training**: Compute POD basis from left singular vectors and time evolution bases from right singular vectors
- **Online**: Compute remaining samples via data-driven time parallelism with global forecast initialization and local forecast coarse propagation

References

[1] L. Brencher, A. Barth, B. Haasdonk, and K. Carlberg.
Data-driven multilevel Monte Carlo method via forecasting.
work in progress, 2018.

[2] K. Carlberg, L. Brencher, B. Haasdonk, and A. Barth.
Data-driven time parallelism via forecasting.
submitted to SIAM SISC, 2016.

[3] K. Carlberg, J. Ray, and B. van Bloemen Waanders.
Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting.
Computer Methods in Applied Mechanics and Engineering, 289:79–103, 2015.

[4] J. Lions, Y. Maday, and G. Turinici.
A "parareal" in time discretization of PDE's.
Comptes Rendus de l'Academie des Sciences Series I Mathematics, 332(7):661–668, 2001.

SimTech
www.simtech.uni-stuttgart.de

ians