Controlling error in multi-level approximations of stochastic PDEs

Raphaël Bulle [1,2]
Franz Chouly [2], Alexei Lozinski [2]
Stéphane P.A. Bordas [1], Jack S. Hale [1]

University of Luxembourg [1]
Université de Bourgogne Franche-Comté [2]

April 10, 2019

The ASSIST project has received funding from the University of Luxembourg Internal Research Project scheme.
The DRIVEN project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 811099.

Table of contents

- Model problem introduction
- First approach: standard Monte-Carlo
- First approach: error control
- Second approach: multi-level Monte-Carlo
- Second approach: error control
- MLMC Algorithm
- Future work

Table of contents

Model problem introduction

We are interested in a model problem for groundwater flow modelling in porous media.
Let D be a physical domain (of dimension d), f a deterministic data function and a a Matérn Gaussian random field defined on $\Omega \times D$ where $(\Omega, \mathcal{A}, P)$ is some probability space.

Darcy problem [Eigel et al., 2016]

Almost everywhere on Ω,

$$\begin{aligned} -\operatorname{div}(\exp(a)\nabla u) &= f \quad \text{in } D, \\ u &= 0 \quad \text{on } \partial D. \end{aligned} \qquad \text{(Darcy)}$$

Model problem introduction

Recalls on Gaussian random fields

Gaussian random field

Let $(E, \mathcal{B}, m)$ be a measure space. A real valued Gaussian random field G on E is a function

$$\begin{array}{rccc} G: & \Omega \times E & \longrightarrow & \mathbb{R} \\ & (\omega, e) & \longmapsto & G_\omega(e), \end{array}$$

such that for any finite set $\{e_1, \cdots, e_n\} \subset E$, the vector $(G(e_1), \cdots, G(e_n))$, is a Gaussian random vector.
A Gaussian random field is charaterized by μ and Σ resp. its mean and autocovariance functions

$$\begin{array}{rccc} \mu: & E & \longrightarrow & \mathbb{R} \\ & e & \longmapsto & \mathbb{E}\left[G(e)\right], \end{array}$$

$$\begin{array}{rccc} \Sigma: & E \times E & \longrightarrow & \mathbb{R} \\ & (e, e') & \longmapsto & \mathbb{E}\left[(G(e) - \mu(e))(G(e') - \mu(e'))\right]. \end{array}$$

Model problem introduction

Recalls on Gaussian random fields

Gaussian white noise

We call Gaussian white noise on $\mathbb{R}^d$ the gaussian random field

$$\dot{\mathcal{W}} : \Omega \times L^2(\mathbb{R}^d) \longrightarrow \mathbb{R},$$

with zero mean and autocovariance function defined by

$$\begin{array}{rccc} \Sigma_{\dot{\mathcal{W}}} : & L^2(\mathbb{R}^d) \times L^2(\mathbb{R}^d) & \longrightarrow & \mathbb{R} \\ & (v, w) & \longmapsto & \displaystyle\int_{\mathbb{R}^d} vw \, \mathrm{d}x. \end{array}$$

Model problem introduction

Recalls on Gaussian random fields

Matérn random fields

Let us denote Γ the Euler gamma function and $\mathcal{K}_\nu$ the Bessel's modified function of the second kind of parameter ν. A Matérn random field on D is a particular Gaussian random field (on D) with autocovariance function $\mathcal{C}$ defined for x, y in D by

$$\mathcal{C}(x,y) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\kappa r)^\nu \mathcal{K}_\nu(\kappa r),$$

where,

$$r := |x-y|_2, \quad \kappa := \frac{\sqrt{8\nu}}{\lambda},$$

and the non-negative real parameters σ^2, ν and λ denote resp. the marginal variance, smoothness and correlation length of the field.

Model problem introduction

Weak form and quantity of interest

Darcy problem [Eigel et al., 2016]

Almost everywhere on Ω,

$$\begin{aligned} -\operatorname{div}(\exp(a)\nabla u) &= f \quad \text{in } D, \\ u &= 0 \quad \text{on } \partial D. \end{aligned} \tag{Darcy}$$

Model problem introduction

Weak form and quantity of interest

Darcy problem [Eigel et al., 2016]

Almost everywhere on Ω,

$$\begin{aligned} -\operatorname{div}(\exp(a)\nabla u) &= f \quad \text{in } D, \\ u &= 0 \quad \text{on } \partial D. \end{aligned} \tag{Darcy}$$

Weak form

Seek u in $L^2(\Omega) \times H_0^1(D)$ such that a.e. in Ω and for every v in $H_0^1(D)$

$$\int_D \exp(a)\nabla u \cdot \nabla v \, \mathrm{d}x = \int_D f v \, \mathrm{d}x. \tag{SPDE}$$

Model problem introduction

Weak form and quantity of interest

We are not interested in the entire solution u but only in the expectation of some linear quantity of interest defined from a deterministic function g by

Quantity of interest

$$\mathbb{E}[Q] := \mathbb{E}[Q(u)] := \int_\Omega \int_D gu \,\mathrm{d}x \,\mathrm{d}P(\omega). \qquad \text{(QoI)}$$

Model problem introduction

Weak form and quantity of interest

We are not interested in the entire solution u but only in the expectation of some linear quantity of interest defined from a deterministic function g by

Quantity of interest

$$\mathbb{E}[Q] := \mathbb{E}[Q(u)] := \int_\Omega \int_D gu \,\mathrm{d}x \,\mathrm{d}P(\omega). \quad \text{(QoI)}$$

Goals

- Estimate $\mathbb{E}[Q]$.
- Control the estimation error.

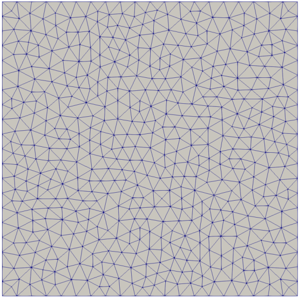
Table of contents

First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

To discretise our problem we need:

- A mesh (triangulation) $\mathcal{T}_h$ composed by cells denoted T,

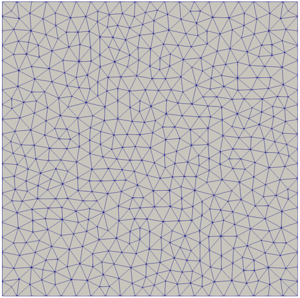


First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

To discretise our problem we need:

- A mesh (triangulation) $\mathcal{T}_h$ composed by cells denoted T,
- sets $\mathcal{P}^k(T)$ of polynomial functions of degree k on T,

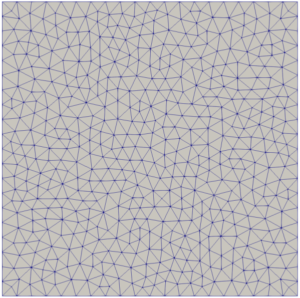


First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

To discretise our problem we need:

- A mesh (triangulation) $\mathcal{T}_h$ composed by cells denoted T,
- sets $\mathcal{P}^k(T)$ of polynomial functions of degree k on T,
- a finite dimensional space $V_h \subset H_0^1$,



$$V_h := \left\{ v_h \in \mathcal{C}^0(D),\ v_h \in \mathcal{P}^k(T)\ \forall T \in \mathcal{T}_h,\ v_{h|\partial D} = 0 \right\}.$$

First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

Weak form

Seek u in $L^2(\Omega) \times H_0^1(D)$ such that a.e. in Ω and for every v in $H_0^1(D)$

$$\int_D \exp(a) \nabla u \cdot \nabla v \; \mathrm{d}x = \int_D f v \; \mathrm{d}x. \qquad \text{(SPDE)}$$

First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

Weak form

Seek u in $L^2(\Omega) \times H_0^1(D)$ such that a.e. in Ω and for every v in $H_0^1(D)$

$$\int_D \exp(a) \nabla u \cdot \nabla v \, \mathrm{d}x = \int_D f v \, \mathrm{d}x. \qquad \text{(SPDE)}$$

Finite element problem

Seek u_h in $L^2(\Omega) \times V_h$ such that a.e. in Ω and for any v_h in V_h,

$$\int_D \exp(a) \nabla u_h \cdot \nabla v_h \, \mathrm{d}x = \int_D f v_h \, \mathrm{d}x. \qquad \text{(FE)}$$

First approach: standard Monte-Carlo

Deterministic discretisation: Finite element method

Weak form

Seek u in $L^2(\Omega) \times H_0^1(D)$ such that a.e. in Ω and for every v in $H_0^1(D)$

$$\int_D \exp(a) \nabla u \cdot \nabla v \, \mathrm{d}x = \int_D f v \, \mathrm{d}x. \tag{SPDE}$$

Finite element problem

Seek u_h in $L^2(\Omega) \times V_h$ such that a.e. in Ω and for any v_h in V_h,

$$\int_D \exp(a) \nabla u_h \cdot \nabla v_h \, \mathrm{d}x = \int_D f v_h \, \mathrm{d}x. \tag{FE}$$

First approach: standard Monte-Carlo

Discretisation of the random field

We need to draw a sample from a discretization of the random field a.

- Cholesky decomposition,
  - simple to derive,
  - dense covariance matrix decomposition,

First approach: standard Monte-Carlo

Discretisation of the random field

We need to draw a sample from a discretization of the random field a.

- Cholesky decomposition,
  - simple to derive,
  - dense covariance matrix decomposition,
- Karhunen-Loève decomposition [Matthies, 2008],
  - dense eigenvalue problem to solve or dense covariance matrix decomposition,
  - can be expensive if the random field is not smooth.

First approach: standard Monte-Carlo

Discretisation of the random field

We need to draw a sample from a discretization of the random field a.

- Cholesky decomposition,
  - simple to derive,
  - dense covariance matrix decomposition,
- Karhunen-Loève decomposition [Matthies, 2008],
  - dense eigenvalue problem to solve or dense covariance matrix decomposition,
  - can be expensive if the random field is not smooth.
- SPDE numerical resolution (with FEM) [Whittle, 1954], [Lindgren et al., 2011], [Bolin et al., 2017]
  - reduced computational complexity due to sparse precision matrix,
  - problem similar to the main one,
  - allows to define generalisations of the Matérn field that are still useful in practice,
  - a «straightforward» generalisation to manifolds using Laplace-Beltrami operator.

First approach: standard Monte-Carlo

Discretisation of the random field

Matérn SPDE [Croci et al., 2018]

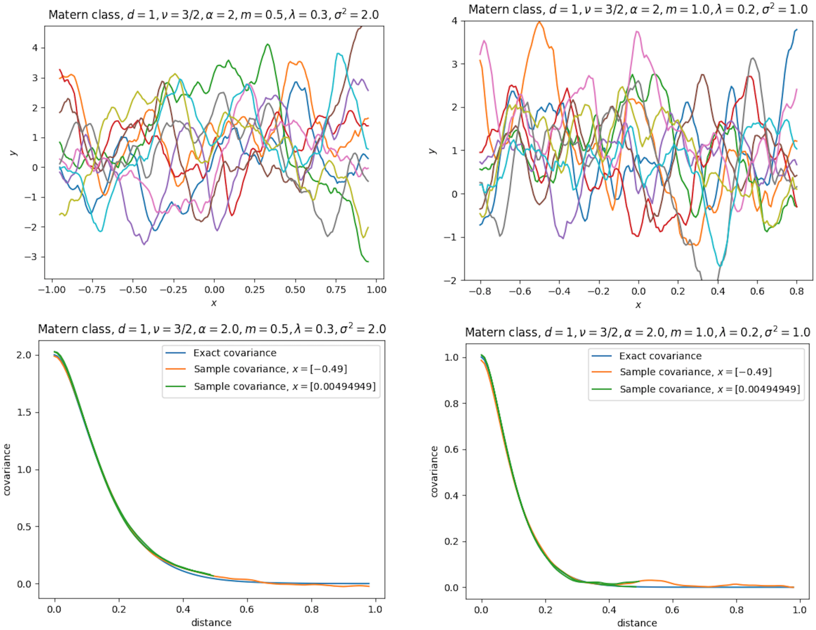
Given a Gaussian white noise $\dot{\mathcal{W}}$ defined on $\mathbb{R}^d$ and real parameters $\kappa > 0$ and $\alpha > d/2$, the solution a of the SPDE

$$(\kappa^2 - \Delta)^{\alpha/2} a = \dot{\mathcal{W}},$$

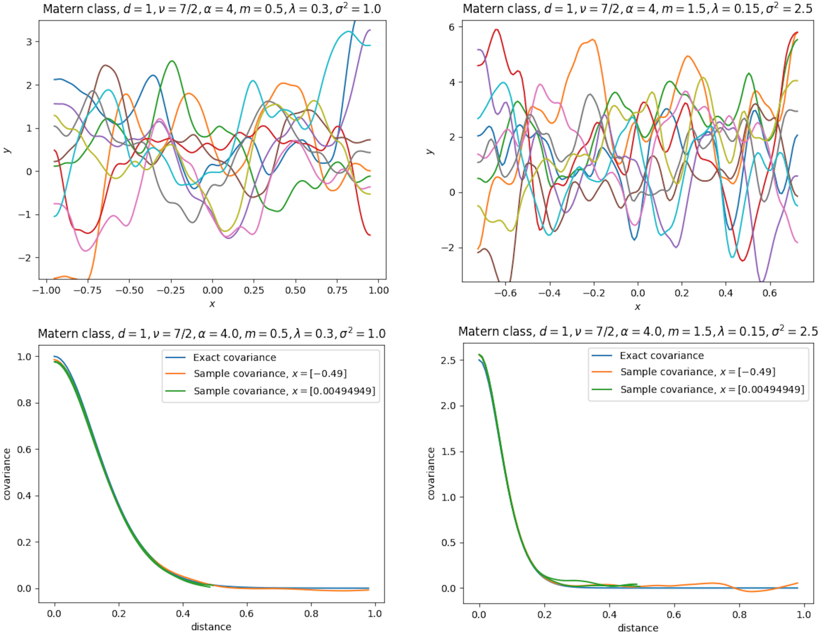
is a Matérn random field defined on $\mathbb{R}^d$ with:

- smoothness $\nu = \alpha - d/2$,
- marginal variance $\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\nu+d/2)(4\pi)^{d/2}\kappa^{2\nu}}$,
- correlation length $\lambda \simeq \frac{\sqrt{8\nu}}{\kappa}$.

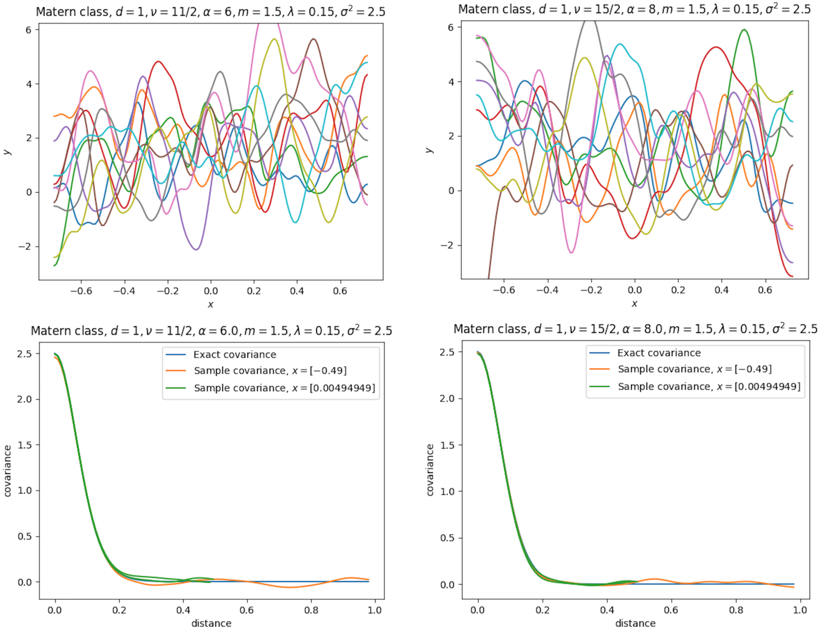
$$(\kappa^2 - \Delta)^{\alpha/2} a = \dot{\mathcal{W}}.$$



$$(\kappa^2 - \Delta)^{\alpha/2} a = \dot{\mathcal{W}}.$$



$$(\kappa^2 - \Delta)^{\alpha/2} a = \dot{\mathcal{W}}.$$



First approach: standard Monte-Carlo

Discretisation of the random field

Once we have solved the Matérn SPDE as well as the (FE) problem, we get a sample of the numerical solution u_h and we can compute an approximation of (QoI)

$$Q_h := Q(u_h) = \int_D g u_h \, \mathrm{d}x.$$

First approach: standard Monte-Carlo

Discretisation of the random field

Once we have solved the Matérn SPDE as well as the (FE) problem, we get a sample of the numerical solution u_h and we can compute an approximation of (QoI)

$$Q_h := Q(u_h) = \int_D g u_h \, \mathrm{d}x.$$

Then,

$$\mathbb{E}\left[Q\right] \simeq \mathbb{E}\left[Q_h\right].$$

First approach: standard Monte-Carlo

Stochastic discretisation: Monte Carlo method

Monte Carlo

Let $\left(Q_h^{(n)}\right)_{n=1}^{N}$ be independent random variables in $L^1(\Omega, \mathbb{R})$ of same law than Q_h, then

$$\mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right] := N^{-1} \sum_{n=1}^{N} Q_h^{(n)} \xrightarrow{a.s.} \mathbb{E}\left[Q_h\right].$$

First approach: standard Monte-Carlo

Stochastic discretisation: Monte Carlo method

Monte Carlo

Let $\left(Q_h^{(n)}\right)_{n=1}^{N}$ be independent random variables in $L^1(\Omega, \mathbb{R})$ of same law than Q_h, then

$$\mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right] := N^{-1} \sum_{n=1}^{N} Q_h^{(n)} \xrightarrow{a.s.} \mathbb{E}\left[Q_h\right].$$

For N large enough we have,

$$\mathbb{E}\left[Q\right] \simeq \mathbb{E}\left[Q_h\right] \simeq \mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right].$$

First approach: error control

$$\mathbb{E}\left[Q\right] \approx \mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right].$$

First approach: error control

$$\mathbb{E}\left[Q\right] \approx \mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right].$$

- **Mean square error:** for a given tolerance ε,

$$\mathbb{E}\left[\left(\mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right] - \mathbb{E}\left[Q\right]\right)^2\right] \quad = \quad \mathrm{Var}\left[\mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right]\right] \quad + \quad \mathbb{E}\left[\mathrm{E}_N^{\mathrm{MC}}\left[Q_h\right] - Q\right]^2$$

First approach: error control

$$\mathbb{E}[Q] \approx \mathrm{E}_N^{\mathrm{MC}}[Q_h].$$

- **Mean square error:** for a given tolerance ε,

$$\begin{aligned}
\mathbb{E}\left[\left(\mathrm{E}_N^{\mathrm{MC}}[Q_h] - \mathbb{E}[Q]\right)^2\right] &= \mathrm{Var}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h]\right] + \mathbb{E}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h] - Q\right]^2 \\
&= N^{-1}\,\mathrm{Var}[Q] + \mathbb{E}[Q_h - Q]^2
\end{aligned}$$

First approach: error control

$$\mathbb{E}[Q] \approx \mathrm{E}_N^{\mathrm{MC}}[Q_h].$$

- **Mean square error:** for a given tolerance ε,

$$\begin{aligned}
\mathbb{E}\left[\left(\mathrm{E}_N^{\mathrm{MC}}[Q_h] - \mathbb{E}[Q]\right)^2\right] &= \mathrm{Var}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h]\right] &+& \quad \mathbb{E}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h] - Q\right]^2 \\
&= N^{-1}\,\mathrm{Var}[Q] &+& \quad \mathbb{E}[Q_h - Q]^2 \\
&= \text{Variance} &+& \quad \text{FE bias} \\
&\leqslant \varepsilon^2.
\end{aligned}$$

First approach: error control

$$\mathbb{E}[Q] \approx \mathrm{E}_N^{\mathrm{MC}}[Q_h].$$

- **Mean square error:** for a given tolerance ε,

$$\begin{aligned}
\mathbb{E}\left[\left(\mathrm{E}_N^{\mathrm{MC}}[Q_h] - \mathbb{E}[Q]\right)^2\right] &= \mathrm{Var}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h]\right] + \mathbb{E}\left[\mathrm{E}_N^{\mathrm{MC}}[Q_h] - Q\right]^2 \\
&= N^{-1}\,\mathrm{Var}[Q] + \mathbb{E}[Q_h - Q]^2 \\
&= \text{Variance} + \text{FE bias} \\
&\leqslant \varepsilon^2.
\end{aligned}$$

- **Computational cost:** if we assume that $|\mathbb{E}[Q_h - Q]| \leqslant ch^{\alpha}$,

$$\mathrm{Cost}(\mathrm{E}_N^{\mathrm{MC}}[Q_h]) = \mathcal{O}\left(Nh^{-1}\right) = \mathcal{O}(\varepsilon^{-2-\alpha}).$$

Table of contents

Second approach: multi-level Monte-Carlo

Deterministic discretisation: Finite element method

Finite element problem

Let $\mathcal{T}_l$ be a triangulation on D of maximum diameter h_l and $V_l \subset H_0^1$ be a finite dimensional function space. Seek u_l in $L^2(\Omega) \times V_l$ such that almost everywhere in Ω and for any v_l in V_l,

$$\int_D \exp(a) \nabla u_l \cdot \nabla v_l \, \mathrm{d}x = \int_D f v_l \, \mathrm{d}x. \tag{FE}$$

Second approach: multi-level Monte-Carlo

Deterministic discretisation: Finite element method

Finite element problem

Let $\mathcal{T}_l$ be a triangulation on D of maximum diameter h_l and $V_l \subset H_0^1$ be a finite dimensional function space. Seek u_l in $L^2(\Omega) \times V_l$ such that almost everywhere in Ω and for any v_l in V_l,

$$\int_D \exp(a) \nabla u_l \cdot \nabla v_l \, \mathrm{d}x = \int_D f v_l \, \mathrm{d}x. \tag{FE}$$

Quantity of interest

The finite element approximation of (QoI) is given by,

$$\mathbb{E}\left[Q_l\right] := \mathbb{E}\left[Q(u_l)\right] := \int_\Omega \int_D g u_l \, \mathrm{d}x \, \mathrm{d}P(\omega).$$

Second approach: multi-level Monte-Carlo

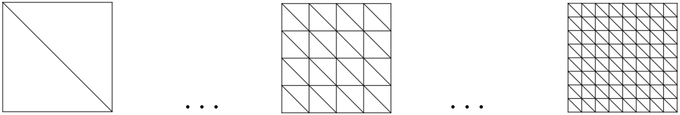
Stochastic discretisation: Multi-level Monte Carlo method

Multi-level Monte Carlo method is a multi-fidelity method and variance reduction method ([Peherstorfer et al., 2018], [Giles, 2015]).

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

Multi-level Monte Carlo method is a multi-fidelity method and variance reduction method ([Peherstorfer et al., 2018], [Giles, 2015]).



u_0 $\quad\dots\quad$ u_l $\quad\dots\quad$ u_L

$$Q(u_0) =: Q_0 \qquad Q(u_l) =: Q_l \qquad Q(u_L) =: Q_L$$

Less precise	$\longrightarrow$	More precise
Less expensive	$\longrightarrow$	More expensive

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

$$\mathbb{E}[Q] \quad \approx \quad \mathbb{E}[Q_L]$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

$$\begin{aligned}\mathbb{E}[Q] &\approx \mathbb{E}[Q_L] \\ &= \mathbb{E}[Q_0] + \sum_{l=1}^{L} \mathbb{E}[Q_l - Q_{l-1}]\end{aligned}$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

$$
\begin{aligned}
\mathbb{E}[Q] &\approx \mathbb{E}[Q_L] \\
&= \mathbb{E}[Q_0] + \sum_{l=1}^{L} \mathbb{E}[Q_l - Q_{l-1}]
\end{aligned}
$$

$$\wr \qquad \wr$$

$$
\mathrm{E}_L^{\mathrm{ML}}[Q_L] := N_0^{-1} \sum_{n=1}^{N_0} Q_0^{(n)} + \sum_{l=1}^{L} N_l^{-1} \sum_{n=1}^{N_l} \left(Q_l^{(n)} - Q_{l-1}^{(n)} \right)
$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

$$
\begin{aligned}
\mathbb{E}[Q] &\approx \mathbb{E}[Q_L] \\
&= \mathbb{E}[Q_0] + \sum_{l=1}^{L} \mathbb{E}[Q_l - Q_{l-1}] \\
& \qquad \wr \qquad\qquad\qquad\qquad \wr \\
\mathrm{E}_L^{\mathrm{ML}}[Q_L] &:= N_0^{-1} \sum_{n=1}^{N_0} Q_0^{(n)} + \sum_{l=1}^{L} N_l^{-1} \sum_{n=1}^{N_l} \left(Q_l^{(n)} - Q_{l-1}^{(n)} \right) \\
&= \mathrm{E}_{N_0}^{\mathrm{MC}}[Q_0] + \sum_{l=1}^{L} \mathrm{E}_{N_l}^{\mathrm{MC}}[Q_l - Q_{l-1}] .
\end{aligned}
$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

Let us rewrite the MLMC estimator by defining

$$Y_l := \begin{cases} Q_0, & l = 0, \\ Q_l - Q_{l-1}, & l > 0. \end{cases}$$

Then,

$$\mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right] := \sum_{l=0}^{L} \mathrm{E}_{N_l}^{\mathrm{MC}}\left[Y_l\right].$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

Lvl	Mesh	Precision & Comp. cost	[Samples]	$\xrightarrow{N^{-1}\sum}$	MC estimator
0		Low	$\left[Y_0^{(1)},\ Y_0^{(2)},\ \cdots,\ Y_0^{(N_0-1)},\ Y_0^{N_0}\right]$	$\xrightarrow{{N_0}^{-1}\sum}$	$\mathrm{E}_{N_0}^{\mathrm{MC}}\left[Y_0\right]$
$\vdots$					
l		Mid	$\left[Y_l^{(1)},\cdots,Y_l^{(N_l-1)},Y_l^{(N_l)}\right]$	$\xrightarrow{{N_l}^{-1}\sum}$	$\mathrm{E}_{N_l}^{\mathrm{MC}}\left[Y_l\right]$
$\vdots$					
L		High	$\left[Y_L^{(1)},\cdots,Y_L^{(N_L)}\right]$	$\xrightarrow{{N_L}^{-1}\sum}$	$\mathrm{E}_{N_L}^{\mathrm{MC}}\left[Y_L\right]$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

Lvl	Mesh	Precision & Comp. cost	[Samples]	$\xrightarrow{N^{-1}\sum}$	MC estimator
0		Low	$\left[Y_0^{(1)},\ Y_0^{(2)},\ \cdots,\ Y_0^{(N_0-1)},\ Y_0^{N_0}\right]$	$\xrightarrow{{N_0}^{-1}\sum}$	$\mathrm{E}_{N_0}^{\mathrm{MC}}[Y_0]$
$\vdots$					
l		Mid	$\left[Y_l^{(1)},\cdots,Y_l^{(N_l-1)},Y_l^{(N_l)}\right]$	$\xrightarrow{{N_l}^{-1}\sum}$	$\mathrm{E}_{N_l}^{\mathrm{MC}}[Y_l]$
$\vdots$					
L		High	$\left[Y_L^{(1)},\cdots,Y_L^{(N_L)}\right]$	$\xrightarrow{{N_L}^{-1}\sum}$	$\mathrm{E}_{N_L}^{\mathrm{MC}}[Y_L]$

$$\mathrm{E}_L^{\mathrm{ML}}[Q_L]$$

Second approach: multi-level Monte-Carlo

Stochastic discretisation: Multi-level Monte Carlo method

Lvl	Mesh	Precision & Comp. cost	[Samples]	$\xrightarrow{N^{-1}\sum}$	MC estimator
0		Low	$\left[Y_0^{(1)},\ Y_0^{(2)},\ \cdots,\ Y_0^{(N_0-1)},\ Y_0^{N_0}\right]$	$\xrightarrow{{N_0}^{-1}\sum}$	$\mathrm{E}^{\mathrm{MC}}_{N_0}[Y_0]$
$\vdots$					
l		Mid	$\left[Y_l^{(1)},\cdots,Y_l^{(N_l-1)},Y_l^{(N_l)}\right]$	$\xrightarrow{{N_l}^{-1}\sum}$	$\mathrm{E}^{\mathrm{MC}}_{N_l}[Y_l]$
$\vdots$					
L		High	$\left[Y_L^{(1)},\cdots,Y_L^{(N_L)}\right]$	$\xrightarrow{{N_L}^{-1}\sum}$	$\mathrm{E}^{\mathrm{MC}}_{N_L}[Y_L]$

How to choose these parameters ?

$$\mathrm{E}^{\mathrm{ML}}_{L}[Q_L]$$

Second approach: error control

Second approach: error control

Mean square error

$$\mathbb{E}\left[Q\right] \approx \mathbb{E}\left[Q_L\right] \approx \mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right].$$

Second approach: error control

Mean square error

$$\mathbb{E}\left[Q\right] \approx \mathbb{E}\left[Q_L\right] \approx \mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right].$$

- **Mean square error:** given a tolerance ε,

$$\begin{aligned}
\mathbb{E}\left[\left(\mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right]-\mathbb{E}\left[Q\right]\right)^2\right] &= \mathrm{Var}\left[\mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right]\right] + \mathbb{E}\left[\mathrm{E}_L^{\mathrm{ML}}\left[Q_L\right]-Q\right]^2 \\
&= \sum_{l=0}^{L} N_l^{-1}\,\mathrm{Var}\left[Y_l\right] + \mathbb{E}\left[Q_L - Q\right]^2 \\
&= \text{Variance} + \text{FE bias} \\
&\leqslant \varepsilon^2.
\end{aligned}$$

Theorem [Giles, 2008], [Giles, 2015]

If there exist independent estimators Y_l based on N_l Monte Carlo samples, and positives constants $\alpha,\ \beta,\ \gamma,\ c_1,\ c_2,\ c_3$ such that $\alpha \geqslant \frac{1}{2}\min(\beta,\gamma)$ and

1/ $|\mathbb{E}[Q_l - Q]| \leqslant c_1 h_l^\alpha$,

2/ $V_l := \mathrm{Var}[Y_l] \leqslant c_2 N_l^{-1} h_l^\beta$,

3/ C_l, the computational complexity of Y_l is bounded by $C_l \leqslant c_3 N_l h_l^{-\gamma}$,

then for any tolerance $\varepsilon < \mathrm{e}^{-1}$ there exist an integer L and a sequence of integers $(N_l)_{l=0}^L$ for which we achieve the mean square error bound

$$\mathbb{E}\left[\left(\mathrm{E}_L^{\mathrm{ML}}[Q_L] - \mathbb{E}[Q]\right)^2\right] < \varepsilon^2.$$

Moreover there exists a constant $c_4 > 0$ such that the overall computational complexity C of the MLMC estimator is bounded by

$$C \leqslant \begin{cases} c_4\varepsilon^{-2}, & \beta > \gamma, \\ c_4\varepsilon^{-2}\ln(\varepsilon)^2, & \beta = \gamma, \\ c_4\varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Theorem [Giles, 2008], [Giles, 2015]

If there exist independent estimators Y_l based on N_l Monte Carlo samples, and positives constants $\alpha,\ \beta,\ \gamma,\ c_1,\ c_2,\ c_3$ such that $\alpha \geqslant \frac{1}{2}\min(\beta,\gamma)$ and

1/ $|\mathbb{E}[Q_l - Q]| \leqslant c_1 h_l^{\alpha}$,

2/ $V_l := \mathrm{Var}[Y_l] \leqslant c_2 N_l^{-1} h_l^{\beta}$,

3/ C_l, the computational complexity of Y_l is bounded by $C_l \leqslant c_3 N_l h_l^{-\gamma}$,

then for any tolerance $\varepsilon < \mathrm{e}^{-1}$ there exist an integer L and a sequence of integers $(N_l)_{l=0}^{L}$ for which we achieve the mean square error bound

$$\mathbb{E}\left[\left(\mathrm{E}_L^{\mathrm{ML}}[Q_L] - \mathbb{E}[Q]\right)^2\right] < \varepsilon^2.$$

Moreover there exists a constant $c_4 > 0$ such that the overall computational complexity C of the MLMC estimator is bounded by

$$C \leqslant \begin{cases} c_4\varepsilon^{-2}, & \beta > \gamma, \\ c_4\varepsilon^{-2}\ln(\varepsilon)^2, & \beta = \gamma, \\ c_4\varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Theorem [Giles, 2008], [Giles, 2015]

If there exist independent estimators Y_l based on N_l Monte Carlo samples, and positives constants α, β, γ, c_1, c_2, c_3 such that $\alpha \geqslant \frac{1}{2}\min(\beta,\gamma)$ and

1/ $|\mathbb{E}[Q_l - Q]| \leqslant c_1 h_l^{\alpha}$,

2/ $V_l := \mathrm{Var}[Y_l] \leqslant c_2 N_l^{-1} h_l^{\beta}$,

3/ C_l, the computational complexity of Y_l is bounded by $C_l \leqslant c_3 N_l h_l^{-\gamma}$,

then for any tolerance $\varepsilon < \mathrm{e}^{-1}$ there exist an integer L and a sequence of integers $(N_l)_{l=0}^{L}$ for which we achieve the mean square error bound

$$\mathbb{E}\left[\left(\mathrm{E}_L^{\mathrm{ML}}[Q_L] - \mathbb{E}[Q]\right)^2\right] < \varepsilon^2.$$

Moreover there exists a constant $c_4 > 0$ such that the overall computational complexity C of the MLMC estimator is bounded by

$$C \leqslant \begin{cases} c_4\varepsilon^{-2}, & \beta > \gamma, \\ c_4\varepsilon^{-2}\ln(\varepsilon)^2, & \beta = \gamma, \\ c_4\varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Second approach: error control

Stochastic error control

$$C \leqslant \begin{cases} c_4 \varepsilon^{-2}, & \beta > \gamma, \\ c_4 \varepsilon^{-2} \ln(\varepsilon)^2, & \beta = \gamma, \\ c_4 \varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

Using the expressions of $(N_l)_{l=0}^{L}$ computed in Giles' theorem, we can write

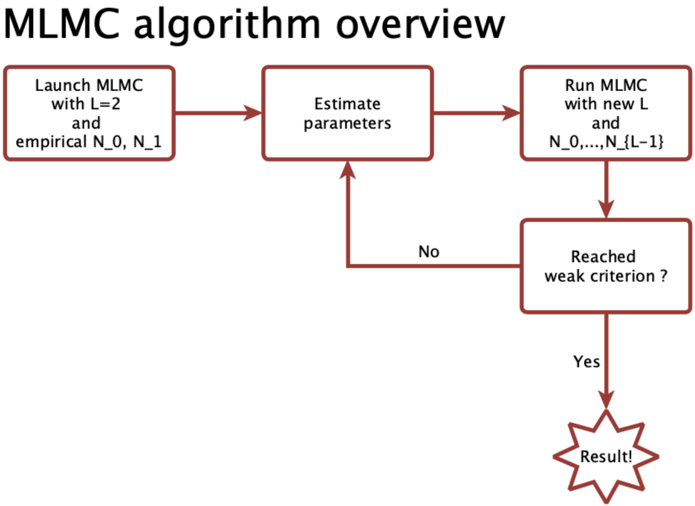
$$C \leqslant 2\varepsilon^{-2} \left(\sum_{l=0}^{L} \sqrt{V_l C_l} \right)^2$$

$$\leqslant 2\varepsilon^{-2} \left(\sum_{l=0}^{L} h_l^{\frac{\beta-\gamma}{2}} \right)^2$$

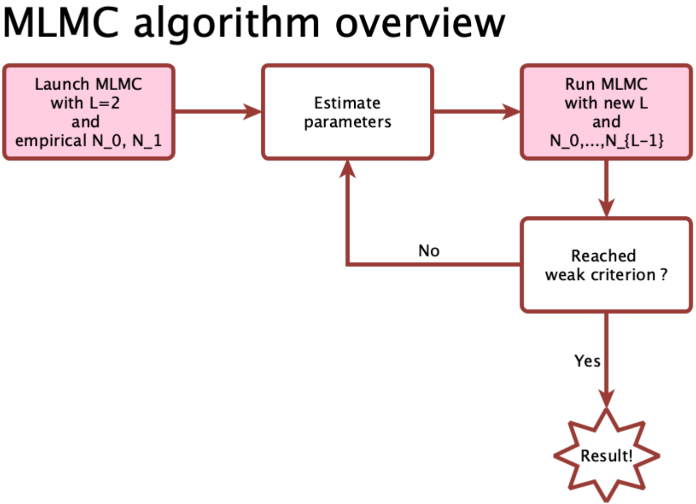
MLMC Algorithm

- Model problem introduction
- First approach: standard Monte-Carlo
- First approach: error control
- Second approach: multi-level Monte-Carlo
- Second approach: error control
- MLMC Algorithm
- Future work

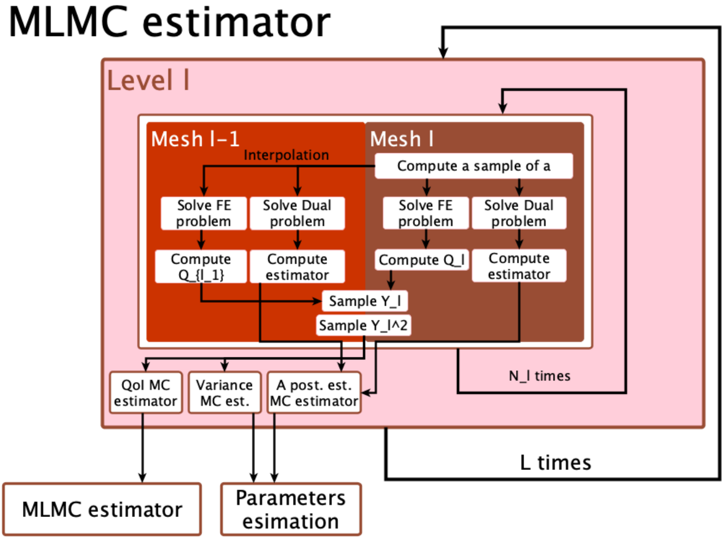
MLMC Algorithm



MLMC Algorithm



MLMC Algorithm



MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

6/ Check if a) $\hat{V}_1 \leqslant \frac{\varepsilon^2}{2}$ and if b) $\eta_l \leqslant \frac{\varepsilon^2}{2}$.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

6/ Check if a) $\hat{V}_1 \leqslant \frac{\varepsilon^2}{2}$ and if b) $\eta_l \leqslant \frac{\varepsilon^2}{2}$.
   - If a) and b) are satisfied, stop the algorithm.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

6/ Check if a) $\hat{V}_1 \leqslant \frac{\varepsilon^2}{2}$ and if b) $\eta_l \leqslant \frac{\varepsilon^2}{2}$.

- If a) and b) are satisfied, stop the algorithm.
- If a) is satisfied and b) is not, (re)compute the optimal number of samples to add to each level and go to 2/.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

6/ Check if a) $\hat{V}_1 \leqslant \frac{\varepsilon^2}{2}$ and if b) $\eta_l \leqslant \frac{\varepsilon^2}{2}$.

- If a) and b) are satisfied, stop the algorithm.
- If a) is satisfied and b) is not, (re)compute the optimal number of samples to add to each level and go to 2/.
- If b) is satisfied and a) is not, add a new level, compute the optimal number of samples to add to each level and go to 2/.

MLMC Algorithm

My idea of the algorithm,

1/ Initiate the algorithm with $L = 2$ and an arbitrary (not too big) number of samples for each level, say $N_0 = N_1 = 10$.

2/ Compute N_0 samples of $Q_0 = Y_0$ as well as the associated MC estimator.

3/ Compute N_1 samples of $Y_1 = Q_1 - Q_0$ and Y_1^2.

4/ Compute the associated MC estimator, as well as the MC estimator of the variance of Y_1, denoted $\hat{V}_1$.

5/ Compute a local a posteriori error estimation η_l of $|\mathbb{E}[Q_1 - Q]|$ to refine the mesh.

6/ Check if a) $\hat{V}_1 \leqslant \frac{\varepsilon^2}{2}$ and if b) $\eta_l \leqslant \frac{\varepsilon^2}{2}$.

- If a) and b) are satisfied, stop the algorithm.
- If a) is satisfied and b) is not, (re)compute the optimal number of samples to add to each level and go to 2/.
- If b) is satisfied and a) is not, add a new level, compute the optimal number of samples to add to each level and go to 2/.
- If neither a) or b) are satisfied, idem.

Future work

- Model problem introduction
- First approach: standard Monte-Carlo
- First approach: error control
- Second approach: multi-level Monte-Carlo
- Second approach: error control
- MLMC Algorithm
- Future work

Future work

- Get some numerical results,

Future work

- Get some numerical results,
- try other ways to measure the error, e.g. confidence interval ([Collier et al., 2015]),

Future work

- Get some numerical results,
- try other ways to measure the error, e.g. confidence interval ([Collier et al., 2015]),
- do some numerical analysis on Giles' theorem/algorithm and include more a posteriori error estimation in it,

Future work

- Get some numerical results,
- try other ways to measure the error, e.g. confidence interval ([Collier et al., 2015]),
- do some numerical analysis on Giles' theorem/algorithm and include more a posteriori error estimation in it,
- run tests on engineering examples (i.e. 3D + HPC),

Future work

- Get some numerical results,
- try other ways to measure the error, e.g. confidence interval ([Collier et al., 2015]),
- do some numerical analysis on Giles' theorem/algorithm and include more a posteriori error estimation in it,
- run tests on engineering examples (i.e. 3D + HPC),
- study and run tests for the Matérn SPDE,

Future work

- Get some numerical results,
- try other ways to measure the error, e.g. confidence interval ([Collier et al., 2015]),
- do some numerical analysis on Giles' theorem/algorithm and include more a posteriori error estimation in it,
- run tests on engineering examples (i.e. 3D + HPC),
- study and run tests for the Matérn SPDE,
- try random field data with non-homogeneous correlation length and random region of smaller correlation length.

Acknowledgement

Thank you for your attention!

The ASSIST project has received funding from the University of Luxembourg Internal Research Project scheme.
The DRIVEN project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 811099.

References I

Bolin, D., Kirchner, K., and Kovács, M. (2017). Numerical solution of fractional elliptic stochastic PDEs with spatial white noise. *IMA J. Numer. Anal.*, pages 1–21.

Collier, N., Lateef, A., Ali, H., Nobile, F., Schwerin, E. V., Haji-Ali, A. L., Nobile, F., von Schwerin, E., and Tempone, R. (2015). A continuation multilevel Monte Carlo algorithm. *BIT Numer. Math.*, 55(2):399–432.

Croci, M., Giles, M. B., Rognes, M. E., and Farrell, P. E. (2018). Efficient white noise sampling and coupling for multilevel Monte Carlo with non-nested meshes. *SIAM/ASA J. Uncertainty Quantification*, pages 1–28.

Eigel, M., Merdon, C., and Neumann, J. (2016). An Adaptive Multilevel Monte Carlo Method with Stochastic Bounds for Quantities of Interest with Uncertain Data. *SIAM/ASA J. Uncertain. Quantif.*, 4(1):1219–1245.

Giles, M. B. (2008). Multilevel Monte Carlo Path Simulation. *Oper. Res.*, 56(3):607–617.

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numer.*, 24(2015):259–328.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *J. R. Stat. Soc. Ser. B*, 73:423–498.

Matthies, H. G. (2008). Stochastic finite elements: Computational approaches to stochastic partial differential equations. *ZAMM Zeitschrift fur Angew. Math. und Mech.*, 88(11):849–873.

References II

Peherstorfer, B., Willcox, K., and Gunzburger, M. (2018). Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Rev.*, 60(3):550–591.

Whittle, P. (1954). On Stationary Processes in the Plane. *Biometrika*, 41(3/4):434–449.