ExaQUte

Exascale Quantification of Uncertainties for Technology and Science Simulation

D5.5 Report on the application of multi-level Monte Carlo to wind engineering

Document information table

Contract number:	800898
Project acronym:	ExaQUte
Project Coordinator:	CIMNE
Document Responsible Partner:	EPFL
Deliverable Type:	Report
Dissemination Level:	PUblic
Related WP & Task:	WP5 task 5.5
Status:	Final version

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No **800898**

Authoring

Prepared by EPFL and CIMNE				
Authors	Partner	Modified	Version	Comments
Quentin Ayoul-Guilmard	EPFL			co-author, coordinator
Fabio Nobile				Expert and reviewer
Sundar Ganesh				co-author
Marc Núñez	CIMNE			co-author
Riccardo Tosi				co-author

Change log

Versions	Modified pages/sections	Comments
1.0	All	Submitted version

Approval

Approved by EPFL and CIMNE				
	Name	Partner	Date	OK
Task leader	Fabio Nobile	EPFL	2021-11-30	✔
Coordinator	Riccardo Rossi	CIMNE	2021-11-25	✔
Reviewer	Javier Principe	UPC	2021-11-30	✔

Executive summary

We study the use of multi-level Monte Carlo methods for wind engineering. This report brings together methodological research on uncertainty quantification and work on target applications of the ExaQUte project in wind and civil engineering.

First, a multi-level Monte Carlo for the estimation of the conditional value at risk and an adaptive algorithm are presented. Their reliability and performance are shown on the time-average of a non-linear oscillator and on the lift coefficient of an airfoil, with both preset and adaptively refined meshes. Then, we propose an adaptive multi-fidelity Monte Carlo algorithm for turbulent fluid flows where multi-level Monte Carlo methods were found to be inefficient. Its efficiency is studied and demonstrated on the benchmark problem of quantifying the uncertainty on the drag force of a tall building under random turbulent wind conditions.

All numerical experiments showcase the open-source software stack of the ExaQUte project for large-scale computing in a distributed environment.

Contents

1 Introduction

1.1 Context

This is a report on scientific research realised within the ExaQUte project ('exascale quantification of uncertainties for technology and science simulation'), part of the European Union's research programme Horizon 2020[1]. This project purposes to develop methods and tools to optimise the shape of civil-engineering structures so as to withstand uncertain wind conditions, with the use of large-scale distributed computing. It spans many scientific fields such as computational fluid dynamics (CFD), high-performance computing (HPC), uncertainty quantification (UQ) and optimisation under uncertainties (OUU). The research reported here pertains to and concludes the work package 'Algorithmic extensions of multi-level Monte Carlo', id est (i.e.) the investigation and advancement of UQ through a particular class of methods.

This document contains the latest research made on these methods and their application to target benchmarks taken from civil and wind engineering. It builds on two previous reports in the same work package.

deliverable 5.3 discussed the use of adaptive mesh refinement with multi-level Monte Carlo (MLMC) methods for fluid problems. This combination was found successful when the Reynolds number of the flow was low. However, flows with a higher Reynolds number presented a number of obstacles to an efficient use of MLMC methods; several possible lines of research were given.

deliverable 5.4 investigated the application and improvement of MLMC methods to the solutions of time-dependent stochastic problems. Good performance was reported on problems whose solutions exhibited certain properties of stability and mesh convergence. On the other hand, the chaotic behaviour of turbulent fluid flows was shown to be incompatible with the requirement of an efficient MLMC method. Several exploratory studies were made on possible solutions to address this issue; notably, a multi-fidelity Monte Carlo method was found promising.

This document is also related to two concurrent reports from other work packages: deliverable 3.3 presents research on CFD which is applied to numerical experiments in this report; deliverable 6.5 reports on OUU which relies on the UQ methods presented here.

All numerical simulations discussed in this report use a similar software stack whose code is publically available under open license, as are the scientific reports.

[1]Funding received under grant agreement 800898.

ParMMG+MMG is a software for simplicial remeshing developed by INRIA. A given mesh can be refined to achieve a given accuracy of interpolation, based on a provided metric (C. Dapogny et al. 2014). ParMMG is an extension developed during the ExaQUte project by Cirrottola and Froehly 2019 to perform this remeshing across several processors in parallel, via the message passing interface (MPI) standard. The latest work on this tool can be found in the concurrent report deliverable 2.5.

Kratos Multiphysics is the solver for the CFD simulation. It also supports MPI parallelisation. Its development is lead by CIMNE (Pooyan Dadvand et al. 2010; P. Dadvand et al. 2013), and its latest release is reported in deliverable 1.4.

COMPSs and Quake as the two alternative frameworks for distributed computing. COMPSs (Lordan et al. 2013; Rosa M. Badia et al. 2015; Tejedor et al. 2017) is developed by the Barcelona Supercomputing Center and Quake by IT4Innovations. They share a common application programming interface (API) developed for the ExaQUte project (see Böhm and Ejarque 2021). The latest developments are in the concurrent report deliverable 4.5

XMC is the library of hierarchical Monte Carlo (MC) methods for UQ. It was born for the ExaQUte project as deliverable 5.1; deliverable 5.2 and developed therefrom as a collaborative effort lead by the EPFL team.

This software stack is common to the whole ExaQUte project and actively developed as part of the project; it is described in further details in deliverable 1.4. Since this report focuses on uncertainty quantification, most of the work discussed hereafter is related to XMC.

1.2 Structure of the report

This report comprises two independent parts with their own methodology and numerical experiments. They are followed by a common concluding section, which also concludes the work package 5 'Algorithmic extensions of multi-level Monte Carlo' of the ExaQUte project.

Section 2 presents a MLMC method for the estimation of the conditional value at risk (CVaR). This risk measure features prominently in most reports of OUU (deliverable 6.2; deliverable 6.3; deliverable 6.4; deliverable 6.5) for its pertinence to robust engineering design. Nevertheless this is the first report of the ExaQUte project thoroughly discussing its accurate and efficient estimation, due to the various challenges it raises. The method will be present in section 2.1, concluding

with an algorithm to calibrate the estimation adaptively; section 2.2 will then illustrate its performance on two different experiments.

Section 3 presents a multi-fidelity Monte Carlo (MFMC) method for the case of turbulent fluid flows. This category of problems has been shown in previous reports to be unfavourable to the use of MLMC methods to accelerate UQ. MFMC was proposed as a viable alternative and is investigated in details in this report. The proposed method is explained in section 3.1, with a corresponding adaptive algorithm. Then, section 3.2 details its application to a reference benchmark problem of the ExaQUte project, based on civil engineering.

Finally, section 4 gives some conclusions and perspective on the research presented here and on the whole package.

1.3 Common framework and notation

This report focuses on the quantification of uncertainties of real-valued random variables, for which methods abound. Exempli gratia (e.g.) stochastic collocation methods such as from Babuška et al. 2007 rely on a quadrature-based approach while polynomial chaos expansions approximate the random variable using a polynomial expansion (see Ghanem and Spanos 1991; Le Maître and Knio 2010). In the following sections, this random variable will generically be referred to as the quantity of interest (QoI). It will typically be a function of the solution of a partial differential equation (PDE) with uncertain parameters of potentially high dimension (e.g. fluid flow problems with random inflow conditions). This entails high-dimensional output uncertainties, for which many methods – including those previously mentioned – become prohibitively expensive. Therefore, the ExaQUte project resorts prominently to Monte Carlo (MC) methods and variations thereof. Let us introduce here a few relevant notions used throughout the report.

Our probability space will be noted $(\Omega, \mathcal{F}, \mathbb{P})$. Let $Q \in \mathrm{L}^2(\Omega, \mathbb{R})$ be a real-valued random variable. The cost of generating a single sample of Q is noted $\in\!\!\!\in(Q)$; it is commonly measured in computational time. The MC estimator of $\mathbb{E}(Q)$ with sample size $m \in \mathbb{N}$

$$\mu_m(Q) := \frac{1}{m} \sum_{i=1}^{m} Q^{(i)}, \tag{1.1}$$

where the random variables[2] $\{Q^{(i)} : i \in [\![1, m]\!]\} \subset \mathrm{L}^2(\Omega, \mathbb{R})^m$ are independent and identically-distributed (i.i.d.) copies of Q. For a finite sample size, $\mu_m(Q)$ is random and approximating the expected value of Q with this estimator comes with

[2] $\forall (a, b) \in \mathbb{R}^2$, $[\![a, b]\!] := [a, b] \cap \mathbb{Z}$ denotes the set of integer numbers between a and b.

a statistical error defined as

$$\sqrt{\mathbb{E}((\mu_m(Q) - \mathbb{E}(Q))^2)} = \sqrt{\frac{\mathbb{V}\mathrm{ar}(Q)}{m}}.$$

Since the exact solution of the underlying PDE is not accessible in most applications of engineering, the QoI is typically computed from an approximate solution. One can often introduce different levels of approximations where higher levels are assumed to provide more accurate solutions at higher computational costs. The QoI computed at the level of accuracy $l \in \mathbb{N}$ will be denoted Q_l; the target level of accuracy required for the application at hand will generally be denoted $L \in \mathbb{N}$. Then approximating $\mathbb{E}(Q)$ by $\mu_m(Q_l)$ leads to a *biased* estimator whose bias error is defined as $|\mathbb{E}(Q_l - Q)|$. In the applications discussed in this report, the levels of approximation differ chiefly by the refinement of the spatial discretisation (or 'mesh') of the PDE, hence the alternative name 'discretisation error'. The mean squared error (MSE) of the MC estimator is the sum of the squares of the statistical and bias errors:

$$\mathrm{MSE}(\mu_m(Q_l)) := \mathbb{E}((\mu_m(Q_l) - \mathbb{E}(Q))^2) \quad = \mathbb{E}\Big((\mu_m(Q_l) - \mathbb{E}(Q_l))^2\Big) + (\mathbb{E}(Q_l - Q))^2.$$

Heinrich 2001 introduced an estimator combining these multiple models:

$$\mu_{\boldsymbol{m}}(\boldsymbol{Q}) := \mu_{m_0}(Q_0) + \sum_{l=1}^{L} \mu_{m_l}(Q_l - Q_{l-1})$$

with associated MSE

$$\mathrm{MSE}(\mu_{\boldsymbol{m}}(\boldsymbol{Q})) = \frac{\mathbb{V}\mathrm{ar}(Q_0)}{m_0} + \sum_{l=1}^{L} \frac{\mathbb{V}\mathrm{ar}(Q_l)}{m_l} + \mathbb{E}(Q_l - Q)^2$$

where $\boldsymbol{m} \in \mathbb{N}^{L+1}$ is the list of sample sizes and $\boldsymbol{Q} := (Q_l : l \in [\![0, L]\!])$ the list of numerical discretisations of the QoI. The fame of this multi-level Monte Carlo (MLMC) method owes much to the celebrated result by Michael B. Giles 2008, theorem 3.1: under assumption 1 below, the number of levels and sample sizes can always be chosen such that the MLMC estimator outperforms the MC estimator.

Assumption 1 (Levelwise asymptotic behaviour). Let $Q \in \mathrm{L}^2(\Omega, \mathbb{R})$ be a random variable. There exist strictly positive constants c_1, r_1, c_2, r_2, c_3, r_3 such that $\forall l \in \mathbb{N}$

$$\begin{aligned} |\mathbb{E}(Q_l - Q)| &\leqslant c_1 \mathrm{e}^{-l r_1}, \\ \mathbb{V}\mathrm{ar}(Q_l - Q_{l-1}) &\leqslant c_2 \mathrm{e}^{-l r_2}, \\ €(Q_l - Q_{l-1}) &\leqslant c_3 \mathrm{e}^{l r_3}. \end{aligned}$$

These inequality impose an upper bound on the levelwise asymptotic behaviour of the bias error, statistical error, and cost. ◊

Under assumption 1, with (n, k) chosen such that $\mathrm{MSE}(\mu_n(Q_k)) \leqslant \epsilon^2$, it can be shown that

$$€(\mu_n(Q_k)) \lesssim \epsilon^{-2-r_3/r_1} \tag{1.3}$$

while Michael B. Giles 2008, theorem 3.1 showed that one can choose L and $\boldsymbol{m} \in \mathbb{N}^{L+1}$ such that

$$€(\mu_{\boldsymbol{m}}(\boldsymbol{Q})) \lesssim \epsilon^{-2} \times \begin{cases} 1 & \text{if } r_2 > r_3 \\ (\ln \epsilon)^2 & \text{if } r_2 = r_3 \\ \epsilon^{-\frac{r_3 - r_2}{r_1}} & \text{if } r_2 < r_3 \end{cases} \tag{1.4}$$

with

$$€(\mu_{\boldsymbol{m}}(\boldsymbol{Q})) = m_0\, €(Q_0) + \sum_{l=1}^{L} m_l (€(Q_l) + €(Q_{l-1})).$$

Even in the unfavourable case where $r_2 < r_3$, (1.4) improves upon (1.3); in the favourable case, the cost is dominated by the sampling on the lower levels.

Since the following sections will consider a time-dependent PDE, we introduce below the finite temporal average.

Definition 1 (Temporal average). Let $T \in \mathbb{R}$, $d \in \mathbb{N}$, and $\boldsymbol{f} : [0, T] \to \mathbb{R}^d$ a function of time. Let $a \in [0, T[$ and $b \in]a, T[$. The temporal average (or 'time average') of the function $\boldsymbol{f}$ from a to b is defined as

$$\langle \boldsymbol{f} \rangle_{a,b} := \frac{1}{b-a} \int_a^b \boldsymbol{f}. \tag{1.5}$$

This notation may be simplified as $\langle \boldsymbol{f} \rangle_b := \langle \boldsymbol{f} \rangle_{0,b}$ and $\langle \boldsymbol{f} \rangle := \langle \boldsymbol{f} \rangle_{0,T}$. ◊

2 Multi-level Monte Carlo method for the conditional value-at-risk

The aim of the ExaQUte project is to carry out risk-averse design of civil engineering structures. To that end, we are required to quantify tail-risks and minimise them. It was shown in deliverable 6.2 that the CVaR was an ideal candidate to quantify these tail risks and possessed favourable properties for use in OUU problems and gradient-based optimisation methods. In deliverable 6.3 and deliverable 6.4, we also demonstrated several algorithms that combined gradient-based methods with the MLMC estimation of such risk-measures. However, the specific details of optimally calibrating the MLMC estimators for the CVaR have not been discussed in these works.

Several studies have been conducted into the estimation of risk-measures using MLMC estimators. A MLMC estimator for the cumulative distribution function (CDF) was proposed and analysed in Michael B Giles, Nagapetyan et al. 2015, 2017 wherein a smoothed approximation to the characteristic function was used. The MLMC method was also used in Michael B Giles and Haji-Ali 2019 for nested conditional expectations from which the value at risk (VaR) and CVaR could be derived. One can also derive the VaR and the CVaR from surrogate distributions derived from moments. For example, in the works Bierig and Chernov 2016; Gou 2016, a maximum entropy approach was used to estimate the probability density function (PDF) and the VaR using moment estimates from MLMC estimators.

The current work expands upon the work in Krumscheid and Nobile 2018. The work in ibid. built upon the ideas presented in Michael B Giles, Nagapetyan et al. 2015 and generalises them further to approximate general parametric expectations, which we introduce later in section 2.1.1. Novel MLMC estimators for the characteristic function were presented in Krumscheid and Nobile 2018 in addition to derived quantities such as the VaR and CVaR based on the idea of pointwise estimation combined with interpolation. An error bound was presented in ibid. to adaptively calibrate the parameters of the MLMC estimators for parametric expectations. This error bound was based on the use of inverse inequalities, which used large leading constants to bound the error. This results in conservative error estimates that led to MLMC hierarchies that are impractically expensive to compute. In Ayoul-Guilmard, Ganesh, Krumscheid et al. 2021 (in preparation), novel error estimators were presented based on the ideas in Krumscheid and Nobile 2018.

In this work, we present the ideas from Ayoul-Guilmard, Ganesh, Krumscheid et al. 2021. In section 2.1.3, we present novel error estimators for parametric expectations, as well as the VaR and the CVaR. We also present in section 2.1.4 a method to select the parameters of the MLMC hierarchy in a cost optimal manner based on these error estimates. Lastly, we present an algorithm in section 2.1.5 for

successfully improving the estimates required for calibrating the parameters of the MLMC hierarchy.

2.1 Method

2.1.1 Parametric expectations

We introduce the concept of parametric expectations, which are expectations of the form:

$$\Phi(\theta) := \mathbb{E}(\phi(\theta, Q)),$$

where Q denotes the QoI and $\theta \in \Theta$ denotes an input parameter to the expectation. In this work, we follow Krumscheid and Nobile 2018 and use the following particular form for the function ϕ.

$$\phi(\theta, Q) := \theta + \frac{1}{1-\tau}(Q-\theta)^+, \quad X^+ := \max(0, X), \quad \theta \in \Theta \subset \mathbb{R}, \tag{2.1}$$

where $\tau \in]0,1[$ denotes a significance parameter. This form has the advantage that after estimating the function Φ and its derivatives

$$\Phi^{(d)}(\theta) := \frac{\partial^d}{\partial \theta^d}\,\mathbb{E}(\phi(\theta, Q)), \quad d \in \{0,1,2\},$$

the CDF $F_Q(\theta) = \mathbb{E}\big(\mathbb{1}_{Q\leqslant\theta}\big)$ and the PDF $f_Q(\theta) = F_Q^{(1)}(\theta)$, as well as the VaR q_τ and the CVaR c_τ of significance τ, can be obtained by simple post-processing:

$$\begin{aligned} F_Q(\theta) &= \tau + (1-\tau)\Phi^{(1)}, & q_\tau &= \operatorname*{argmin}_{\theta\in\Theta} \Phi(\theta), \\ f_Q(\theta) &= (1-\tau)\Phi^{(2)}, & c_\tau &= \min_{\theta\in\Theta} \Phi(\theta) = \Phi(q_\tau). \end{aligned} \tag{2.2}$$

One can as well estimate the CDF for example by directly estimating the expectation in $F_Q(\theta) = \mathbb{E}\big(\mathbb{1}_{Q\leqslant\theta}\big)$ using MLMC. However, using MLMC to estimate the expected value of a discontinuous function can lead to suboptimal performance of the MLMC method, as shown in ibid. The parametric expectation approach overcomes this problem since ϕ is Lipschitz continuous in Q.

In the framework of MLMC, let us consider a sequence of approximations $Q_l, l \in [\![0, L]\!]$, of the target QoI of increasing accuracy and cost (see deliverable 5.4, and we denote $\phi_l(\theta) = \phi(\theta, Q_l)$. The form of the function ϕ given in (2.1) has some favourable properties. The first is that since ϕ is Lipschitz continuous in Q, the differences $|\phi_l(\theta) - \phi_{l-1}(\theta)|$ decay at the same rate as $|Q_l - Q_{l-1}|$. Similar assumptions can be made on ϕ as in assumption 1. This results in similar complexity results as in Michael B. Giles 2008 for the MLMC estimator of Φ as the MLMC estimator of $\mathbb{E}(Q)$. Secondly, this allows us to derive error bounds that relate the MSE on the function Φ to the MSE on pointwise estimates of ϕ, leading to computable error estimates.

2.1.2 MLMC estimator for parametric expectations

We focus on the problem of approximating parametric expectations using the MLMC method. The approach we follow is motivated by Michael B Giles, Nagapetyan et al. 2015; Krumscheid and Nobile 2018. We approximate the parametric expectation Φ and its derivatives $\Phi^{(d)}$ on an interval $\Theta \subset \mathbb{R}$ via the MLMC method as follows. We consider a set of $n \in \mathbb{N}$ nodes:

$$\boldsymbol{\theta} := \{\theta_j \in \Theta : j \in [\![1, n]\!];\ \theta_j < \theta_{j+1}\},$$

such that $\Theta = [\theta_1, \theta_n]$. The function Φ is then approximated at any point θ_j as:

$$\begin{aligned}\Phi(\theta_j) &\approx \mathbb{E}\big(\phi(\theta_j, Q_L)\big) \\ &= \mathbb{E}\big(\phi(\theta_j, Q_0)\big) + \sum_{l=1}^{L} \mathbb{E}\big(\phi(\theta_j, Q_l) - \phi(\theta_j, Q_{l-1})\big),\end{aligned}$$

and each of the expectations are estimated using independent Monte Carlo estimators; we then define the MLMC estimator $\hat{\Phi}_L(\theta_j)$ of $\Phi(\theta_j)$ as

$$\hat{\Phi}_L(\theta_j) := \frac{1}{m_0}\sum_{i=1}^{m_0} \phi(\theta_j, Q_0^{(i,0)}) + \sum_{l=1}^{L} \frac{1}{m_l}\sum_{i=1}^{m_l}\Big(\phi(\theta_j, Q_l^{(i,l)}) - \phi(\theta_j, Q_{l-1}^{(i,l)})\Big). \quad (2.3)$$

It is important to note that the same set of random events is used to evaluate the estimator for all θ_j. Finally, we obtain a MLMC estimator of the whole function $\Phi(\theta), \theta \in \Theta$ by interpolating over the pointwise estimates as below.

$$\hat{\Phi}_L = \mathcal{S}_n\Big(\hat{\Phi}_L(\boldsymbol{\theta})\Big),$$

where $\mathcal{S}_n$ denotes an appropriate interpolation operator and $\hat{\Phi}_L(\boldsymbol{\theta})$ denotes the set of pointwise MLMC estimates:

$$\hat{\Phi}_L(\boldsymbol{\theta}) = \{\hat{\Phi}_L(\theta_1), \hat{\Phi}_L(\theta_2), \dots, \hat{\Phi}_L(\theta_n)\}.$$

The function derivative estimate denoted by $\hat{\Phi}_L^{(d)}$ is then obtained by computing the derivative of the resultant interpolated function, provided it exists:

$$\hat{\Phi}_L^{(d)} := \mathcal{S}_n^{(d)}\Big(\hat{\Phi}_L(\boldsymbol{\theta})\Big),$$

where the superscript d denotes the order of the derivative. We use cubic spline interpolation over equally spaced points throughout this work and hence, $d < 3$. In addition, all interpolation operations in this work are computed using SciPy (Virtanen et al. 2020), wherein the boundary conditions on the first derivatives for cubic splines are inferred from the given pointwise data.

We use the following error criteria to quantify the accuracy of the function derivative estimate. Specifically, we consider the MSE

$$\mathrm{MSE}\big(\hat{\Phi}_L^{(d)}\big) := \mathbb{E}\Big(\big\|\Phi^{(d)} - \hat{\Phi}_L^{(d)}\big\|^2_{\mathrm{L}^\infty(\Theta)}\Big),$$

where the norm $\|f\|_{\mathrm{L}^\infty(\Theta)}$ of a function $f : \Theta \to \mathbb{R}$ is defined as

$$\|f\|_{\mathrm{L}^\infty(\Theta)} := \operatorname*{ess\,sup}_{\theta\in\Theta} |f(\theta)|.$$

By the triangle inequality, the MSE can be separated into three terms:

$$\begin{aligned}\mathrm{MSE}\big(\hat{\Phi}_L^{(d)}\big) \leqslant 3\Bigg(&\underbrace{\big\|\Phi^{(d)} - \mathcal{S}_n^{(d)}(\Phi(\boldsymbol{\theta}))\big\|^2_{\mathrm{L}^\infty(\Theta)}}_{\text{interpolation error}} + \underbrace{\big\|\mathcal{S}_n^{(d)}\big(\Phi(\boldsymbol{\theta}) - \mathbb{E}\big(\hat{\Phi}_L(\boldsymbol{\theta})\big)\big)\big\|^2_{\mathrm{L}^\infty(\Theta)}}_{\text{bias error}} \\ &+ \underbrace{\mathbb{E}\Big(\big\|\mathcal{S}_n^{(d)}\big(\hat{\Phi}_L(\boldsymbol{\theta}) - \mathbb{E}\big(\hat{\Phi}_L(\boldsymbol{\theta})\big)\big)\big\|^2_{\mathrm{L}^\infty(\Theta)}\Big)}_{\text{statistical error}}\Bigg) \\ =: 3\Big((\mathfrak{e}_i^{(d)})^2 + (\mathfrak{e}_b^{(d)})^2 + (\mathfrak{e}_s^{(d)})^2\Big),\end{aligned} \tag{2.4}$$

where we have used the notation $\mathfrak{e}_i^{(d)}$, $\mathfrak{e}_b^{(d)}$ and $\mathfrak{e}_s^{(d)}$ for the interpolation, bias and statistical errors; the first quantifying the error due to the spline interpolation, the second the error due to computing the QoI only with a finite accuracy at level L and the third being the statistical error due to MLMC sampling.

The cost and thus the complexity of the MLMC estimator is determined by three different sets of parameters - the number of interpolation points n, the level-wise sample sizes m_l at each level l and the number of levels L, each of which control respectively the errors $\mathfrak{e}_i^{(d)}$, $\mathfrak{e}_b^{(d)}$ and $\mathfrak{e}_s^{(d)}$. These should be chosen in a cost optimal way based on suitable a priori or a posteriori error estimates.

2.1.3 Error estimators

We introduce in this section error estimators for each of the terms in (2.4) based on the works of Krumscheid and Nobile 2018 and Ayoul-Guilmard, Ganesh, Krumscheid et al. 2021. The main idea is to exploit the properties of the particular form of ϕ given in (2.1). Since $\phi(\theta, Q)$ is Lipschitz continuous in Q, uniformly in $\theta \in \Theta$, we have that

$$|\phi(\theta, Q_l) - \phi(\theta, Q_{l-1})| \leqslant C_{\mathrm{Lip}}|Q_l - Q_{l-1}| \qquad \forall\theta \in \Theta$$

for $C_{\mathrm{Lip}} \in \mathbb{R}^+$.

We first begin with the interpolation error $\mathfrak{e}_i^{(d)}$, for which we propose the following error estimator:

$$\hat{\mathfrak{e}}_i^{(d)} := C_1(d)\left\|\Phi^{(4)}\right\|_{\mathrm{L}^\infty(\Theta)}\left(\frac{|\Theta|}{n}\right)^{(4-d)},$$

where the constant $C_1(d)$ is related to properties of the cubic spline interpolant. The procedure to estimate $\left\|\Phi^{(4)}\right\|_{\mathrm{L}^\infty(\Theta)}$ is as follows. We begin by selecting the level $\lceil L/2\rceil$ from the hierarchy. This level is selected since $m_{\lceil L/2\rceil}$ is sufficiently large to justify the approach described next, but still providing a good approx $\hat{\Phi}_{\lceil L/2\rceil}$ of Φ. kernel density estimation (KDE) techniques are used to produce a smoothed function estimate $\Upsilon_{\lceil L/2\rceil}(\theta) := \mathbb{E}^{\mathrm{kde}}_{\lceil L/2\rceil}(\phi(\theta,\cdot))$ of the function $\hat{\Phi}_{\lceil L/2\rceil}$. The true expectation is replaced by a KDE approximation based on isotropic Gaussian kernels centred on the $m_{\lceil L/2\rceil}$ samples of $Q_{\lceil L/2\rceil}$ collected at level $\lceil L/2\rceil$. The fourth derivative $\Upsilon^{(4)}_{\lceil L/2\rceil}$ is then computed using a second order central difference approximation where $\Upsilon_{\lceil L/2\rceil}$ is evaluated on a uniform grid on Θ with $n' \gg n$ points. The norm as well is approximated on the same grid as

$$\left\|\Upsilon^{(4)}_{\lceil L/2\rceil}\right\|_{\mathrm{L}^\infty(\Theta)} \approx \left\|\Upsilon^{(4)}_{\lceil L/2\rceil}\right\|_{\ell^\infty(\Theta)} := \max_{i\in[\![1,n']\!]}\left|\Upsilon^{(4)}_{\lceil L/2\rceil}(\theta_i)\right|. \tag{2.5}$$

For the bias error, we first note that $\lim_{l\to\infty}\mathbb{E}(\phi(\cdot,Q_l)) = \Phi$. This, in combination with the bias decay hypothesis from proposition 1, allows us to rewrite the bias term as

$$\begin{aligned}
\mathfrak{e}_b^{(d)} &= \left\|\mathcal{S}_n^{(d)}\left(\Phi(\boldsymbol{\theta}) - \mathbb{E}\left(\hat{\Phi}_L(\boldsymbol{\theta})\right)\right)\right\|_{\mathrm{L}^\infty(\Theta)} \\
&= \left\|\sum_{l=L}^{\infty}\mathcal{S}_n^{(d)}(\mathbb{E}(\phi_{l+1}(\boldsymbol{\theta}) - \phi_l(\boldsymbol{\theta})))\right\|_{\mathrm{L}^\infty(\Theta)} \\
&\approx \frac{\left\|\mathcal{S}_n^{(d)}(\mathbb{E}(\phi_L(\boldsymbol{\theta}) - \phi_{L-1}(\boldsymbol{\theta})))\right\|_{\mathrm{L}^\infty(\Theta)}}{(\mathrm{e}^{r_1}-1)},
\end{aligned}$$

where r_1 denotes the exponential decay rate of the differences $\left\|\mathbb{E}(\phi_l - \phi_{l-1})\right\|_{\mathrm{L}^\infty(\Theta)}$ in the levels. The expectation on the right hand side is estimated using a similar KDE approach as before wherein Gaussian kernels are centred at the m_L finest sample pairs to create a surrogate density. Given the known form of ϕ, one can derive a closed form expression for the right hand side. As a result, we propose the following computable error estimator for $\mathfrak{e}_b^{(d)}$:

$$\hat{\mathfrak{e}}_b^{(d)} := \frac{\left\|\mathcal{S}_n^{(d)}\left(\mathbb{E}^{\mathrm{kde}}_L(\phi_L(\boldsymbol{\theta},\cdot) - \phi_{L-1}(\boldsymbol{\theta},\cdot))\right)\right\|_{\ell^\infty(\boldsymbol{\theta})}}{(\mathrm{e}^{r_1}-1)},$$

where $\mathbb{E}_l^{\text{kde}}$ denotes the expectation calculated with respect to the KDE-smoothed density using the m_l samples at level l and again, the continuous norm $\|\cdot\|_{\mathrm{L}^\infty(\Theta)}$ has been replaced by its discrete counterpart over the finite set of points $\boldsymbol{\theta} = \{\theta_i : i \in [\![1, n]\!]\}$.

For the statistical error, we proceed as follows. The statistical error term in (2.4) has the form

$$(\mathfrak{e}_s^{(d)})^2 = \mathbb{E}\left(\left\|\mathcal{S}_n^{(d)}\left(\hat{\Phi}_L(\boldsymbol{\theta}) - \mathbb{E}\left(\hat{\Phi}_L(\boldsymbol{\theta})\right)\right)\right\|^2_{\mathrm{L}^\infty(\Theta)}\right). \tag{2.6}$$

We propose the use of the bootstrapping technique (Tibshirani and Efron 1993) to estimate this term as follows. First, observe that a MLMC estimator $\hat{\Phi}_L^{(d)}$ is defined through the hierarchy of samples denoted by

$$\mathcal{Q} \equiv \left\{\{Q_l^{(i,l)}, Q_{l-1}^{(i,l)}\}_{i=1}^{m_l}\right\}_{l=0}^{L},$$

with the convention $Q_{-1} = 0$. The idea behind bootstrapping is to create $m_{\text{bs}} \in \mathbb{N}$ new MLMC estimators $\{\Psi_j : j \in [\![1, m_{\text{bs}}]\!]\}$, each defined by a hierarchy of samples of the same size as the original hierarchy $\mathcal{Q}$. For each Ψ_j, this is done by randomly selecting m_l sample pairs $(\tilde{Q}_l^{(i,l)}, \tilde{Q}_{l-1}^{(i,l)}) = (Q_l^{(j_i,l)}, Q_{l-1}^{(j_i,l)})$, $i \in [\![1, m_l]\!]$ with $j_1, ..., j_{m_l} \overset{\text{i.i.d}}{\sim} \{1, ..., m_l\}$ at each level l to define a hierarchy $\mathcal{Q}_j$ as follows:

$$\mathcal{Q}_j \equiv \left\{\{\tilde{Q}_l^{(i,l)}, \tilde{Q}_{l-1}^{(i,l)}\}_{i=1}^{m_l}\right\}_{l=0}^{L}, \quad j \in [\![1, m_{\text{bs}}]\!].$$

The bootstrapped MLMC estimate Ψ_j defined through $\mathcal{Q}_j$ then is also an estimator of Φ. Using the m_{bs} bootstrapped MLMC estimators, one can approximate each of the expectations in (2.6) by a sample average over the bootstrapped samples. That is, the statistical error $\mathfrak{e}_s^{(d)}$ can be estimated by the bootstrapped estimate $\hat{\mathfrak{e}}_s^{(d)}$ as follows:

$$(\hat{\mathfrak{e}}_s^{(d)})^2 := \frac{1}{m_{\text{bs}}} \sum_{j=1}^{m_{\text{bs}}} \left\|\mathcal{S}_n^{(d)}\left(\Psi_j(\boldsymbol{\theta}) - \bar{\Psi}(\boldsymbol{\theta})\right)\right\|^2_{\mathrm{L}^\infty}, \tag{2.7}$$

where $\bar{\Psi}$ denotes the sample average over $\{\Psi_j\}_{j=1}^{m_{\text{bs}}}$. The norm is estimated using a uniform fine grid on Θ with $n' \gg n$ points as in (2.5). The choice of m_{bs} is made adaptively. First, it is set to an initial fixed value. Then, since (2.7) is a Monte Carlo estimator, the sample variance of the L^∞ norms is used to estimate the error on the statistical error estimate. If this error exceeds a fixed fraction of the statistical error tolerance ϵ_s^2, the number of bootstrapped samples m_{bs} is doubled and the process is repeated until the tolerance is satisfied.

In summary, we present below the error bounds on the MSE of the function derivative $\hat{\Phi}_L^{(d)}$.

$$\begin{aligned}\mathrm{MSE}\big(\hat{\Phi}_L^{(d)}\big) \leqslant 3C_1^2(d)\big\|\Upsilon^{(4)}_{\lceil L/2\rceil}\big\|^2_{\ell^\infty(\Theta)}\left(\frac{|\Theta|}{n}\right)^{2(4-d)} \\ + 3\frac{\Big\|\mathcal{S}_n^{(d)}\Big(\mathbb{E}_L^{\mathrm{kde}}(\phi_L(\boldsymbol{\theta},\cdot)-\phi_{L-1}(\boldsymbol{\theta},\cdot))\Big)\Big\|^2_{\ell^\infty(\boldsymbol{\theta})}}{(\mathrm{e}^{r_1}-1)^2} \\ + \frac{3}{m_{\mathrm{bs}}}\sum_{j=1}^{m_{\mathrm{bs}}}\big\|\mathcal{S}_n^{(d)}\big(\Psi_j(\boldsymbol{\theta})-\bar{\Psi}(\boldsymbol{\theta})\big)\big\|^2_{\ell^\infty(\Theta)}. \quad (2.8)\end{aligned}$$

2.1.4 Adaptive Strategy

The next step will be to adapt the MLMC hierarchy based on the error estimators presented in section 2.1.3 to achieve a prescribed tolerance on the MSE in a cost-optimal way. This implies that one should choose adaptively the number of interpolation points n, the maximum discretisation level L and the level-wise sample sizes m_l. We discuss a possible way to do this for the MSE on $\hat{\Phi}_L^{(d)}$ and quantities derived from it, focusing on the VaR and the CVaR as defined in (2.2).

We first present a method to calibrate the MLMC hierarchy for $\hat{\Phi}_L^{(d)}$. We require the MSE to satisfy a tolerance ϵ^2 with each of the error contributions $\mathfrak{e}_i^{(d)}$, $\mathfrak{e}_b^{(d)}$ and $\mathfrak{e}_s^{(d)}$ satisfying a fraction w_1, w_2, w_3 respectively such that $w_1+w_2+w_3=1$. We define the following tolerances,

$$\epsilon_i^2 = \frac{w_1\epsilon^2}{3}, \qquad \epsilon_b^2 = \frac{w_2\epsilon^2}{3}, \qquad \epsilon_s^2 = \frac{w_3\epsilon^2}{3},$$

and require that each of the errors satisfies the respective tolerance.

The interpolation error is controlled by the number of interpolation points. We require it to satisfy the tolerance ϵ_i^2, yielding

$$\begin{aligned} C_1(d)\big\|\Upsilon^{(4)}_{\lceil L/2\rceil}\big\|_{\ell^\infty(\Theta)}\left(\frac{|\Theta|}{n}\right)^{(4-d)} &\leqslant \epsilon_i \\ \implies n = \left(\frac{C_1(d)\big\|\Upsilon^{(4)}_{\lceil L/2\rceil}\big\|_{\ell^\infty(\Theta)}}{\epsilon_i}\right)^{\frac{1}{(4-d)}} &|\Theta|. \quad (2.9)\end{aligned}$$

The bias error is controlled by the number of levels. The bias error term takes the form $\hat{\mathfrak{e}}_b^{(d)} = c_1\mathrm{e}^{-Lr_1}/(\mathrm{e}^{r_1}-1)$, where the model $c_1\mathrm{e}^{-lr_1}$ is constructed by least

squares fit of the level-wise bias estimates $\left\|\mathcal{S}_n^{(d)}\left(\mathbb{E}_l^{\text{kde}}(\phi_l(\boldsymbol{\theta},\cdot)-\phi_{l-1}(\boldsymbol{\theta},\cdot))\right)\right\|_{\ell^\infty(\boldsymbol{\theta})}$ for $l \in [\![1, L]\!]$. We then pick the minimum level L^* such that this bias error satisfies a tolerance of ϵ_b. This is given by

$$L^* = \left\lceil \frac{1}{r_1} \log\left(\frac{c_1}{(\mathrm{e}^{r_1}-1)\epsilon_b} \right) \right\rceil. \tag{2.10}$$

For the choice of the level-wise sample sizes, this requires a notion of how each level contributes to the overall statistical error. An a priori estimator for $\mathfrak{e}_s^{(d)}$ was discussed in Krumscheid and Nobile 2018 where the error naturally splits over the levels as

$$(\mathfrak{e}_s^{(d)})^2 \approx \sum_{l=0}^{L} K(n,d)\frac{V_l}{m_l}.$$

Here, $K(n,d)$ is a constant related to the properties of the cubic spline interpolant, and the level-wise variances V_l are defined as:

$$V_l = \mathbb{E}\left(\|\phi_l(\cdot)-\phi_{l-1}(\cdot)\|^2_{\ell^\infty(\boldsymbol{\theta})}\right). \tag{2.11}$$

In practice, V_l is replaced by its estimate $\hat{V}_l$ which is computed by estimating the expectation (2.11) with a sample average over the m_l samples at level l. On the other hand, the new error estimator $\hat{\mathfrak{e}}_s^{(d)}$ based on bootstrapping does not provide such a notion of how each level contributes to the overall statistical error. To overcome this limitation, we propose to use the error splitting of the a priori estimator, however replacing the large constant $K(n,d)$, which is responsible for an overly conservative error bound, with a new one so that the total error matches the estimator $\hat{\mathfrak{e}}_s^{(d)}$. In particular, we redefine the level-wise variances so that

$$(\hat{\mathfrak{e}}_s^{(d)})^2 = \sum_{l=0}^{L} \frac{\tilde{V}_l}{m_l}.$$

This is done by rescaling $\hat{V}_l$ to yield the redefined variances $\tilde{V}_l$ as follows:

$$\tilde{V}_l = \frac{\hat{V}_l}{\sum_{k=0}^{L} \frac{\hat{V}_k}{m_k}} (\hat{\mathfrak{e}}_s^{(d)})^2.$$

Once the statistical error contributions are localised to each level, the level-wise sample sizes m_l, $l \in [\![0, L]\!]$ are then selected to minimise the cost of computing the MLMC estimator:

$$\mathfrak{C}\left(\hat{\Phi}_L^{(d)}\right) \leqslant \sum_{l=0}^{L} m_l(\mathfrak{c}_l + n\mathfrak{c}_\phi) + n\mathfrak{c}_{\text{int}},$$

subject to the constraint on the statistical error that

$$\sum_{l=0}^{L} \frac{\tilde{V}_l}{m_l} \leqslant \epsilon_s^2.$$

Here, $\mathfrak{c}_l = €(Q_l) + €(Q_{l-1})$ is the cost of computing one correlated pair of simulations (Q_l, Q_{l-1}) at level l, $\mathfrak{c}_\phi$ is the constant that bounds the cost of evaluating the function $\phi(\theta, Q)$ for any $(\theta, Q) \in \Theta \times \mathbb{R}$ and $\mathfrak{c}_{\text{int}}$ is the cost of constructing the uniform cubic spline interpolant over n points. In Krumscheid and Nobile 2018, the level-wise sample sizes were selected under the assumption that $\mathfrak{c}_{\text{int}}$ and $\mathfrak{c}_\phi$ were non-zero. However, for the applications addressed in this work, it was found that $\mathfrak{c}_{\text{int}}$ and $\mathfrak{c}_\phi$ are often negligible in comparison to the cost of $\mathfrak{c}_l$. Hence, we select the level-wise sample sizes $\{m_l\}_{l=0}^L$ based on the simplified cost model

$$€\Big(\hat{\Phi}_L^{(d)}\Big) \approx \sum_{l=0}^{L} m_l \mathfrak{c}_l. \tag{2.12}$$

Consequently, the level-wise sample sizes are selected as, $\forall l \in [\![0, L]\!]$,

$$m_l = \left\lceil \frac{1}{\epsilon_s^2} \sqrt{\frac{\tilde{V}_l}{\mathfrak{c}_l}} \sum_{k=0}^{L} \sqrt{\tilde{V}_k \mathfrak{c}_k} \right\rceil. \tag{2.13}$$

Below, we present a complexity result based on the simplified cost model using the bounds presented in (2.8) and the corresponding adaptive strategies. This result is a simplified version of the one presented in ibid.

PROPOSITION 1. *We define the quantities:*

$$b_l := \|\Phi - \mathbb{E}(\phi_l(\cdot))\|_{\ell^\infty(\boldsymbol{\theta})}, \quad \text{and} \quad V_l := \mathbb{E}\Big(\|\phi_l(\cdot) - \phi_{l-1}(\cdot)\|^2_{\ell^\infty(\boldsymbol{\theta})}\Big).$$

Suppose that there exist positive constants r_1, r_2, and r_3 such that $2r_1 \geqslant \min(r_2, r_3)$ and that

(I) *b_l decays exponentially with order $r_1 > 0$ in l, in the sense that $b_l \leqslant c_1 \mathrm{e}^{-l r_1}$ for some constant $c_1 > 0$,*

(II) *V_l decays exponentially with order $r_2 > 0$ in l, in the sense that $V_l \leqslant c_2 \mathrm{e}^{-l r_2}$ for some constant $c_2 > 0$,*

(III) *the cost to compute each i.i.d. realisation of (Q_l, Q_{l-1}) increases exponentially with rate $r_3 > 0$ in l, in the sense that $\mathfrak{c}_l = €(Q_l, Q_{l-1}) \leqslant c_3 \mathrm{e}^{l r_3}$ for some constant c_3,*

for all $l \in \mathbb{N}_0$*, when* $h_{l-1} = sh_l$ *for some* $s > 1$ *and* $d \in \{0, 1, 2\}$*. For any* $0 < \epsilon < \mathrm{e}^{-1}$*, the* d^{th} *derivative of the* MLMC *estimator* $\hat{\Phi}_L$ *of* $\Phi \in C^4(\Theta)$ *with the number* n *of (uniform) nodes chosen according to* (2.9)*, the maximum number of levels* L *as in* (2.10)*, and level-wise sample sizes* m_l *by* (2.13)*, satisfies* $\mathrm{MSE}\big(\hat{\Phi}_L^{(d)}\big) \leqslant \epsilon^2$ *at a computational cost that is bounded by*

$$\mathscr{C}\big(\hat{\Phi}_L^{(d)}\big) \lesssim \log(\epsilon^{-1})\epsilon^{\frac{-8}{4-d}} \begin{cases} 1, & \text{if } r_2 > r_3, \\ \log(\epsilon^{-1})^2, & \text{if } r_2 = r_3, \\ \epsilon^{\frac{r_2 - r_3}{r_1}\frac{4}{4-d}}, & \text{if } r_2 < r_3, \end{cases}$$

$\Diamond$

We now present a result to demonstrate that the MSE on the VaR and the CVaR is related to the MSE on $\hat{\Phi}_L^{(d)}$.

LEMMA 1. *Let* $\hat{\Phi}_L$ *be the multi-level Monte Carlo estimator defined in* (2.3) *to approximate* Φ*. If there exist* $\hat{q}_\tau, q_\tau \in \Theta$ *such that* $\hat{\Phi}_L(\hat{q}_\tau) = \tau = \Phi(q_\tau)$ *for some given* $\tau \in \mathbb{R}$*, then it holds that*

$$\mathbb{E}\big(|\hat{q}_\tau - q_\tau|^2\big) \leqslant \left\| \frac{1}{\Phi^{(2)}} \right\|^2_{\mathrm{L}^\infty([\hat{q}_\tau, q_\tau])} \mathrm{MSE}\big(\hat{\Phi}_L^{(1)}\big),$$

as well as that

$$\mathbb{E}\big(|\hat{c}_\tau - c_\tau|^2\big) \leqslant 2\big\|\Phi^{(1)}\big\|^2_{\mathrm{L}^\infty([\hat{q}_\tau, q_\tau])} \left\| \frac{1}{\Phi^{(2)}} \right\|^2_{\mathrm{L}^\infty([\hat{q}_\tau, q_\tau])} \mathrm{MSE}\big(\hat{\Phi}_L^{(1)}\big) + 2\,\mathrm{MSE}\big(\hat{\Phi}_L\big),$$

where $\hat{c}_\tau = \hat{\Phi}_L(\hat{q}_\tau)$ *and* $c_\tau = \Phi(q_\tau)$*.* $\Diamond$

The results demonstrate that the same adaptive strategies can be used for statistics whose MSE can be bounded by linear combinations of the MSE on $\hat{\Phi}_L^{(d)}$ with different values of d. For the number of interpolation points n, a closed form expression can be obtained for the VaR. However, since the MSE on the CVaR is a linear combination of the MSEs on $\Phi^{(0)}$ and $\Phi^{(1)}$, the interpolation error is given by an 8^{th}-order polynomial in n. The smallest integer larger than the smallest positive real root is taken to be the number of interpolation points in this case. For the number of levels L and level-wise sample sizes m_l, the methods described earlier can be directly used. Lastly, since we expect the interval $[q_\tau, \hat{q}_\tau]$ to be small, we replace each of the constants $\big\|\Phi^{(1)}\big\|_{\mathrm{L}^\infty(\Theta)}$ with $\big|\hat{\Phi}_L^{(1)}(\hat{q}_\tau)\big|$ and $\big\|1/\Phi^{(2)}\big\|_{\mathrm{L}^\infty(\Theta)}$ with $\big|1/\hat{\Phi}_L^{(2)}(\hat{q}_\tau)\big|$.

2.1.5 Adaptive multi-level Monte Carlo algorithm

It was shown earlier that the cost optimal number of interpolation points, level-wise sample sizes and the number of levels could be calculated with knowledge of the quantities b_l, V_l and $\mathfrak{c}_l$ and their corresponding decay rates and constants. However, these constants and rates are rarely readily available. Theoretical considerations of the underlying model can give their decay rates but typically not explicit constants. One alternative is to compute an initial small hierarchy of a pre-determined size. Rates and constants can then be fitted on estimates of the level-wise bias, variance and cost obtained from this hierarchy. However, it is possible that one obtains inaccurate initial estimates with such a small hierarchy. This can lead to unreliable behaviour of the error estimators.

We propose the use of a variation of the continuation multi-level Monte Carlo (CMLMC) algorithm introduced in Collier et al. 2015. The algorithm was used in deliverable 5.4 to estimate the expected value of a forced oscillator problem. The CMLMC algorithm begins with a screening hierarchy and a geometrically decreasing sequence of tolerances $\epsilon_0 > \epsilon_1 > \cdots > \epsilon_N = \epsilon$ where ϵ is the target tolerance. One then adapts for the tolerance ϵ_i based on the estimates from the hierarchy tuned on ϵ_{i-1}. For ϵ_0, one uses the estimates from the screening hierarchy. The advantage of this method is that the estimates of b_l, V_l and $\mathfrak{c}_l$ are successively improved. This makes the algorithm more robust to inaccurate initial estimates from the screening hierarchy. The algorithmic description of the CMLMC algorithm is presented in algorithm 1. The reader is referred to Collier et al. 2015 for a more detailed overview.

Algorithm 1: Continuation multi-level Monte Carlo algorithm

```
Data: Sequence of tolerances
      ε_0 > ε_1 > ··· > ε_N = ε, ε_j = ελ^(N−j), λ > 1, κ > 1
Result: Optimally calibrated MLMC estimator
Compute screening hierarchy, i = 0
While i ⩽ N or MSE ⩾ ε^2 do
    If i ⩽ N then
        Set ε_a = ε_i
    else
        Set ε_a = ε/κ^(i−N)
    Compute estimates b_l, V_l, c_l and model parameters c_1, c_2, c_3, r_1, r_2, r_3
    Compute n(ε_a), {m_l(ε_a)}_{l=0}^L, L(ε_a)
    Compute new QoI based on above hierarchy and new error estimates
```

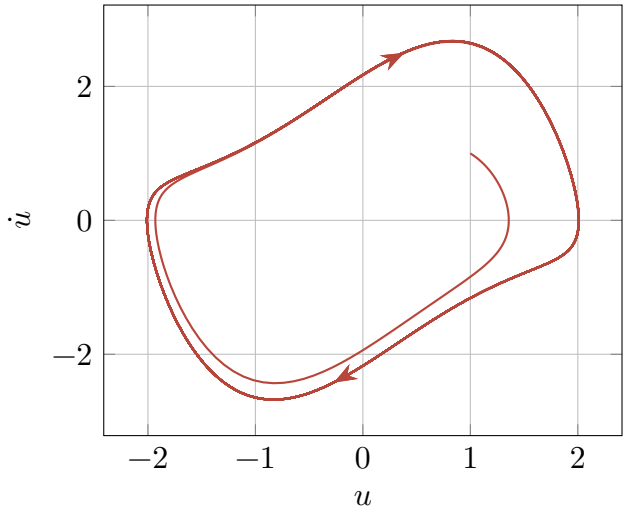


Figure 1: Example of trajectory of an unforced Van der Pol oscillator in the phase space

2.2 Numerical experiments

2.2.1 Van der Pol Oscillator

The Van der Pol oscillator (Van der Pol 1920) is a non-linear oscillator that has been used as a simple model for vortex shedding (e.g. by Ehsan and Scanlan 1990). The oscillator was presented in deliverable 5.4 as an alternative to study the MLMC method for time dependent problems. With an uncertain forcing term, it provides us with a non-chaotic, unsteady problem that is simple to implement, computationally easy to solve and with a suitable asymptotic behaviour in time.

Let W denote a standard Wiener process. We define the Van der Pol oscillator with stochastic forcing as follows for $u : [0, T] \to \mathbb{R}$ which satisfies

$$\begin{aligned} \ddot{u} - z(1 - u^2)\dot{u} + u &= \sigma \dot{W} \quad \text{over }]0, T], \\ u(0, \cdot) &= u_0, \\ \dot{u}(0, \cdot) &= \dot{u}_0, \end{aligned}$$

where $z \in \mathbb{R}^+$ is a property of the oscillator that determines the shape of its limit cycle and $u_0, \dot{u}_0 \in \mathbb{R}^+$ are initial conditions. Setting σ to 0 yields the canonical unforced Van der Pol oscillator, which is asymptotically periodic with a period and amplitude independent of the initial conditions $(u_0, \dot{u}_0)$. An example of such a trajectory is shown in figure 1.

To numerically solve the system, we first re-write it as an ordinary differential

equation (ODE) system:

$$\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = \begin{pmatrix} v \\ z(1-u^2)v - u + \sigma\dot{W}\end{pmatrix}$$

We discretise the interval $[0,T]$ with a hierarchy of uniform grids $t_j = j\,\Delta t_l$ with $\Delta t_l = \Delta t_0/2^l$. The discretised system reads as follows for a given Δt:

$$\begin{pmatrix}u_{n+1}\\ v_{n+1}\end{pmatrix} = \begin{pmatrix}u_n\\ v_n\end{pmatrix} + \Delta t\begin{pmatrix} v_n \\ z(1-u_n^2)v_n - u_n\end{pmatrix} + \sqrt{\Delta t}\begin{pmatrix}0\\ \sigma\xi_n\end{pmatrix}, n \in \mathbb{N},$$

where u_n denotes the approximation to $u(t_n)$. The QoI considered in this study is the time average of u^2 over the interval $[0,T]$ defined as below:

$$Q = \langle u^2 \rangle_T,$$

where the definition of the time-average is introduced in (1.5). The integral is approximated using a piecewise constant approximation as follows:

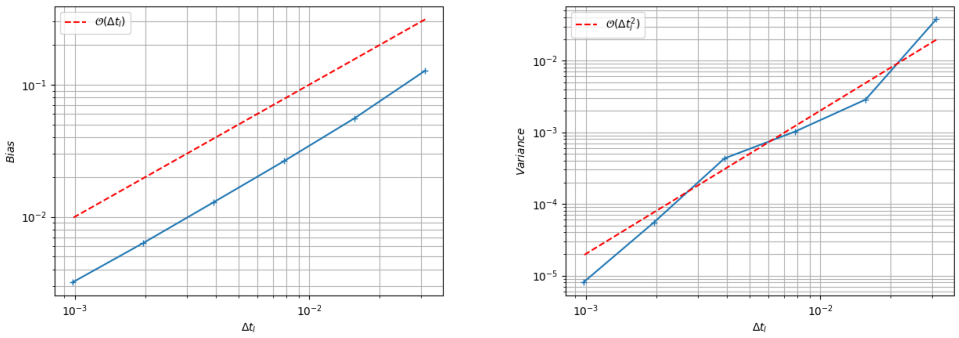
$$Q \approx Q_l := \sum_{n=0}^{N_{T,l}-1} u_n^2\,\Delta t_l, \quad N_{T,l} = T/\Delta t_l.$$

We are in particular interested in estimating the 70 %-CVaR of the QoI up to a prescribed accuracy. We use the estimator described in section 2.1.2. The error in the estimator is estimated using the procedure described in section 2.1.3. The parameters of the MLMC hierarchy are selected based on these estimates using the methodology of section 2.1.4. The CMLMC algorithm as described in algorithm 1 is used to optimally calibrate the MLMC estimator $\hat{\Phi}_L$ with $\lambda = 1.5$ and $\kappa = 1.1$. The combination of the above is implemented in XMC, which was used extensively for the studies carried out in this report.

We report in figures 2a and 2b the variation of the biases b_l and variances V_l with time step size Δt_l. From theoretical considerations, we know that the differences in the QoI decay with rates 1 and 2 in the time step size for the bias and variance respectively. We find that this is also the case for the parametric expectation Φ and the corresponding bias and variance terms b_l and V_l.

We also study the variation of the level-wise sample sizes for different iterations of the CMLMC algorithm to assess its behaviour. In figure 3, we show the level-wise samples for different CMLMC iterations for one run of the algorithm for the finest tested tolerance. As can be seen in figure 3, the level-wise samples are increased in every iteration due to the decreasing sequences of tolerances, leading to a more robust algorithm.

To assess the robustness of the error estimators, a reliability study is conducted. For a given tolerance, the entire MLMC simulation is carried out 20 times. For each



(a) Bias decay (b) Variance decay

Figure 2: Convergence properties for the Van der Pol oscillator

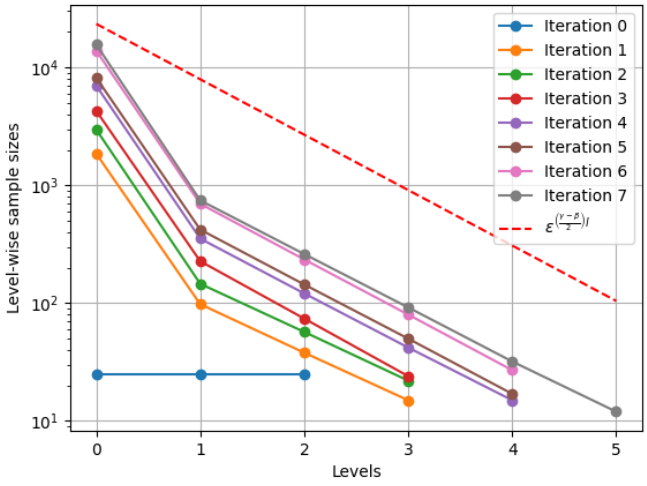
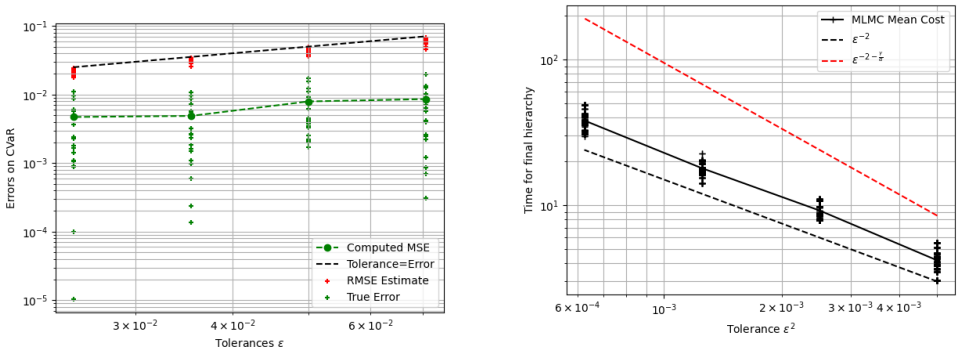


Figure 3: Level-wise sample sizes in the CMLMC algorithm



(a) Reliability of error estimator

(b) Complexity behaviour of CMLMC estimator

Figure 4: Summary of results for the Van der Pol oscillator

simulation, an estimate of the CVaR and a corresponding estimate of the MSE are produced. Since the true value of the CVaR is known from a reference simulation, we compute the corresponding true squared error. We expect the MSE estimates to be a tight bound on the sample average of the true squared errors, which we take here as the reference value for the true MSE. The results of this study are shown in figure 4a. As can be seen from the figure, the error estimates bound the error on the CVaR and lead to practically computable hierarchies. For all the tolerances tested, the error estimate is not larger than 10 times the true error, which we consider acceptable for practical applications.

To verify the predictions of proposition 1, we also compute the cost of each MLMC simulation according to (2.12). The time taken to compute each of the m_l samples is measured and $\mathfrak{c}_l$ is taken to be their average. The cost is computed using the level-wise sample sizes corresponding to the final iteration of the CMLMC that satisfies the target tolerance for all 20 simulations conducted for a given target tolerance. The results are summarised in figure 4b, where the average cost over all the simulations for each final tolerance is plotted versus the final tolerance. In addition, a cost is estimated for attaining the final tolerance using a simple Monte Carlo estimator instead of MLMC. This cost is shown together with a least squares fit rate over the estimated Monte Carlo cost.

The raw results of these numerical experiments can be found in the data set publically available as Ayoul-Guilmard, Rosa M. Badia et al. 2021.

2.2.2 Potential flow over an airfoil

In order to demonstrate the MLMC estimators for the CVaR, the potential flow over an airfoil is considered. This problem features the solution of the simplified expres-

sion of the mass equation, which requires a complete discretisation of the domain and the geometry. The mass equation is solved for an inviscid and irrotational flow, which effectively lowers the computational cost. This hypothesis is valid for a certain range of applications, especially on classical commercial aircraft flying at cruise speed. Thus, this problem stands as a convenient option for demonstrating purposes as it requires reduced computational resources. This is specially important in the context of uncertainty quantification problems, where many realisations are to be solved. Nonetheless, this problem still requires the complete discretisation of the geometry and the domain, so the complexity associated with performing adaptive mesh refinement, or the evaluation of many quantities of interests is still kept.

The method used to solve the full potential equation was presented by the authors in Davari et al. 2019 which features the solution of the problem on body-fitted meshes with an embedded wake. In Núñez et al. 2022, the solution was extended to a full embedded approach. This same method and the discussion of its sensitivity analysis was also introduced in deliverable 6.4. Also, the authors in Tosi, Amela et al. 2021 used this problem to develop and demonstrate an asynchronous framework to solve uncertainty quantification problems with hierarchical Monte Carlo algorithms. For the sake of completeness, the governing equations and discretisation introduced in the aforementioned documents is briefly summarised next.

Under certain flight conditions and for high Reynolds flows, the domain over a wing can be separated into two domains. An inviscid, irrotational region where the viscous and thermal effects are negligible, and a small region close to the wing where these effects cannot be neglected. The small viscous region is commonly known as wake. For the specific case of commercial aircraft at cruise speed, this region can be simplified to a single line. Thus, potential flow assumptions can be adopted for the whole domain except wake region, which is considered infinitely thin. This is depicted in figure 5.

Navier–Stokes equations can be simplified for this scenario, assuming that the flow is irrotational and inviscid, and assuming that the velocity can be written as the gradient of a potential such that $v = \nabla u$. This reduces the mass conservation equation to:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla u) = 0 \tag{2.14}$$

where the density, ρ, can be written in terms of far-field quantities and using the isentropic flow hypothesis:

$$\frac{\rho}{\rho_\infty} = \left(1 + \frac{\gamma - 1}{2} \frac{u_\infty^2}{a_\infty^2} \left(1 - \frac{\nabla u \cdot \nabla u}{u_\infty^2}\right)\right)^{\frac{1}{\gamma - 1}} \tag{2.15}$$

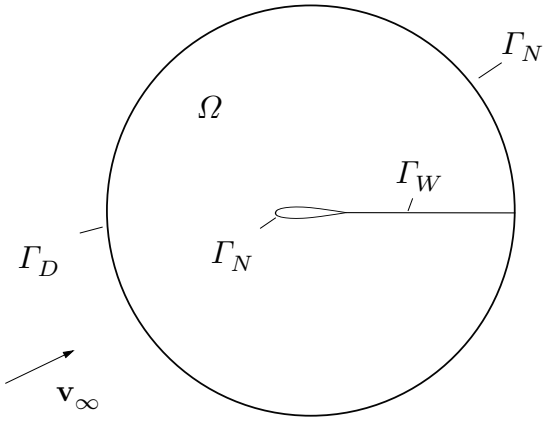


Figure 5: Simplified representation of the domain Ω and its boundary Γ. The wake Γ_W is modelled as a straight line. Extracted from Núñez et al. 2022

where pressure can be computed using the isentropic relation:

$$\frac{p}{p_\infty} = \left(\frac{\rho}{\rho_\infty}\right)^\gamma$$

The boundary conditions defined in the problem are equivalent as those introduced in Davari et al. 2019; a schematic is shown in figure 5, where the far-field boundary is represented by the outer circle. A line denoted by Γ_W represents the wake modelled in the problem. Depending on the flux passing through the domain, the imposed conditions are defined as Γ_D or Γ_N, which will be considered the inlet and the outlet of the domain, respectively. If $\boldsymbol{n}$ is defined as the outer normal on the boundary, then Γ_N and Γ_D are defined as follows.

$$\Gamma_\infty = \begin{cases} \Gamma_D & \text{if} \quad \boldsymbol{V}_\infty \cdot \boldsymbol{n} < 0 \\ \Gamma_N & \text{if} \quad \boldsymbol{V}_\infty \cdot \boldsymbol{n} \geqslant 0 \end{cases}$$

At the inlet, a Dirichlet condition is imposed by fixing an initial potential value as specified in (2.16). A Neumann condition is imposed in (2.17) at the outlet by setting a mass flux g. At the walls, a no-penetration condition is set with $g = 0$. Also, the wake requires the imposition of two specific boundary conditions: (2.18) imposes mass conservation across the wake; (2.19) imposes pressure equality between the upper and lower parts of the wake.

$$u = u_\infty \quad \text{on} \quad \Gamma_D, \tag{2.16}$$
$$\boldsymbol{n} \cdot (\rho \nabla u) = g \quad \text{on} \quad \Gamma_N, \tag{2.17}$$
$$\boldsymbol{n} \cdot (\rho^+ \nabla u^+ - \rho^- \nabla u^-) = 0 \quad \text{on} \quad \Gamma_W, \tag{2.18}$$
$$|\nabla u^+|^2 - |\nabla u^-|^2 = 0 \quad \text{on} \quad \Gamma_W, \tag{2.19}$$

From (2.14), the weak form of the system can be derived by applying the Galerkin method and the divergence theorem:

$$\int_{\Omega} \rho \nabla \varphi \cdot \nabla u \, \mathrm{d}\Omega = \int_{\Gamma_N} \varphi \boldsymbol{n} \cdot (\rho \nabla u) \, \mathrm{d}\Gamma.$$

This weak form is discretised by linear finite elements: it is expressed in terms of the M nodal shape functions N_j and the M corresponding nodal values u_j:

$$\forall i \in [\![1, M]\!], \quad \sum_{j=1}^{M} \int_{\Omega} \rho \nabla N_i \cdot \nabla N_j \, \mathrm{d}\Omega u_j = \int_{\Gamma_N} N_i g \, \mathrm{d}\Gamma. \tag{2.20}$$

The right-hand side of the equation has been replaced by the Neumann boundary condition. The dependence of ρ on the potential makes the problem non-linear, so the system is written in residual form $R(u) = 0$. The system is solved by means of the Newton method, which requires the calculation of the Jacobian $\mathfrak{J}$. The solution is updated by an increment Δu^k. For every element Ω_e in the mesh with boundary $\Gamma_{N_e} = \Gamma_N \cap \partial\Omega_e$, its contribution to the residual and the Jacobian is computed, denoted by R_i^e and $\mathfrak{J}_{i,j}^e$ respectively.

$$R_i^e(u) = \sum_{j=1}^{M} \int_{\Omega_e} \rho \nabla N_i \cdot \nabla N_j \, \mathrm{d}\Omega u_j - \int_{\Gamma_{N_e}} N_i g \, \mathrm{d}\Gamma,$$

$$\begin{aligned} \mathfrak{J}_{i,j}^e(u) &= \frac{\partial R_i^e(u)}{\partial u_j} \\ &= \int_{\Omega_e} \rho \nabla N_i \cdot \nabla N_j + 2 \frac{\partial \rho}{\partial |v|^2} (\nabla N_j \cdot \nabla u)(\nabla N_i \cdot \nabla u). \end{aligned} \tag{2.21}$$

In (2.21), the derivative of the density with respect to the local velocity is obtained from (2.15):

$$\frac{\partial \rho}{\partial |v|^2} = -\frac{\rho_\infty}{2a_\infty^2} \left(1 + \frac{\gamma - 1}{2} \frac{u_\infty^2}{a_\infty^2} \left(1 - \frac{|v|^2}{u_\infty^2} \right) \right)^{\frac{2-\gamma}{\gamma-1}}$$

The wake is implicitly defined with a level-set function that cuts the elements that form the wake, easing mesh generation. This was first introduced by Davari et al. 2019. The wake represents the enforced vorticity that is applied in the domain to generate lift under the potential flow hypothesis. To this end, the elements that are cut by the wake have duplicated degrees of freedom and use modified shape functions to deal with the split elements and to create a domain discontinuity. The domain is therefore split into two by the wake, as shown in figure 6. To refer to

each side of the wake, the '+' and the '−' signs will be used, representing the upper and lower sides of the wake respectively. This leads to the definition of the domains $\Omega^+_{W,e}$, $\Omega^-_{W,e}$, $\Gamma^+_{W,e} = \Gamma_N \cap \partial\Omega^+_{W,e}$, $\Gamma^-_{W,e} = \Gamma_N \cap \partial\Omega^-_{W,e}$, which refer to the elements and the boundary of the elements that form the wake, for which the system of equations for the wake will be defined.

The duplicated degrees of freedom on the wake elements are denoted by Θ. The residuals and Jacobians derived from the full potential equation are rewritten for the elements that are intersected by the wake, denoted with $R^{e,+}_{i,W}$, $R^{e,-}_{i,W}$ and $\mathfrak{J}^{e,-}_{i,j,W}$, $\mathfrak{J}^{e,-}_{i,j,W}$ respectively. As detailed previously, the sign denotes if the equations are evaluated in the upper or lower side of the wake.

Thus, (2.20) can be rewritten into:

$$R^{e,+}_{i,W}(\Theta^+) = \sum_{j=1}^{M} \int_{\Omega^+_{W,e}} \rho^+ \nabla \hat{N}^+_i \cdot \nabla \hat{N}^+_j \, \mathrm{d}\Omega\Theta^+_j - \int_{\Gamma^+_{W,e}} \hat{N}^+_i g \, \mathrm{d}\Gamma,$$

$$R^{e,-}_{i,W}(\Theta^-) = \sum_{j=1}^{M} \int_{\Omega^-_{W,e}} \rho^- \nabla \hat{N}^-_i \cdot \nabla \hat{N}^-_j \, \mathrm{d}\Omega\Theta^-_j - \int_{\Gamma^-_{W,e}} \hat{N}^-_i g \, \mathrm{d}\Gamma.$$

Where $\hat{N}_i$ denotes the modified shape functions due to the element cutting. From these residuals, Jacobians for the wake elements are derived in the same manner as in (2.21), by computing $\mathfrak{J}^{e,+}_{i,j,W} = \frac{\partial R^e_i(\Theta^+)}{\partial \Theta^+_j}$ and $\mathfrak{J}^{e,-}_{i,j,W} = \frac{\partial R^e_i(\Theta^-)}{\partial \Theta^-_j}$. Here, Θ^+ and Θ^- are composed as:

$$\Theta^+_i = \begin{cases} u_i & \text{if } i^{\text{th}} \text{ node} \in \Omega_{W^-} \\ \Psi_i & \text{if } i^{\text{th}} \text{ node} \in \Omega_{W^-} \end{cases} \tag{2.22}$$

$$\Theta^-_i = \begin{cases} \Psi_i & \text{if } i^{\text{th}} \text{ node} \in \Omega_{W^+} \\ u_i & \text{if } i^{\text{th}} \text{ node} \in \Omega_{W^-} \end{cases} \tag{2.23}$$

This division effectively duplicates the degrees of freedom on the wake elements in order to allow for a discontinuity in the potential. Consider the element in figure 6, where the wake splits the element, leaving one node above the wake and two nodes below the wake. Each node has duplicated degrees of freedom u_i, Ψ_i. When evaluating the upper terms of the wake in Ω_{W^+}, (2.22) is used, assigning u_i and Ψ_i accordingly. If the lower terms are evaluated in Ω_{W^-}, (2.23) is used instead.

Boundary conditions for both sides of the wake are imposed by a least-square finite-element approach, combining (2.18) and (2.19) in a single two-dimensional vector equation, by minimising the functional:

$$\Pi(\Theta^+, \Theta^-) = \frac{1}{2} \int_{\Gamma_W} \| \rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^- \|^2 \, \mathrm{d}\Gamma = 0 \tag{2.24}$$

This expression acts as a constraint in the element formulation. The residual terms and its Jacobian are found by deriving the functional in (2.24), where derivatives

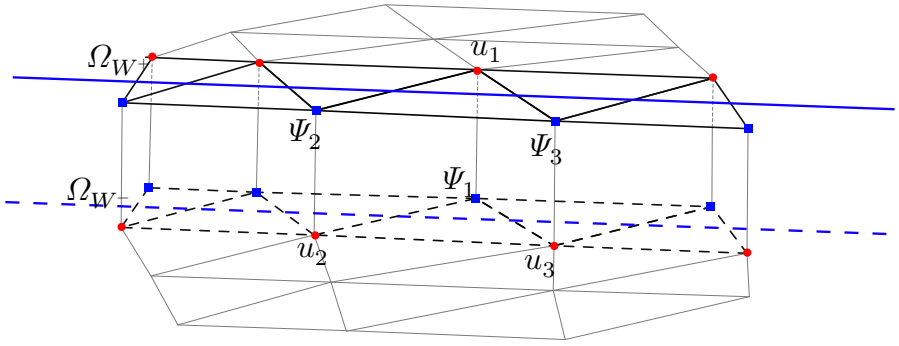


Figure 6: Embedded wake visualisation, where the wake is represented by a blue line. The elements intersected by the wake have duplicated degrees of freedom, which creates a discontinuity in the domain, as if the mesh was split in two parts. Note that the separation shown in the figure is simply a representation, as the mesh is not physically split in this approach. Extracted from Núñez et al. 2022.

on the density are neglected due to the assumption of small velocity perturbations $u = v - V_\infty \ll V_\infty$, as only streamlined bodies without sharp edges are considered. Here, u represents the velocity difference between the free-stream unperturbed flow V_∞, and the actual velocity around the airfoil v. The elemental contributions of the residual for the constraint are written with the subscript $R^{e,+}_{i,B}$, to refer to the terms enforcing the boundary conditions of the wake:

$$R^{e,+}_{i,B}(\Theta) = \frac{\partial \Pi(\Theta)}{\partial \Theta^+_i} = \int_{\Gamma^+_{W,e}} \nabla \hat{N}^+_i (\rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^-)\, \mathrm{d}\Gamma$$

$$R^{e,-}_{i,B}(\Theta) = \frac{\partial \Pi(\Theta)}{\partial \Theta^-_i} = -\int_{\Gamma^-_{W,e}} \nabla \hat{N}^-_i (\rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^-)\, \mathrm{d}\Gamma$$

where Jacobians are computed and renamed to $B^{e,+}_{i,j}$ and $B^{e,-}_{i,j}$ to refer to the terms enforcing the boundary conditions of the wake.

$$B^{e,+}_{i,j} = \frac{\partial R^{e,+}_{i,B}}{\partial \Theta^+_j}(\Theta) = \int_{\Gamma^+_{W,e}} \rho^+ \nabla \hat{N}^+_i \nabla \hat{N}^+_j + 2\frac{\partial \rho^+}{\partial \|v\|^2}(\nabla \hat{N}^+_j \Theta^+_i)(\nabla \hat{N}^+_i \Theta^+_i)\, \mathrm{d}\Gamma$$

$$B^{e,-}_{i,j} = \frac{\partial R^{e,-}_{i,B}}{\partial \Theta^-_j}(\Theta) = -\int_{\Gamma^-_{W,e}} \rho^- \nabla \hat{N}^-_i \nabla \hat{N}^-_j + 2\frac{\partial \rho^-}{\partial \|v\|^2}(\nabla \hat{N}^-_j \Theta^-_i)(\nabla \hat{N}^-_i \Theta^-_i)\, \mathrm{d}\Gamma$$

Since linear elements are considered, their gradients are constant and the boundary

integrals can be replaced by volume integrals.

$$B_{i,j}^{e,+} = \frac{\partial R_{i,B}^{e,+}}{\partial \Theta_j^+}(\Theta) = \int_{\Omega_{W,e}^+} \rho^+ \nabla \hat{N}_i^+ \nabla \hat{N}_j^+ + 2\frac{\partial \rho^+}{\partial \|v\|^2}(\nabla \hat{N}_j^+ \Theta_i^+)(\nabla \hat{N}_i^+ \Theta_i^+)\, \mathrm{d}\Omega \quad (2.25)$$

$$B_{i,j}^{e,-} = \frac{\partial R_{i,B}^{e,-}}{\partial \Theta_j^-}(\Theta) = -\int_{\Omega_{W,e}^-} \rho^- \nabla \hat{N}_i^- \nabla \hat{N}_j^- + 2\frac{\partial \rho^-}{\partial \|v\|^2}(\nabla \hat{N}_j^- \Theta_i^-)(\nabla \hat{N}_i^- \Theta_i^-)\, \mathrm{d}\Omega \quad (2.26)$$

The terms in the system of equations obtained in (2.25) and (2.26) are equivalent to those defined by the full-potential equation in (2.21), only adapting the degrees of freedom according to (2.22) and (2.23).

In order to model the uncertainty in the problem the approach used in Tosi, Amela et al. 2021 is considered. The Dirichlet condition on Γ_D is transformed to a stochastic condition by considering some variability in the free stream velocity $\boldsymbol{v}_\infty$, which can be expressed in terms of the Mach number and the angle of attack:

$$\boldsymbol{v}_\infty = M_\infty a_\infty \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

where the free stream sound velocity is set as $a_\infty = 340\,\mathrm{m\,s^{-1}}$. The Mach number and the angle of attack are therefore two independent random variables that model the stochastic problem. The probability distribution of the two stochastic variables are $M_\infty \sim \mathcal{N}(0.3, 0.1)$ and $\alpha \sim \mathcal{N}(5.0°, 0.02)$, respectively. $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution of mean μ and variance σ^2.

The main quantity of interest is the lift coefficient, which is computed using the Kutta–Joukowski theorem:

$$C_l = \frac{\Gamma}{\frac{1}{2} v_\infty c}$$

where Γ is the circulation of the flow, ρ_∞ is the free stream density and v_∞ the free stream velocity. The circulation Γ can be expressed as the potential jump across the domain $\Gamma = u^+ - u^-$, shown by Nishida and Drela 1995. This leads to the definition of the lift coefficient in terms of the potential jump across the wake:

$$C_l = \frac{2}{v_\infty c}(u^+ - u^-)$$

Two approaches are adopted to create the MLMC hierarchy: a fixed hierarchy of meshes and an adaptive refinement approach. The fixed hierarchy consists on a set of classical body-fitted meshes where the compressible potential flow over an airfoil is solved. Some details on the mesh used in the analysis is shown in table 1. The three first levels of the fixed hierarchy are shown in figure 7.

Level	n_{nodes}	n_{elem}	h_{min}
0	7238	14 230	10^{-2}
1	18 931	37 054	10^{-3}
2	35 016	68 364	10^{-4}
3	51 129	99 672	10^{-5}
4	68 471	133 430	10^{-6}

Table 1: Information of the fixed hierarchy of meshes. The number of nodes n_{nodes}, the number of elements n_{elem}, and the minimal size h_{min} is depicted.

The adaptive refinement approach starts from a pre-generated mesh that is considered the starting level in the MLMC algorithm. Metric-based refinement has been employed to automatically adapt the meshes from the starting mesh, using the method from C. Dapogny et al. 2014 implemented in MMG. Using this approach allows generating the subsequent levels on the fly. The metric is constructed from the Hessian of solution, as introduced by Frey and Alauzet 2005. The metric is of the form:

$$\mathcal{M} = \mathcal{R}\hat{\Lambda}^{\top}\mathcal{R} \qquad \text{where} \qquad \hat{\Lambda} = \text{diag}(\hat{\lambda}_i) \tag{2.27}$$
$$\hat{\lambda}_i = \min\left(\max\left(\frac{c_d|\lambda_i|}{\varepsilon}, h_{\text{max}}^{-2}\right), h_{\text{min}}^{-2}\right)$$

where λ_i are the eigenvalues and $\mathcal{R}$ the matrix of eigenvectors of the Hessian $\mathcal{H}_u$ of a given variable u. The eigenvalues are truncated for some user-defined minimal and maximal sizes, h_{min} and h_{max}. The metric depends on a constant c_d and the interpolation error ε. This error is defined as the error that is committed by the discretisation of the domain and the representation of u as u_h:

$$\varepsilon = \left\|u - u^h\right\|_{\text{L}^\infty}.$$

This Hessian-based metric was previously described in deliverable 5.3, § 2.2; further details are available there, such as the estimation of the Hessian. In order to generate the meshes on the fly, the analysis starts from a mesh of 70 000 elements with an already well-defined airfoil. The subsequent levels are generated for every realisation starting from the initial mesh, and each level enforces a decaying interpolation

error. For some level L, the interpolation error used to refine the mesh is computed according to:

$$\varepsilon_L = \varepsilon_0 2^{-L}$$

for some starting ε_0, which has been set to $\varepsilon_0 = 0.2$

As done for the Van der Pol oscillator in section 2.2.1, we are interested in computing the 70 %-CVaR for the above mentioned uncertainties. The considered geometry is a NACA0012 airfoil at an angle of attack $\alpha = 5°$. We use the strategy described in Section 2.1.2 for estimating the CVaR and adaptively calibrate the hierarchy using the approach described in Section 2.1.4 based on the error estimators in Section 2.1.3. We repeat the MLMC simulations multiple times for a given target tolerance using both mesh-adaptivity strategies.

The results can be summarised as follows. We begin first with the results for the fixed-hierarchy case. First, we show in figure 9 the bias and variance decay of the first derivative of $\hat{\Phi}_L$ estimated using the final hierarchy of one of the finest tolerances tested. As can be seen from the figure, the problem exhibits ideal properties for the application of MLMC methods. In addition, figure 10 shows the variation of the level-wise sample sizes for different iterations of the CMLMC algorithm for one run of the fixed-hierarchy MLMC simulations at the finest tolerance tested. As can be seen from the figure, the algorithm successfully improves the predicted hierarchy on a decreasing sequence of tolerances while maintaining the theoretically predicted rate of decay of the level-wise samples with respect to the levels.

Lastly, we present in figure11 the complexity behaviour for the airfoil. For each tolerance, we run the MLMC simulations 5 times. We plot the cost estimated for computing the optimal hierarchy for each simulation, as well as the mean cost over the 5 simulations. As can be seen from the figure, the least squares rate of the costs is better than the best-case theoretically expected value for MLMC simulations. It also has a significantly better rate than the theoretically expected rate for Monte Carlo simulations, thus demonstrating the improved performance of an optimally calibrated MLMC estimator over the Monte Carlo estimator.

We ran similar simulations for the adaptive hierarchy case. The decay of the bias and variance are not shown since the finest tolerance simulated triggered only 3 levels of refinement, leading to only 2 data points. However, the decay rates obtained are comparable to those obtained by the fixed hierarchy simulations. Figure 12 shows the results of the complexity study carried out for the adaptive hierarchy case. We find as well in this case that the complexity behaviour is better than the theoretically predicted values for the MLMC and Monte Carlo estimators.

An example of the distribution of the lift coefficient obtained for a run with 256 samples on the finest level of the adaptive refinement hierarchy is shown in figure 13 as an histogram. The lift coefficient values are set as negative values in

order to evaluate the upper tail of the distribution to compute both the VaR and the CVaR. The obtained values for the mean, VaR and CVaR are $\mathbb{E}(C_l) = 0.638$, $\text{VaR}_{0.7}(C_l) = 0.615$ and $\text{CVaR}_{0.7}(C_l) = 0.609$, respectively. It can be seen that the distribution is not symmetric, and that the distribution is not normal. These results could help to understand the effect of the uncertain inlet, especially on the less likely and less favourable scenarios. In this case, we are assessing with the CVaR the average of the all the scenarios where the airfoil would perform the worst.

Figure 14 shows the distribution of several estimated statistics of the pressure coefficient field over the airfoil, including the mean, the VaR and the CVaR for a significance value of $\tau = 0.7$. Other statistics introduced by Rockafellar and Royset 2015 such as the *worst-case* and *safety margin* statistics are also illustrated. This analysis was performed by post-processing the pressure coefficient gathered after solving each realisation in the MLMC analysis on the adaptive refinement strategy.

The line denoted in red in figure 14 shows the VaR distribution along the airfoil. Using a significance of $\tau = 0.7$, this line shows the region that leaves 70 % of the samples above the line and 30 % of the samples below. The CVaR line, illustrated in black, represents the expected value of the 30 % samples below the VaR. In other words, it gives the information of the behaviour of the samples that are less favourable and that yield lower values of the pressure coefficient. This can be compared to the other statistics shown in the plot, which do not share the same properties as the CVaR.

The distribution of the mean, depicted in blue, shows the average performance of the airfoil across all the realisations. The mean line is extended with a filling area, covering the mean values plus or minus three times the standard deviation for visualisation purposes. If the distribution of the pressure coefficient followed a normal distribution, the illustrated area would cover 99.7 % of the samples. Using the mean as risk measure does not give information of the behaviour of the tails of the distributions and the extreme scenarios.

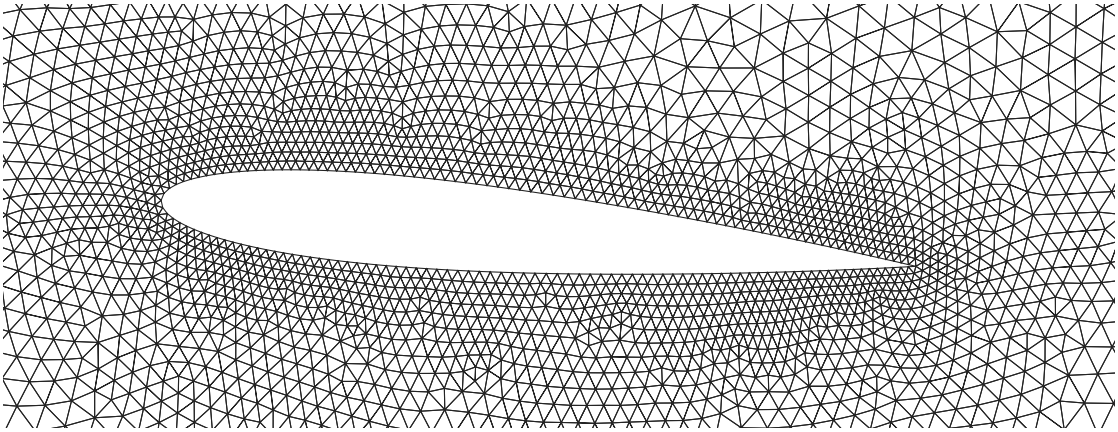
The green dashed line shows the worst-case values of the pressure coefficient, denoted as the *supremum* values of the pressure coefficient, $\sup C_p$. This line represents the actual edge values that the pressure coefficient took across the 256 samples. As pointed out by ibid., this risk measure does not give information on the distribution of the pressure coefficient values, and it might be too pessimistic. The orange dashed line denotes the so-called safety margin, for a significance of 70 % and calculated as $\mathbb{E} + \lambda_{0.7}\sqrt{\mathbb{V}\text{ar}}$. Here, $\lambda_{0.7} = 0.524$ is the percent point function of the normal distribution, defined as the inverse of the CDF. It represents the region that would split the distribution leaving 70 % of the samples above it, if the distribution was a normal. The fact that this line does not match the VaR line, shows that the distribution is not following a normal behaviour.

From an engineering point of view, the CVaR risk measure is a robust choice

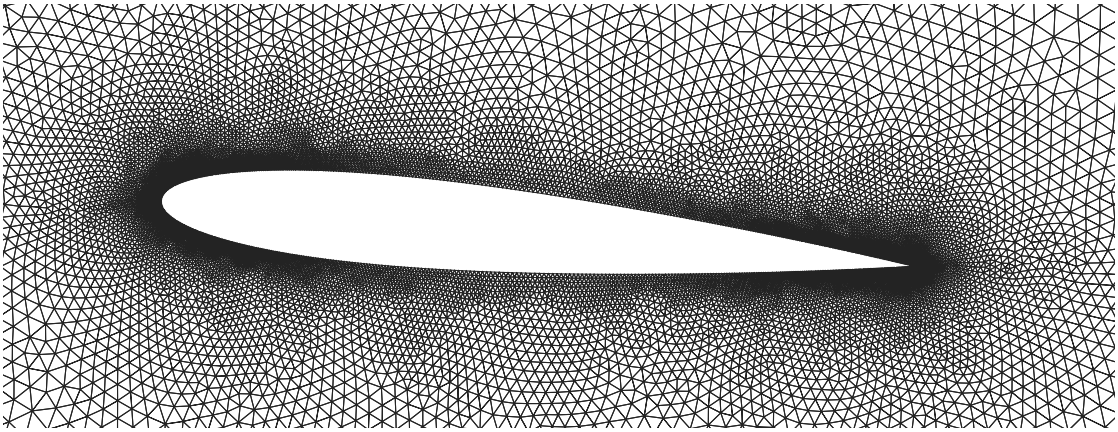
to assess the less favourable scenarios. It does not require the distribution to be normal, and it gives information of the less likely scenario, those appearing at the tails of the distribution.

The obtained results allow concluding that the MLMC method can be applied for the presented problem using the CVaR as risk measure for which good decay rates were obtained. Statistical information can be obtained from this analysis which can aid engineers in the decision-making at different design phases. Also, the obtained results enable the use of this method in the context of optimisation, which is applied in deliverable 6.5.

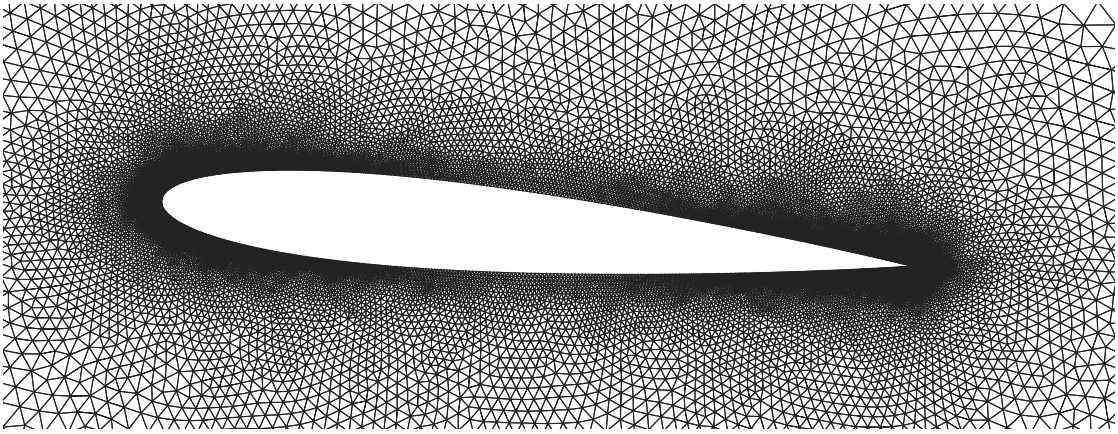
The data set of raw results of these numerical experiments is publically available as a part of Ayoul-Guilmard, Rosa M. Badia et al. 2021.



(a) Pre-generated level 0 mesh

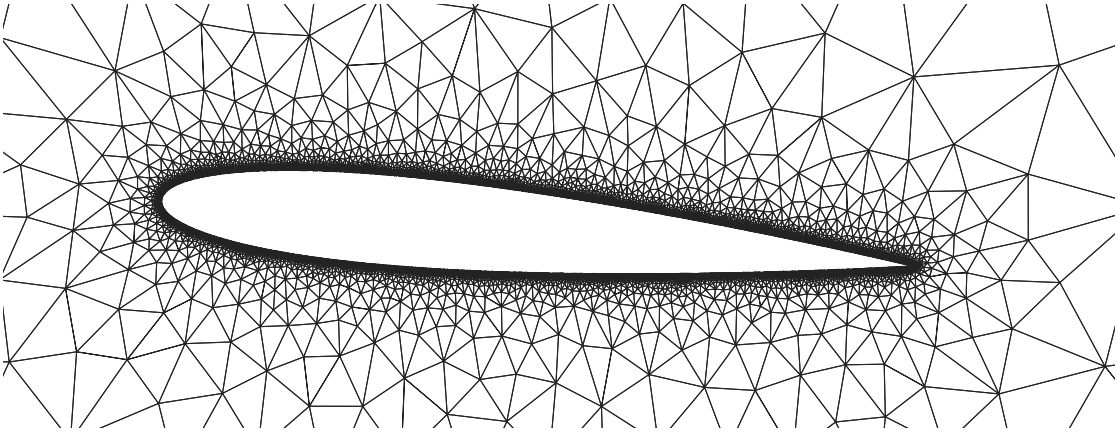


(b) Pre-generated level 1 mesh

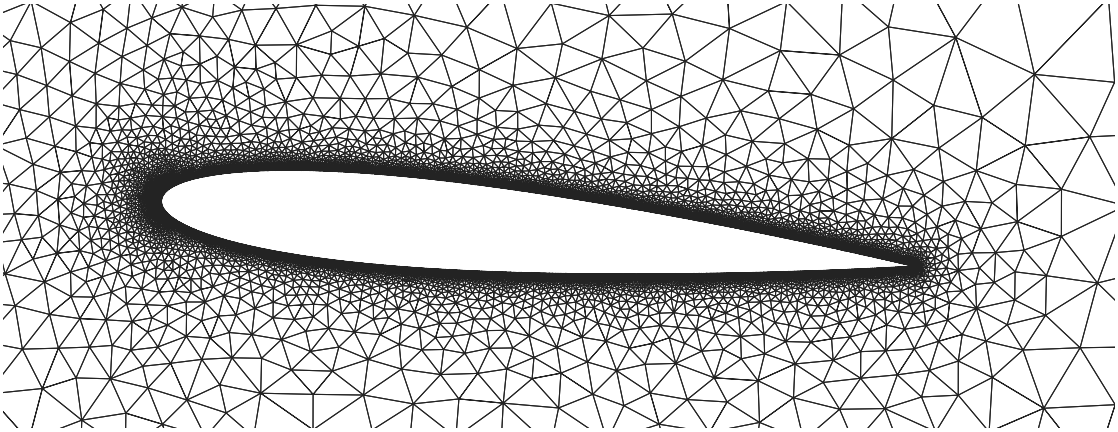


(c) Pre-generated level 2 mesh

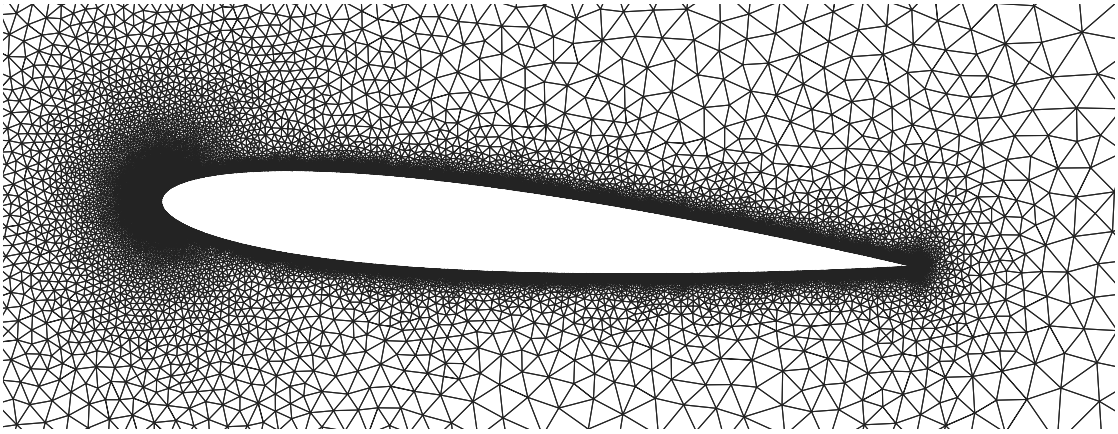
Figure 7: Different snapshots corresponding to the first, second and third level of the MLMC analysis using a fixed hierarchy of meshes. Further details can be found in table 1.



(a) Starting mesh of the adaptive algorithm. It is considered the level 0 mesh

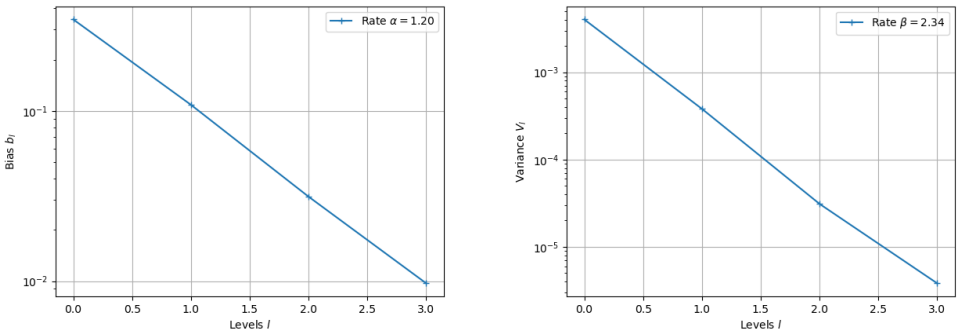


(b) Adaptively refined mesh corresponding to level 1



(c) Adaptively refined mesh corresponding to level 2

Figure 8: Different snapshots corresponding to the first, second and third level of the MLMC analysis using an adaptively refined hierarchy of meshes. For increasing levels, the interpolation error is halved according to equation 2.2.2.



(a) Bias decay (b) Variance decay

Figure 9: Convergence properties for the potential flow solver with a fixed hierarchy of meshes

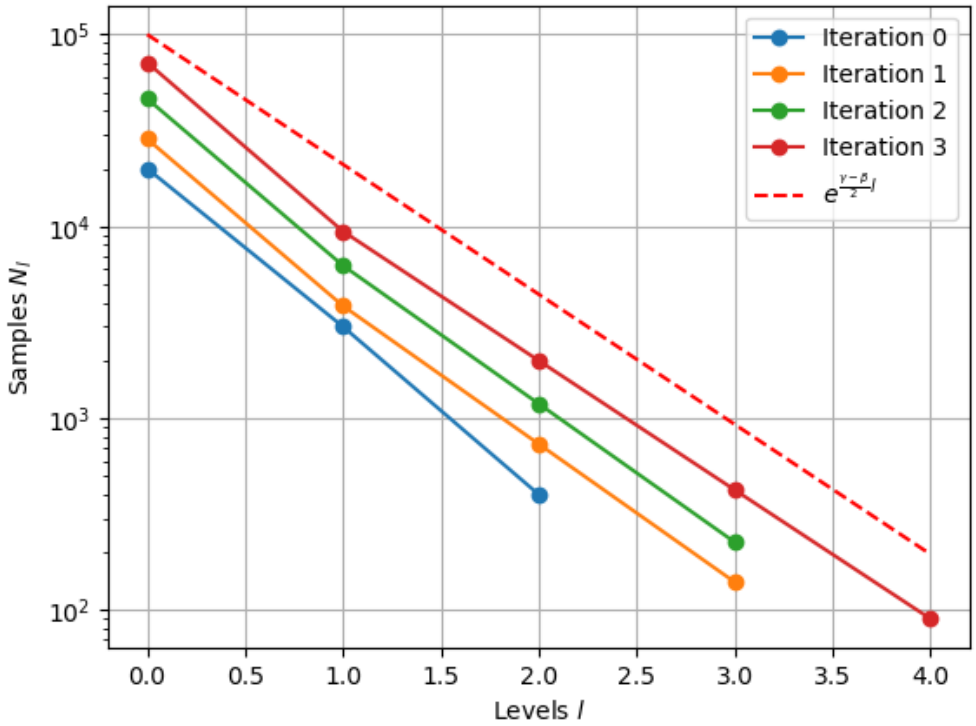


Figure 10: Level-wise sample sizes in the CMLMC algorithm for the fixed hierarchy of meshes in the potential flow application

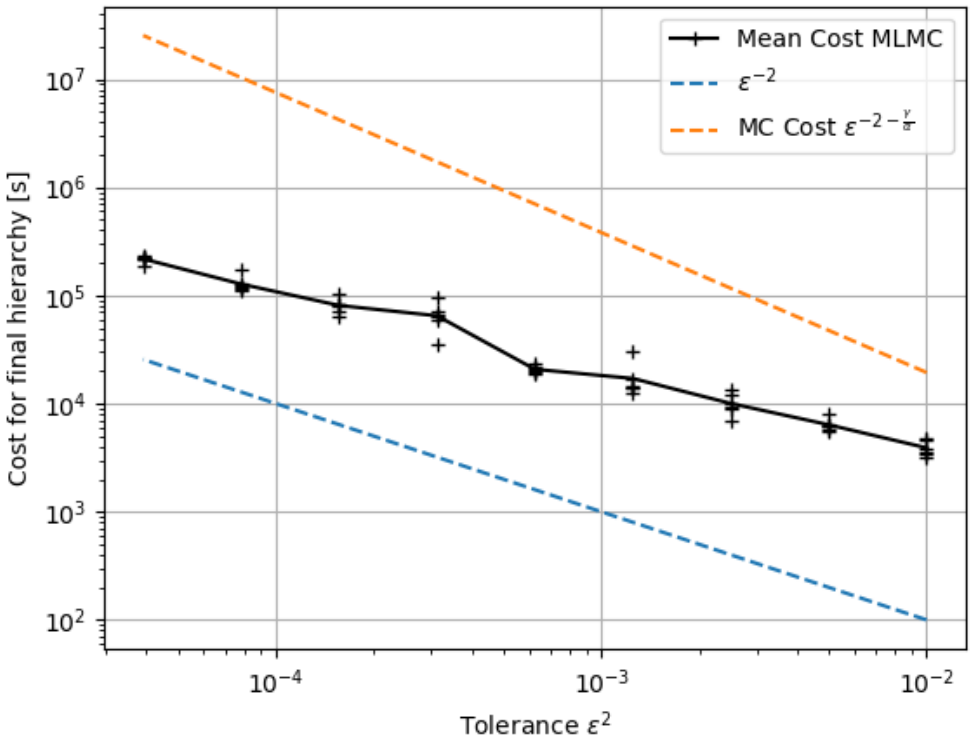


Figure 11: Complexity behaviour for the fixed hierarchy of meshes in the potential flow application

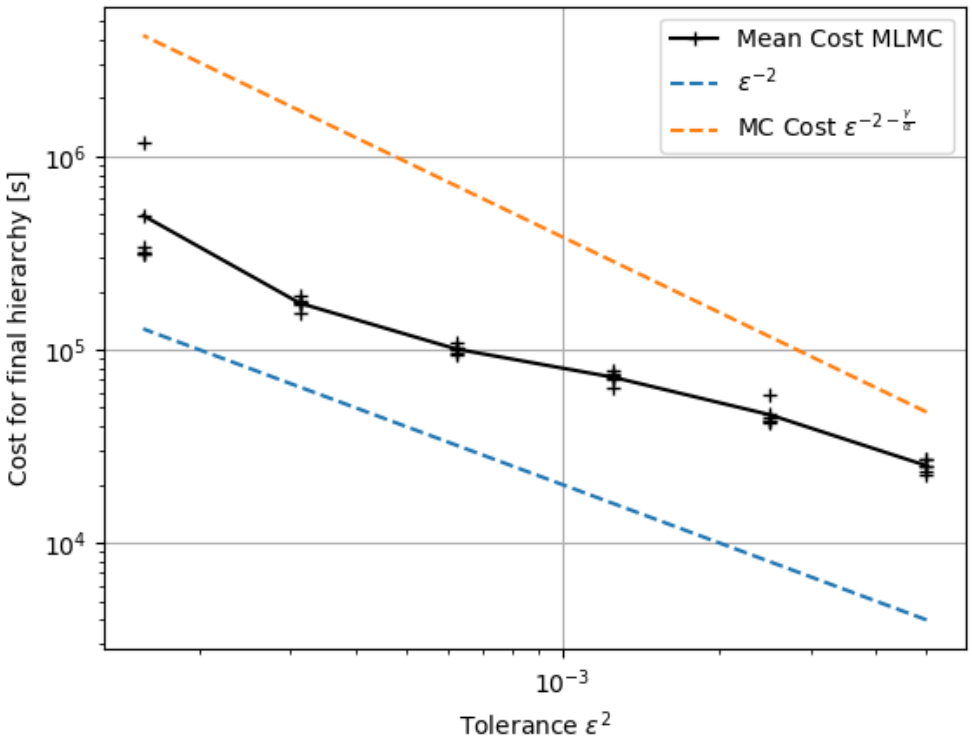


Figure 12: Complexity behaviour for the adaptively refined hierarchy of meshes in the potential flow application

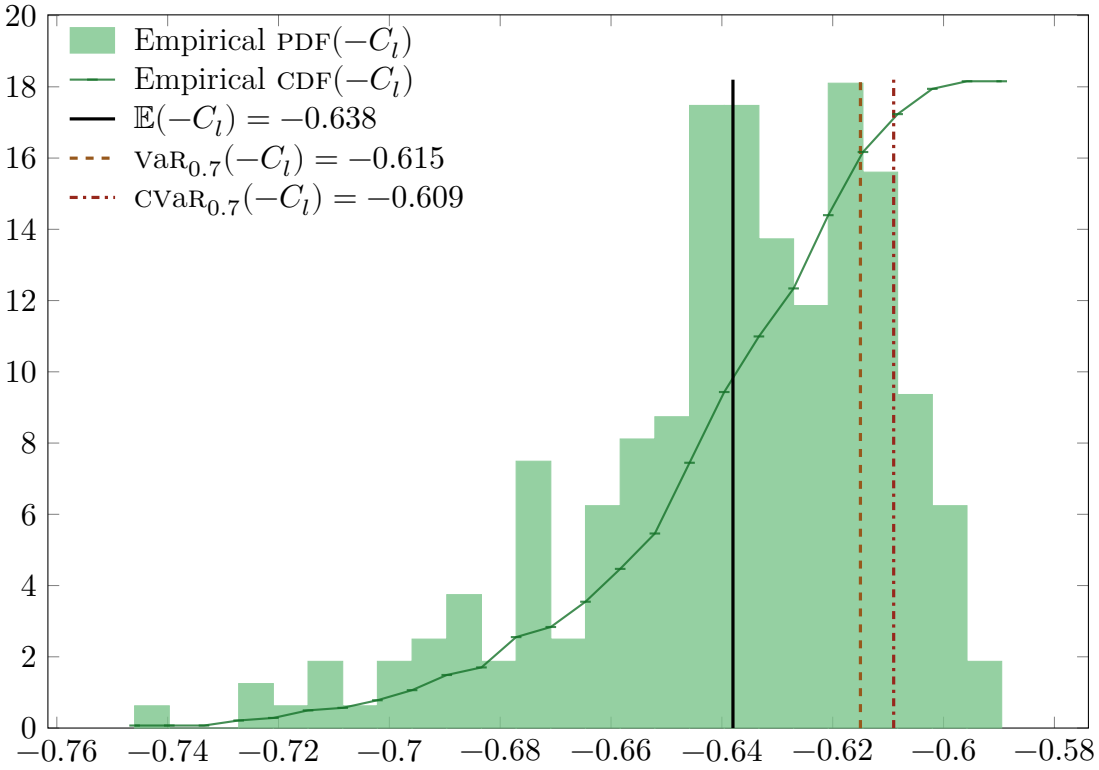
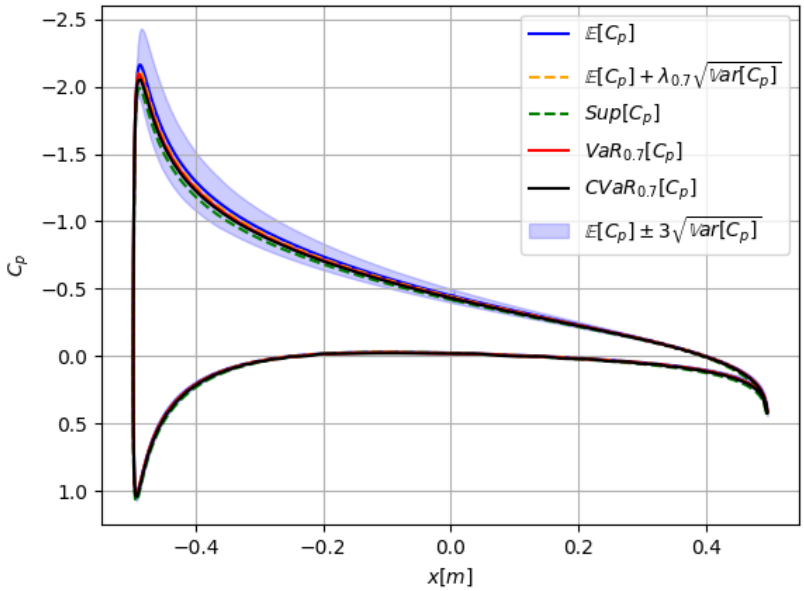
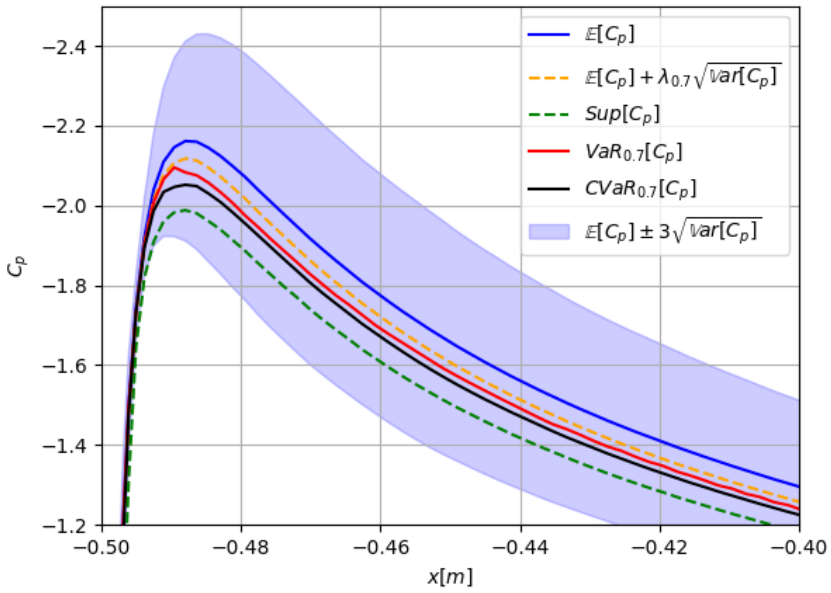


Figure 13: Distribution of the opposite of the lift coefficient



(a) Pressure distribution along the airfoil shape



(b) Zoom in at the trailing edge

Figure 14: Distribution of statistical quantities for the pressure field over a NACA0012 airfoil at 5° (significance value $\tau = 0.7$)

3 Multi-fidelity Monte Carlo method for turbulent fluid problem

The ExaQUte project initially planned to use MLMC methods to accelerate the estimation of statistics of quantities of interest computed from random fluid flow simulations. Although this proved efficient in cases exhibiting a low Reynolds number, deliverable 5.3, § 3 reported poor performance in the case of turbulent flows. An extensive study in deliverable 5.4, § 2 evinced clearly that assumption 1 was not satisfied in the turbulent case for our quantities of interest and with our adaptive remeshing strategy. Several ideas were proposed to accelerate uncertainty quantification in the turbulent case.

A promising idea was the use of multi-fidelity Monte Carlo (MFMC) methods instead of multi-level Monte Carlo (MLMC) methods. The former category is broader, with more varied – and often weaker – requirements on the numerical models used. Instead of assumption 1, their efficiency typically relies on the correlation between the numerical models. While the quantities of interest considered showed no convergence with mesh refinement, this does not preclude that approximations at different levels are strongly correlated. However, correlation is difficult to predict and this motivated a feasibility study in ibid., § 5.1. Its results encouraged us to pursue this approach.

An additional argument in favour of a MFMC approach is that it is similar enough to a MLMC approach that it can benefit from the on-going work already put in motion for the latter. For example the work on adaptive remeshing and parallel computing benefits both equally. Even the research and software development begun – and still active – for MLMC methods can be at least partially applied or adapted for MFMC estimations.

Unlike in section 2, this section of the report will address the estimation of the expectation of the QoI. We will first propose a suitable MFMC estimator, discuss possible strategies for its calibration and propose an adaptive algorithm. Then, we will demonstrate its application to a turbulent fluid problem pertinent to wind engineering.

3.1 Multi-fidelity Monte Carlo method

Let $(X_l)_{l\in[\![0,L]\!]} =: \boldsymbol{X}$ be random quantities of interest associated with a list of numerical models. X_L is the 'high-fidelity' model while the others are 'low-fidelity' models; we wish to estimate $\mathbb{E}(X_L)$. Low-fidelity models should be less expensive yet correlated to the high-fidelity model; they are typically less descriptive models, reduced-order models, or – as will be the case in this report – similar numerical models with a coarser discretisation.

The field of multi-fidelity methods is much more diverse than that of MLMC, thus the choice of a specific estimator is not trivial. We will first recall the gist of our previous study on MFMC for purposes of perspective and comparison, before presenting the MFMC method and finally the adaptive algorithm to be used here.

3.1.1 Previous study

As an alternative to MLMC for turbulent flows, a MFMC method was first proposed in deliverable 5.4, § 5.1 for the purpose of a performance study. This estimator was defined as

$$\frac{1}{m_L}\sum_{i=1}^{m_L}\left(X_L^{(i)} - \sum_{l=0}^{L-1}\alpha_l X_l^{(i)}\right) + \sum_{k=0}^{L-1}\frac{\alpha_k}{m_k}\sum_{j=1}^{m_k}X_k^{(j)},$$

with control coefficients $\boldsymbol{\alpha} \coloneqq (\alpha_i)_{i=0}^{L-1} \in \mathbb{R}^L$ and sample sizes $\boldsymbol{m} \coloneqq (m_i)_{i=0}^{L} \in \mathbb{N}^{L+1}$ chosen such that $\forall l \in [\![0, L-1]\!]$, $m_l > m_L > 0$. A first item of note is the lack of hierarchy: no assumption has been made regarding the comparative accuracy of the low-fidelity models. A second noteworthy point is the sub-sampling: for every pair $\{l, k\} \subset [\![0, L]\!]$ and for any $i \in \mathbb{N}$, $X_l^{(i)}$ and $X_k^{(i)}$ use the same random event. The advantage of this latter point is that the cost of a given accuracy may be lower than if different random events – and therefore different solutions of the PDE – were required. The downside is that it makes the expression of the variance less straightforward, which in turns leads to more intricate expressions for the error estimation and the samples sizes optimised therefrom. For these, we refer the reader to ibid., eq. (56)–(57) and appendix B.

The efficiency of this estimator was predicted from an existing set of solutions. The low-fidelity models consisted of the same fluid problem with coarser meshes (see ibid., table 2). Based on the variances, covariances and computational costs of this set of models, the variance reduction factor compared to a MC estimator for the same budget was predicted to be in $[0.65, 0.75]$, depending on the models selected (see ibid., tables 3–4).

3.1.2 Hierarchical multi-fidelity estimator

Here we discuss a MFMC estimator different from the one used for the feasibility study recalled in 3.1.1 from ibid., § 5.1. First, we expect that the selected low-fidelity models follow a hierarchy of accuracy and cost and consider there a MFMC estimator with a structure closer to that of a MLMC estimator. Second, we will not use sub-sampling, in order to have simpler constructions for the error estimation and calibration. This choice also yields a method closer to our existing framework, and thus more likely to benefit from it as well as easier to implement within XMC.

Let $\boldsymbol{\alpha} \in \mathbb{R}^L$ and let us write the following telescopic sum:

$$\begin{aligned}\mathbb{E}(X_L) &= \mathbb{E}((X_L - \alpha_{L-1}X_{L-1}) + \alpha_{L-1}(X_{L-1} - \alpha_{L-2}X_{L-2}) + \cdots \\ &\qquad + \alpha_{L-1} \times \alpha_{L-2} \times \cdots \times \alpha_0 \times X_0) \\ &= \sum_{l=1}^{L} \mathbb{E}(X_l - \alpha_{l-1}X_{l-1}) \prod_{k=l}^{L-1} \alpha_k + \mathbb{E}(X_0) \prod_{k=0}^{L-1} \alpha_k. \end{aligned} \tag{3.1}$$

To write (3.1) more concisely, we use the convention $\prod_{x\in\emptyset} x = 1$. We also note $\forall l \in [\![1, L]\!]$, $Y_{l,\alpha_{l-1}} := X_l - \alpha_{l-1}X_{l-1}$ and $Y_{0,\alpha_{-1}} := X_0$. Then (3.1) becomes

$$\mathbb{E}(X_L) = \sum_{l=0}^{L} \left(\mathbb{E}\Big(Y_{l,\alpha_{l-1}}\Big) \prod_{k=l}^{L-1} \alpha_k \right). \tag{3.2}$$

For any list of sample sizes $\boldsymbol{m} \in \mathbb{N}_*^{L+1}$ we define our MFMC estimator $\mu_{\boldsymbol{m},\boldsymbol{\alpha}}$ as

$$\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X}) := \sum_{l=0}^{L} \mu_{m_l}(Y_{l,\alpha_{l-1}}) \prod_{k=l}^{L-1} \alpha_k, \tag{3.3}$$

using the notation introduced in (1.1). Whenever $\boldsymbol{\alpha}$ is independent from the random variables $(Y_{l,\alpha_{l-1}})_{l=0}^{L}$ in (3.3) (e.g. set a priori), $\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})$ is an unbiased estimator of $\mathbb{E}(X_L)$. Every random variable $Y_{l,\alpha_{l-1}}$ in (3.3) is sampled independently from the others; however, every sample of $Y_{l,\alpha_{l-1}}$ is constructed from samples of X_l and X_{l-1} generated for the same random event. Consequently, the variance of this estimator is

$$\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) = \sum_{l=0}^{L} \frac{v_l}{m_l}$$

with $\forall l \in [\![0, L]\!]$

$$v_l := \mathbb{V}\mathrm{ar}\Big(Y_{l,\alpha_{l-1}}\Big) \prod_{k=l}^{L-1} \alpha_k^2$$

and

$$\mathbb{V}\mathrm{ar}\Big(Y_{l,\alpha_{l-1}}\Big) = \mathbb{V}\mathrm{ar}(X_l) + \alpha_{l-1}^2\, \mathbb{V}\mathrm{ar}(X_{l-1}) - 2\alpha_{l-1}\mathbb{C}\mathrm{ov}(X_l, X_{l-1}).$$

We define the statistical error of this estimator as $\sqrt{\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big)}$ and its cost as

$$€\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) := \sum_{l=0}^{L} m_l\, €\Big(Y_{l,\alpha_{l-1}}\Big) = m_L\, €(X_L) + \sum_{l=0}^{L-1} €(X_l)(m_l + m_{l+1}).$$

We will present a first, simpler approach to choose the sample sizes and control coefficients so as to satisfy either a given budget or a given tolerance, then propose one way to improve it.

Separate calibration In this first strategy we take the simplified approach of optimising separately the control coefficients and the sample sizes. To this effect, let us optimise first control coefficient under the assumptions that (i) the expectation of the lowest-fidelity model $\mathbb{E}(X_0)$ is known; (ii) the sample sizes are the same across all levels: $\forall l \in [\![0, L]\!]$, $m_l = m_0$. Then the variance of the MFMC estimator simplifies to

$$\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) = \frac{1}{m_0}\,\mathbb{V}\mathrm{ar}\left(\sum_{l=1}^{L} Y_{l,\alpha_{l-1}} \prod_{k=l}^{L-1} \alpha_k\right).$$

Let us choose the optimal control coefficients $\check{\boldsymbol{\alpha}} \in \mathbb{R}^L$ so as to minimise this variance:

$$\check{\boldsymbol{\alpha}} := \mathrm{argmin}\left\{\sum_{l=1}^{L} \mathbb{V}\mathrm{ar}\big(Y_{l,a_{l-1}}\big) \prod_{k=l}^{L-1} a_k^2 : \boldsymbol{a} \in \mathbb{R}^L\right\}.$$

Then, $\forall l \in [\![0, L-1]\!]$,

$$\check{\alpha}_l = \frac{\mathbb{C}\mathrm{ov}(X_{l+1}, X_l)}{\mathbb{V}\mathrm{ar}(X_l) + \sum_{i=1}^{l} \mathbb{V}\mathrm{ar}\big(Y_{i,\check{\alpha}_{i-1}}\big) \prod_{j=0}^{i-1} \check{\alpha}_j}, \tag{3.4}$$

with the conventions $\sum_{x\in\emptyset} x = 0$ and $\prod_{x\in\emptyset} x = 1$. The detailed reasoning and proof of (3.4) is given in appendix A.1.1.

In a second stage, we now withdraw the two assumptions (i) and (ii) and seek the sample sizes so as to satisfy a given tolerance $\eta \in]0, +\infty[$ on the statistical error for a minimal cost. We define these optimal sample sizes as

$$\check{\boldsymbol{m}} := \mathrm{argmin}\big\{\mathbb{C}\big(\mu_{\boldsymbol{n},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big) : \mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{n},\check{\alpha}}(\boldsymbol{X})\big) \leqslant \eta^2; \boldsymbol{n} \in \mathbb{N}_*^L\big\} \tag{3.5}$$

Then, $\forall l \in [\![0, L]\!]$

$$\check{m}_l = \left\lceil \eta^{-2} \sqrt{\frac{\mathbb{V}\mathrm{ar}\big(Y_{l,\check{\alpha}_{l-1}}\big)}{\mathbb{C}\big(Y_{l,\check{\alpha}_{l-1}}\big)}} \sum_{i=0}^{L} \sqrt{\mathbb{V}\mathrm{ar}\big(Y_{i,\check{\alpha}_{i-1}}\big)\,\mathbb{C}\big(Y_{i,\check{\alpha}_{i-1}}\big)} \right\rceil. \tag{3.6}$$

The proof of (3.6) is given in appendix A.1.2. The calibrated estimator is therefore $\mu_{\check{\boldsymbol{m}},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})$. This approach to calibrate the estimator $\mu_{\boldsymbol{m},\boldsymbol{\alpha}}$ is simple in the sense that it optimises separately the control coefficients and the sample sizes. However, it does not necessarily yield the lowest variance, since the control coefficients are sub-optimal.

Remark 1 (Fixed budget). We discussed exclusively the calibration of the MFMC estimator to satisfy an upper bound on its statistical error at a minimal cost. It is equivalent to tune the estimator by a minimisation of its statistical error with a computational budget (i.e. an upper bound on the cost) as a constraint. Either constrained minimisation problem yields the same distribution of sample sizes; i.e. $\{m_l/m_0 : l \in [\![0, L]\!]\}$ is independent of the tolerance or the budget. The only difference is that a budget-constrained estimator would round down ($\lfloor\cdot\rfloor$) instead of up ($\lceil\cdot\rceil$) in formula (3.6); see remark 2 in appendix for details. ◊

COUPLED CALIBRATION Compared to the previous calibration strategy, better performance (e.g. lower cost for the same statistical error) can be achieved by tuning concurrently both the sample sizes and control coefficients so as to minimise the cost of the MFMC estimator. Let $\eta \in]0, +\infty[$ be the tolerance on the statistical error. We define the optimised sample sizes and control coefficients as

$$[c](\breve{\boldsymbol{m}}, \breve{\boldsymbol{\alpha}}) := \operatorname{argmin}\Big\{ €\big(\mu_{\boldsymbol{n},\boldsymbol{a}}(\boldsymbol{X})\big) : \boldsymbol{n} \in \mathbb{N}_*^{L+1};\ \boldsymbol{a} \in \mathbb{R}_*^{L}; \\ \mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{n},\boldsymbol{a}}(\boldsymbol{X})\big) \leqslant \eta^2 \Big\}. \quad (3.7)$$

This problem is more complicated to solve than (3.5) and therefore we give here the solution for the specific case that will be used in section 3.2.5, i.e. $L := 1$.

$$\breve{m}_0 = \left\lceil \frac{\mathbb{V}\mathrm{ar}(X_1)}{\eta^2 r}\left(1 - \frac{\rho^2}{r+1}\right)\right\rceil ; \qquad \breve{m}_1 = \left\lceil \frac{\mathbb{V}\mathrm{ar}(X_1)}{\eta^2}\left(1 - \frac{\rho^2}{r+1}\right)\right\rceil ; \quad (3.8)$$

$$\breve{\alpha}_0 = \frac{\mathbb{C}\mathrm{ov}(X_0, X_1)}{\mathbb{V}\mathrm{ar}(X_0)\left(1 + \frac{\breve{m}_1}{\breve{m}_0}\right)} ; \quad (3.9)$$

with

$$r := \frac{1 - \rho^2 + 2\rho\sqrt{(1-\rho^2)\left(\frac{€(X_0)+€(X_1)}{€(X_0)} - 1\right)}}{2\left(\rho^2 \frac{€(X_0)+€(X_1)}{€(X_0)} - 1\right)}, \qquad \rho := \frac{|\mathbb{C}\mathrm{ov}(X_0, X_1)|}{\sqrt{\mathbb{V}\mathrm{ar}(X_0)\,\mathbb{V}\mathrm{ar}(X_1)}}.$$

Unlike in the previous approach, the control coefficient α_0 now depends on the ratio of sample sizes $\breve{m}_1/\breve{m}_0 = r$. The detailed proof of this result is given in appendix A.2.

An interesting point of this proof is the requirement of the following assumption.

ASSUMPTION 2 (Validity of single low-fidelity model).

$$\frac{\mathbb{C}\mathrm{ov}(X_0, X_1)^2}{\mathbb{V}\mathrm{ar}(X_0)\,\mathbb{V}\mathrm{ar}(X_1)} > \frac{€(X_0)}{€(X_0) + €(X_1)}.$$

◊

We interpret the inequality above as a *necessary* condition for the low-fidelity model X_0 to be beneficial[3] to the MFMC estimation. If assumption 2 does not hold, (3.7) has no solution that yields a lower cost than a MC estimation at the same accuracy. However, assumption 2 is not sufficient to guarantee that the MFMC estimator will perform better than a MC estimator: it is sufficient only under the approximation $\mathbb{C}(X_0) + \mathbb{C}(X_1) \approx \mathbb{C}(X_1)$; see appendix A.2 for details. These considerations will be illustrated in section 3.2.4.

3.1.3 Adaptive MFMC algorithm

We propose algorithm 2 to implement the calibration strategy discussed above. One feature of interest is the difference between η and $\tilde{\eta}$. Since there is typically little information available a priori about the distribution of the random variables $\{X_l : l \in [\![0, L]\!]\}$, the initial choice of sample sizes is at best an informed guess. Consequently, the initial estimations of variances and covariances are unreliable. In a favourable case, these values would be underestimated and the algorithm would progressively increase the sample sizes until the desired accuracy η is reached. However, if they are overestimated then the sample sizes required would be overestimated as well and the eventual cost would be excessive, possibly by a vast margin. The strategy we propose here to address this issue is to calibrate the sample sizes for a tolerance $\tilde{\eta}$ higher than the final target η. This 'calibration tolerance' $\tilde{\eta}$ is progressively reduced across iterations until it equals η; by that time, the sample estimations are more accurate. Any decreasing sequence of values converging to η is acceptable, and we propose in algorithm 2 a geometric sequence. A similar strategy was used in algorithm 1, section 2.1.5. For more insights into this 'continuation' strategy, and an approach to blend sample estimations and model predictions, we refer the reader to Collier et al. 2015.

For any pair $(i, j) \in [\![0, L]\!]^2$ the covariance is estimated from $n \in \mathbb{N}_*$ events $(\omega_k)_{k=1}^n \in \Omega^n$ with the unbiased estimator

$$\begin{aligned}\mathbb{C}\mathrm{ov}\big(X_i, X_j\big) \approx \frac{1}{n-1} \sum_{k=1}^{n} \left(X_i(\omega_k) - \frac{1}{n} \sum_{l=1}^{n} X_i(\omega_l) \right) \\ \times \mathbb{C}\mathrm{ov}\big(X_i, X_j\big) \approx \frac{1}{n-1} \sum_{k=1}^{n} \left(X_j(\omega_k) - \frac{1}{n} \sum_{m=1}^{n} X_j(\omega_m) \right). \quad (3.10)\end{aligned}$$

As annotated, the computations in line 7 are expected to dominate the cost. This is where parallel computing is leveraged first and foremost.

[3] A low-fidelity model is not beneficial if its cost outweighs the information it contributes to the estimation (i.e. the variance reduction). Then, the MFMC estimator performs worse than a direct MC estimation on the high-fidelity model.

Algorithm 2: Adaptive multi-fidelity Monte Carlo algorithm

1 **Input:** $\boldsymbol{X} \in \mathrm{L}^2(\Omega, \mathbb{R})^{L+1}$; $\boldsymbol{m} \in \mathbb{N}_*^{L+1}$; $\eta > 0$; $\tilde{\eta} > \eta$; $a \in]0, 1[$; $i_{\max} > 0$
2 **Initialise:** $i = 0$, $\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) := \eta^2 + 1$
3 **While** $\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) > \eta^2$ **and** $i < i_{\max}$ **do**
4 **For** $l \in [\![0, L]\!]$ **do**
5 **For** $s \in [\![1, m_l]\!]$ **do**
6 Draw independently $\omega \in \Omega$
7 Compute sample $(X_{l-1}(\omega), X_l(\omega))$ `// expensive, parallel`
8 Estimate $\mathbb{V}\mathrm{ar}(X_{l-1})$, $\mathbb{V}\mathrm{ar}(X_l)$, $\mathbb{C}\mathrm{ov}(X_{l-1}, X_l)$
9 Tune the control coefficients $\boldsymbol{\alpha} \leftarrow$ (3.9)
10 Estimate $\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big)$
11 Update adaptivity target $\tilde{\eta} := \max\{\eta, a\tilde{\eta}\}$
12 Tune the sample sizes $\boldsymbol{m} \leftarrow$ (3.8) for $\tilde{\eta}$
13 $i := i + 1$
14 **Return** $\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})$

Algorithm 2 uses the coupled calibration strategy introduced last in section 3.1.2. To use instead the separate calibration strategy instead, it is enough to replace (3.9) with (3.4) in line 9 and (3.8) with (3.6) in line 12.

3.2 Application to a turbulent fluid problem

3.2.1 Problem of twisted building under uncertain wind

The demonstrator problem considered is the wind flow past a modern building, presented in figure 15. Its horizontal section is an ellipse, rotated and shrunk as the height increases, hence the name 'twisted building'. Such a building has been featured in other works of the ExaQUte project, e.g. deliverable 7.2; deliverable 7.4; and the numerical wind-tunnel experiment of Keith et al. 2021. Nevertheless, this is a significant progression from the previous applications presented in this work package. Indeed, deliverable 5.3, § 2.2 and deliverable 5.4, § 4 used a simplified bi-dimensional problem, while announcing the application that is being described here. See (deliverable 5.3, fig. 6) for a hierarchy of benchmarks of the ExaQUte project.

Let us note D the spatial domain; the temporal one is $[0\,\mathrm{s}, 600\,\mathrm{s}]$. The spatial boundary is partitioned as $\partial D =: D_{\text{inlet}} \cup D_{\text{ground}} \cup B \cup D_{\text{walls}} \cup D_{\text{ceiling}} \cup D_{\text{outlet}}$ where B is the building, the inlet is the smallest rectangle closest to the building, and the others are self-explanatory. The centre of the base of the building is 382.5 m

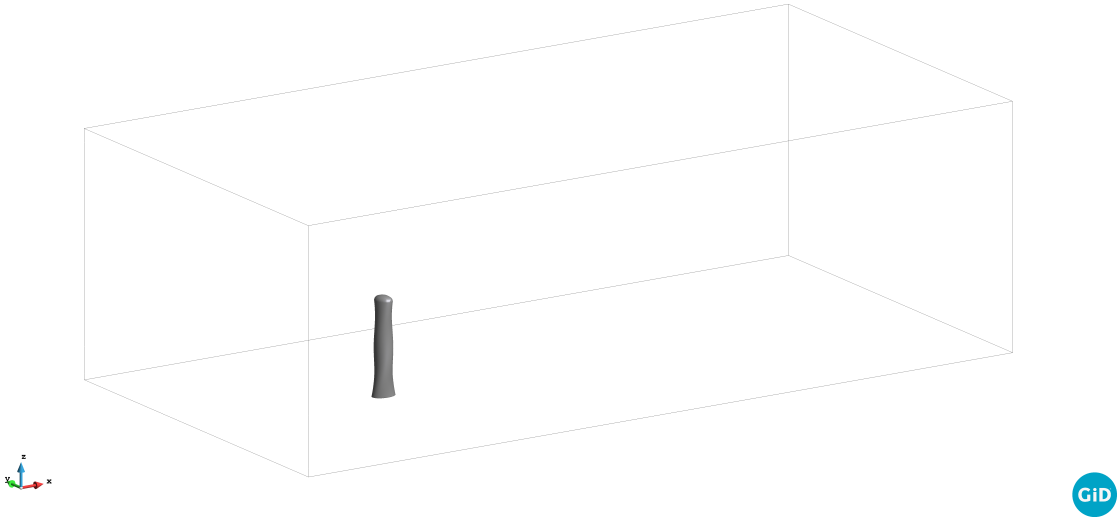


Figure 15: Simulation domain of the modern twisted building (1440 m × 720 m × 450 m; see table 2 for building dimensions)

from the inlet; the other dimensions are in table 2. The wind flow is modelled with the incompressible Navier–Stokes equations, which read

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \frac{\mu}{\rho} \Delta \boldsymbol{u} + \nabla p = \boldsymbol{g} \qquad \text{in } [0, 600] \times D, \tag{3.11a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } [0, 600] \times D, \tag{3.11b}$$

with the boundary conditions

$$\boldsymbol{u} = \boldsymbol{u}_{\text{inlet}} \qquad \text{in } [0, 600] \times D_{\text{inlet}}, \tag{3.11c}$$

$$p = 0 \qquad \text{in } [0, 600] \times D_{\text{outlet}}, \tag{3.11d}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{0} \qquad \text{in } [0, 600] \times \left(D_{\text{walls}} \cup D_{\text{ceiling}}\right), \tag{3.11e}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{in } [0, 600] \times \left(D_{\text{ground}} \cup D_{\text{building}}\right), \tag{3.11f}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\text{init}} \qquad \text{in } \{0\} \times D. \tag{3.11g}$$

The vector of gravity acceleration is noted $\boldsymbol{g} = -9.81\,\text{m}\,\text{s}^{-2}\boldsymbol{e}_3$ and $\boldsymbol{n}$ is the vector normal to the surface at the point considered. All relevant values are available in table 2, barring $\boldsymbol{u}_{\text{init}}$ and $\boldsymbol{u}_{\text{inlet}}$ which are discussed below.

Boundary conditions (3.11e) and (3.11f) are respectively called 'slip'[4] and 'no-slip' conditions (see Pope 2000, eq. (2.36)–(2.37) p. 17)). Condition (3.11f) is actually changed to a slip boundary condition on the ground close to the outlet in order to avoid an inflow of vortices from that outlet. See e.g. Dong et al. 2014

[4] Applied with the method described in Felippa 2004, chapter 8.

Table 2: Fluid and building properties

Quantity	Notation	Value	Unit
Reference height	H	80	m
Reference velocity	$\bar{u}_H$	40	$\mathrm{m\,s^{-1}}$
Roughness height	z_0	0.4 ± 0.3	m
Air density	ρ	1.225	$\mathrm{kg\,m^{-3}}$
Air dynamic viscosity	μ	$1.846 \cdot 10^{-5}$	$\mathrm{kg\,m^{-1}\,s^{-1}}$
Height		180	m
Major base diameter		40	m
Minor base diameter		32	m
Characteristic length		40	m
Reynolds number	Re	$1.062 \cdot 10^8$	

for considerations about domain truncation for simulations with a high Reynolds number.

The mean profile of the wind velocity $\boldsymbol{u}_{\text{inlet}}$ at the inlet is the sum of a logarithmic mean profile and fluctuations, which are modelled as in Mann 1998. The reference velocity $\bar{u}_H$ of the mean profile is the wind velocity at the reference height $80\,\mathrm{m} =: H$. In this study it is not stochastic but fixed to $40\,\mathrm{m\,s^{-1}}$, following standard practice in e.g. the standard tall building model of the Commonwealth advisory aeronautical council (CAARC) from Braun and Awruch 2009; also used in deliverable 7.2; deliverable 7.3; deliverable 1.4. A stochastic model for the reference velocity based of observed data is proposed in deliverable 7.4. The roughness height z_0 follows a uniform random distribution: $z_0 \in \mathcal{U}(0.1\,\mathrm{m}, 0.7\,\mathrm{m})$. Such a roughness height is typical of sparsely built urban areas according to Joint committee on structural safety 2001. In other words, the wind inlet is the sum of a stochastic, time-independent mean profile and of realistic, time-dependent fluctuations. The modelling and generation of this random wind inflow are explained in details in deliverable 7.3, § 2.

The initial condition $\boldsymbol{u}_{\text{init}}$ is a spatially-correlated and divergence-free field built on top of a time-averaged velocity field $\langle \boldsymbol{u}_0 \rangle$; it is not random. We refer to deliverable 3.3, § 3 for details about the generation of initial conditions for simulation of fluid dynamics. Using such initial conditions reduces the transient time of the fluid flow, which is the simulation time that must be discarded to avoid the bias of initial conditions[5]. This topic will be discussed further in section 3.2.4.

The problem is discretised using linear elements for both pressure and velocity fields. The variational multiscale method (Codina et al. 2017) is used to stabilise

[5] Also called 'burn-in time' or 'initial time'.

the problem and to solve the large-eddy simulation of the turbulent flow. The time-stepping uses a second-order fractional step method, which treats both pressure and velocity implicitly.

This study will discuss two time-dependent quantities of interest previously mentioned in deliverable 6.4, § 5.2.1: the force of the wind drag on the building

$$\boldsymbol{F}(t) \coloneqq \int_B p(t,\cdot)\boldsymbol{n}\,\mathrm{d}S$$

and the mechanical moment it creates at the centre O of the base of the building

$$\boldsymbol{M}(t) \coloneqq \int_{\boldsymbol{x}\in B} (\boldsymbol{x}-O)\times p(t,\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}S(\boldsymbol{x}).$$

More specifically, the study will focus on the time-averages of their major component[6]:$\langle \boldsymbol{F}\cdot\boldsymbol{e}_1\rangle_{t_0,600}$ and $\langle \boldsymbol{M}\cdot\boldsymbol{e}_2\rangle_{t_0,600}$ – the others components being negligible. The value of the transient time t_0 will be discussed in section 3.2.4.

3.2.2 Domain discretisation

To solve the stochastic computational fluid dynamics problem described above using the MFMC method, multiple levels of discretisation are generated to provide low-fidelity models. Meshes are refined adaptively to have a finer resolution where needed, and a coarser discretisation elsewhere. This is particularly important in the case of high Reynolds number flows, since turbulent phenomena are characterised by multiple spatial and temporal scales (see Pope 2000).

The mesh discretisations are refined with MMG and its ParMMG extension, presented in the introduction. This library employs a metric-based adaptive refinement strategy for the mesh, previously described in deliverable 5.3, § 2.2.2. An initial mesh of 3 783 908 elements in multiple uniform layers is generated with the GiD software from Melendo et al. 2021. The flow problem (3.11) is solved on this initial mesh for a duration of 400 s. Then the Hessian-based metrics of the velocity components and pressure field are computed as defined in section 2.2.2, equation (2.27), for every time step from 100 s to 400 s; the final metric is obtained by intersecting these following the method from Bui et al. 2012, § 4.1. This final metric is used by ParMMG to refine the mesh, then the flow problem (3.11) is solved on this new mesh; the process is iterated until the adaptively refined mesh is not longer dependent on the initial mesh (see C. Dapogny et al. 2014, for a detailed explanation).

[6] In the direct orthonormal basis $(\boldsymbol{e}_1,\boldsymbol{e}_2,\boldsymbol{e}_3)$ oriented such that $\boldsymbol{e}_1$ points from inlet to outlet and $\boldsymbol{e}_3$ points from the basis to the top of the building.

The velocity field solution of the problem on the initial mesh (generated with GiD) is reported in figure 16. An example of a refined mesh with approximately 5 000 000 elements is given in figure 17. A snapshot of the velocity field on this refined mesh can be observed in figure 18.

Regarding the temporal discretisation: the default time step choice is 0.25 s, independently of the spatial discretisation. This independence is reasonable because the minimal element size varies little between meshes, even though the number of elements changes much. This is a consequence of the metric-based refinement, with a metric computed from the Hessian of the solution.

3.2.3 Parallel computing

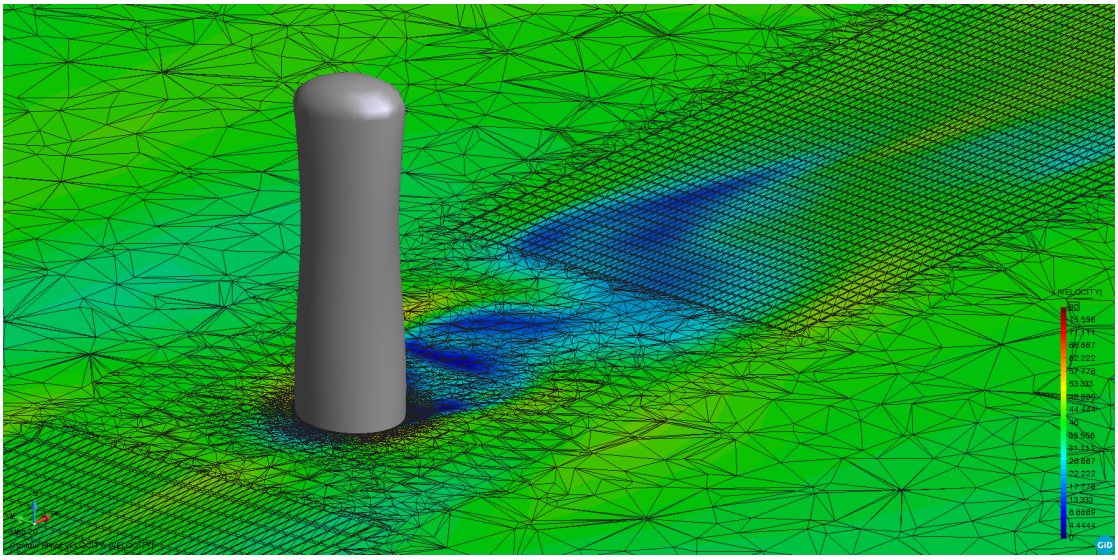
These computations are run on the super-computer Karolina, the latest and most powerful of the IT4Innovations cluster built in 2021 in Ostrava, Czech Republic. Karolina is operated by Red Hat Enterprise Linux 7.9.2009 and its computing resources are allocated through OpenPBS. Each computing node has the following specifications:

- 2 processors AMD EPYC™ 7H12 of 64 cores at 2.6 GHz;
- 256 GiB DDR4 of physical memory with 3200 MT s^{-1};
- 5324.8 GFLOPS;
- 100 Gbit s^{-1} data transfer.

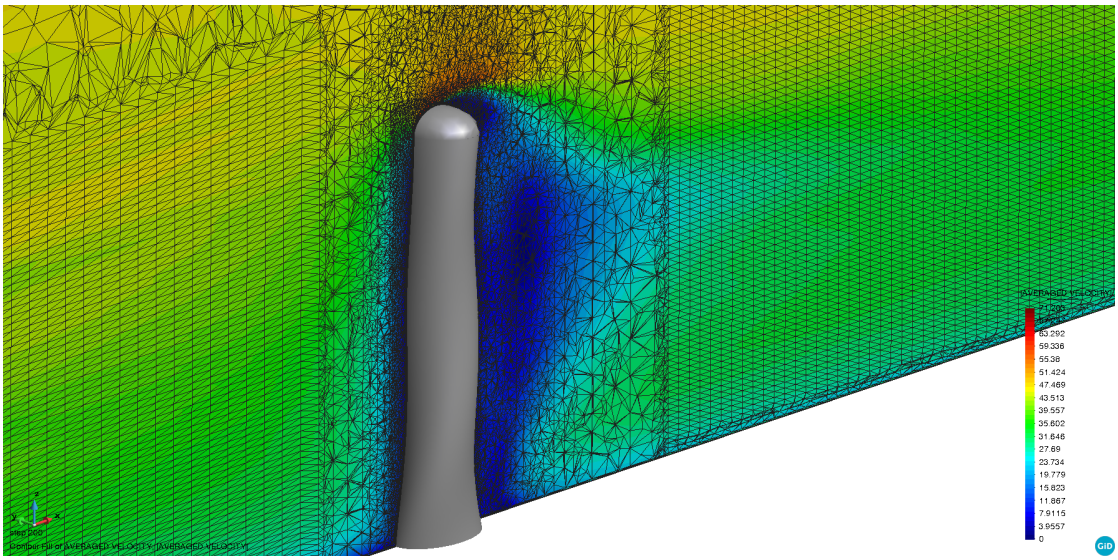
The software stack used is as described in the introduction: XMC, Kratos Multiphysics and MMG+ParMMG; COMPSs is the framework used for parallel computing. Each mesh used – including the initial, unrefined one – is divided into 96 partitions. ParMMG therefore performs the adaptive remeshing in 96 processes running in parallel, communicating via the MPI standard. Likewise, Kratos runs each simulation as an MPI-parallel task with 96 threads. These threads can be shared across computing nodes by groups of 32, so that 4 simulations fit exactly 3 computing nodes. Nota bene (NB) 32 processors are reserved on the main node for COMPSs.

3.2.4 Statistical study

Models and costs As was mentioned in the description of the MFMC in section 3.1, a good low-fidelity model has a low cost and high correlation to the next model in the hierarchy. Although costs are reasonably easy to predict or estimate it is much more difficult for the correlations, as they are affected by many parameters.

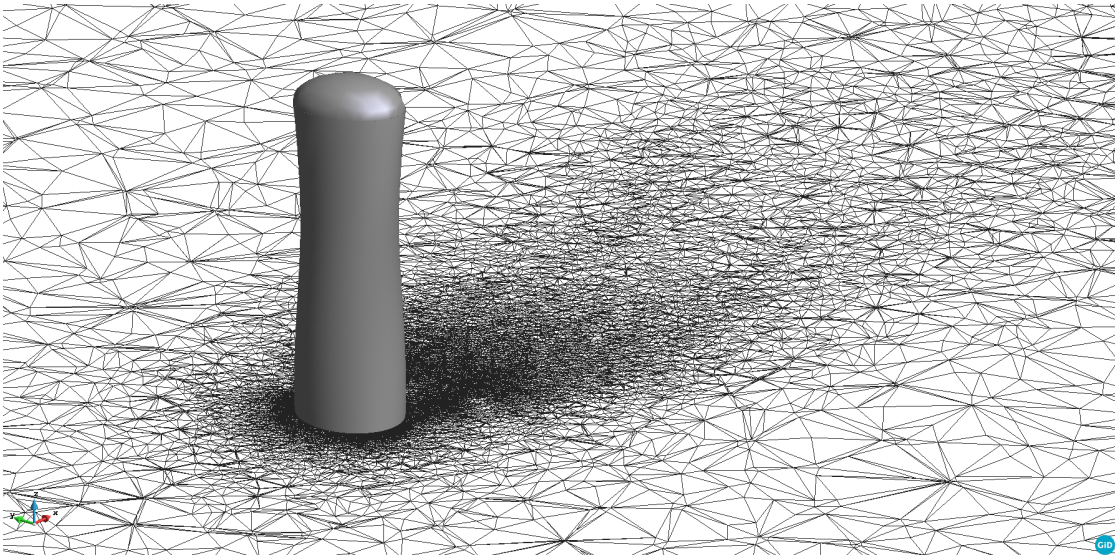


(a) Horizontal cut

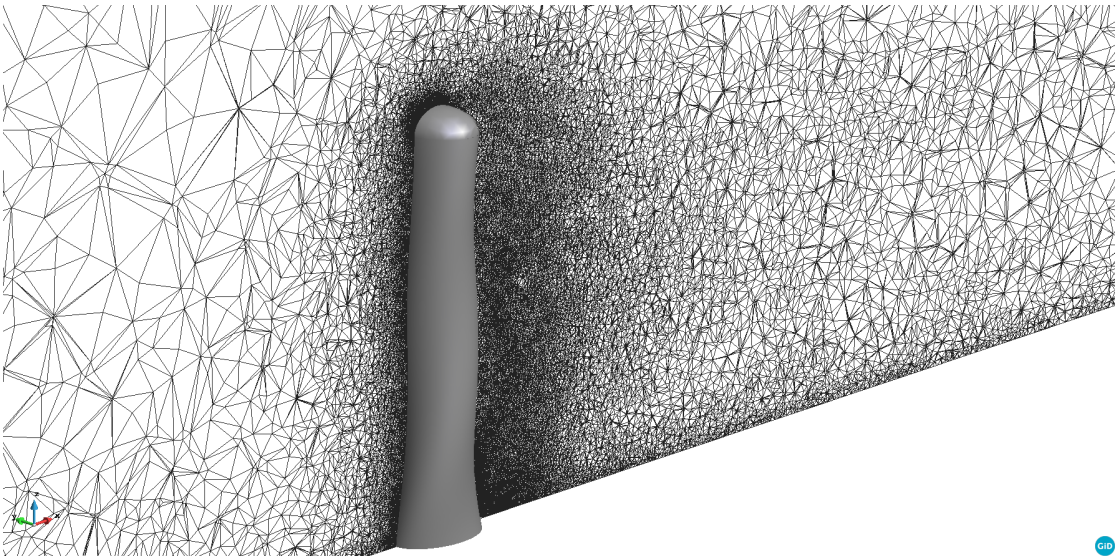


(b) Vertical cut

Figure 16: Example of velocity field on initial mesh



(a) Horizontal cut



(b) Vertical cut

Figure 17: Refined mesh of 5 000 000 elements

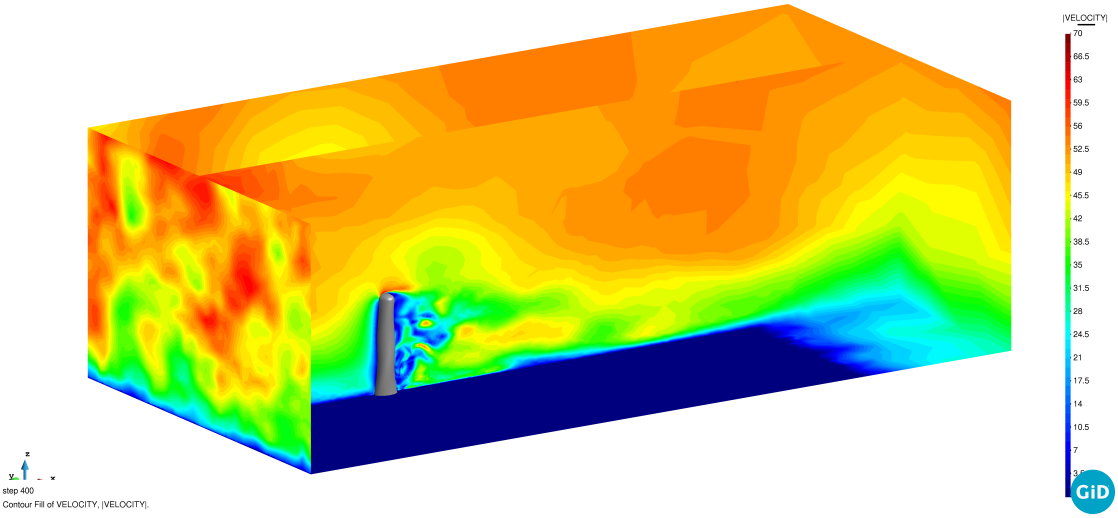


Figure 18: Example of velocity field at 400 s on refined mesh

Table 3: List of meshes generated.

Model	Elements	Time step (s)	Cost (s)
5	5 000 000	0.25	3 539 086
4	2 000 000	0.25	1 445 968
3	1 000 000	0.25	826 990
2	500 000	0.25	390 901
1	100 000	0.25	128 506
0	61 719	0.90	44 149

The point of this statistical study is to assess the quality of several candidate models and chose the ones to be used in the MFMC estimator.

Following the process described in section 3.2.2, we generate the mesh of the high-fidelity model and five coarser meshes candidate to be low-fidelity models. They are presented in table 3, where model 5 is the high-fidelity one; the 'Cost' column is explained below. The mesh of model 0 was created by coarsening the initial mesh as much as possible while preserving mesh quality. To further reduce its cost, its time step was also increased.

To assess the cost, correlations and variances of this set of six models, we draw their realisations for a set of 50 events: (I) fifty i.i.d. realisations of roughness heights $(z_0^{(i)})_{i=1}^{50}$ and associated wind fluctuations are drawn; (II) problem (3.11) is solved by Kratos Multiphysics with each of the six models for each of the fifty events; (III) from the values of the pressure over the building at each time step, the quantities of interest $\langle \boldsymbol{F} \cdot \boldsymbol{e}_1 \rangle_{t_0,600}$ and $\langle \boldsymbol{M} \cdot \boldsymbol{e}_2 \rangle_{t_0,600}$ are computed for each

model and event. Another measure taken for each fluid simulation is its cost, i.e. computational time. This cost is the sum of the time spent by each of the 96 processors used by the simulation (see § 3.2.3); it includes the computation of the quantities of interest. For each model, the average of this cost across all fifty simulations is given in table 3. NB this cost showed very little variation between simulations.

Then, the relevant statistics of each quantity of interest can be computed from these fifty samples of each model. For the sake of brevity, we will refer henceforth to $\langle \boldsymbol{F} \cdot \boldsymbol{e}_1 \rangle_{t_0,600}$ as 'drag force' and to $\langle \boldsymbol{M} \cdot \boldsymbol{e}_2 \rangle_{t_0,600}$ as 'base moment'. One can see from their definition that they depend on the transient time t_0 chosen.

Statistics for a priori transient time Our a priori choice is to set the transient time equal to the washout time, which is the time the flow needs to go from the inlet to the outlet with the reference velocity. It is shown in deliverable 3.3, § 4 that this is generally a conservative choice for this class of problems. In our setting the distance from the inlet to the outlet is 1440 m and the reference velocity is $40\,\mathrm{m\,s^{-1}}$, therefore the washout time is 36 s. For a transient time of 36 s, the correlation matrix of the six models is given in table 4 along with the variance and expectation of each model, for both QoI. The expectation is given only to provide perspective on the size of the variance; the difference in expectation between models is not relevant to this study, as will be demonstrated.

Let us consider the drag force statistics first. Since the high-fidelity model is set to be model 5, let us focus on the associated column (or row) in the correlation matrix. First, the correlation does not vary monotonously with the mesh refinement; indeed, the three models with the least correlation to model 5 are models 2, 3 and 4. This can also be observed on other pairs of consecutive models – e.g. (3,4), (2,3). Second, models 0 and 1 are almost identical in their correlations to other models, despite having quite different expectations and variances; they are almost perfectly correlated themselves. Since model 0 is significantly less expensive than model 1, it seems a better choice.

Let us study the base moment statistics next. Although the correlations are generally higher, the highest correlations with model 5 are not much higher than for the drag force. Unlike for the drag force, the correlation of models 1, 2, 3 and 4 to model 5 increases with the mesh refinement. It is surprising that models 0 and 1 appear negatively correlated with the others. Although the sign of the correlation does not affect the quality of a low-fidelity model, in this case it is unexpected and does not encourage to trust these models – especially since their drag force shows positive correlation. Finally, a lesser surprise is that model 0 shows correlations quite different than model 1 despite being highly-correlated to it; particularly, model 0 has the highest correlation to model 5 while model 1 has the lowest. As

Table 4: Model correlations, variances and expectations for $t_0 = 36\,\mathrm{s}$

(a) Drag force

Models	0	1	2	3	4	5	Variance	Expectation
0	1	0.997	0.414	0.567	0.462	0.594	$7.74 \cdot 10^9$	$1.16 \cdot 10^7$
1		1	0.421	0.588	0.45	0.596	$1.12 \cdot 10^9$	$6.90 \cdot 10^6$
2			1	0.278	0.403	0.458	$1.41 \cdot 10^9$	$5.86 \cdot 10^6$
3				1	0.171	0.223	$1.48 \cdot 10^9$	$5.74 \cdot 10^6$
4					1	0.352	$1.52 \cdot 10^9$	$5.95 \cdot 10^6$
5						1	$2.33 \cdot 10^9$	$6.15 \cdot 10^6$

(b) Base moment

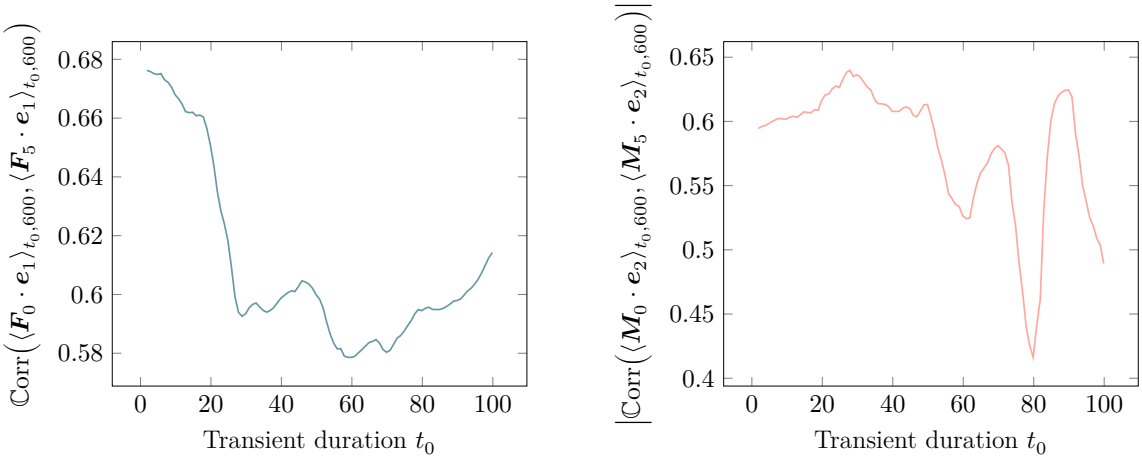
Models	0	1	2	3	4	5	Variance	Expectation
0	1	0.947	−0.752	−0.71	−0.676	−0.616	$6.98 \cdot 10^7$	$7.15 \cdot 10^6$
1		1	−0.591	−0.526	−0.581	−0.469	$4.57 \cdot 10^8$	$8.36 \cdot 10^6$
2			1	0.774	0.507	0.557	$7.23 \cdot 10^9$	$7.76 \cdot 10^6$
3				1	0.519	0.563	$8.13 \cdot 10^9$	$7.52 \cdot 10^6$
4					1	0.614	$5.90 \cdot 10^9$	$7.08 \cdot 10^6$
5						1	$6.06 \cdot 10^9$	$7.08 \cdot 10^6$

for the drag force, model 0 seems like the best low-fidelity model.

Choice of transient time The information available from these simulations can be used to improve the a priori choice of transient time. This duration must be large enough to remove the bias of initial conditions. However, increasing this duration either reduces the accuracy of the model (if the final time T is kept fixed) or increases the cost (if the effective time $[t_0, T]$ is kept fixed). Extensive experiments have been conducted during the ExaQUte project on strategies to detect an optimal transient time duration for different time-dependent systems. A strategy studied in deliverable 5.4, § 5.2 and adapted from Beyhaghi et al. 2018, § 1 is to choose the transient duration which minimises the variance of the MC estimation of the time average. A more comprehensive study with further suggestions is made in deliverable 3.3, § 3.

In the particular case of this study, correlation is chosen as the basis for the selection of the transient time. On the one hand the transient time should be large enough for the influence of the initial conditions on the correlation not to be felt; on the other hand the effective duration[7] should be as large as possible, and we expect a long transient time to reduce correlation. Model 0 is selected as the most promising low-

[7]Duration over which the temporal averages are computed. In this case, it equals $600 - t_0$.



(a) Drag force

(b) Base moment

Figure 19: Correlation between models 0 and 5 as a function of transient time

fidelity model, and its correlation to model 5 is plotted as a function of the transient time for both QoI in figure 19. Figure 19a shows $\mathbb{C}\mathrm{orr}\left(\langle \boldsymbol{F}_0 \cdot \boldsymbol{e}_1 \rangle_{t_0,600}, \langle \boldsymbol{F}_5 \cdot \boldsymbol{e}_1 \rangle_{t_0,600}\right)$ while figure 19b shows $\mathbb{C}\mathrm{orr}\left(\langle \boldsymbol{M}_0 \cdot \boldsymbol{e}_2 \rangle_{t_0,600}, \langle \boldsymbol{M}_5 \cdot \boldsymbol{e}_2 \rangle_{t_0,600}\right)$, with

$$\mathbb{C}\mathrm{orr}(Q_0, Q_5) = \frac{\mathbb{C}\mathrm{ov}(Q_0, Q_5)}{\sqrt{\mathbb{V}\mathrm{ar}(Q_0)\,\mathbb{V}\mathrm{ar}(Q_5)}}$$

where Q stands for the chosen QoI; see (3.10) for the sample estimation of covariances. At the reference velocity of $40\,\mathrm{m\,s^{-1}}$ there are two durations of note: (I) the distance from inlet to building is covered in 9.56 s; (II) the distance from inlet to outlet is covered in 36.0 s.

Figure 19a shows the correlation of the drag force for a transient time up to 90 s; we deem that increasing it beyond 90 s would be detrimental to the models' accuracy. The bias of the fixed initial conditions is obvious on the first thirty seconds. It seems that any value from 30 s to 50 s would be a valid transient time. We choose 46 s, as it is further away from the tipping point of 30 s yet still shows good correlation

The behaviour in figure 19b is more surprising: the effect of initial conditions is not visible as in figure 19b, and the correlation varies wildly with the transient duration. If anything, the only apparent effect of the initial conditions seems to be to attenuate these variations. We choose 28 s, although any value below 50 s appear to be an acceptable choice. NB the *absolute* correlation is plotted here in figure 19b; the correlation values are negative.

Table 5: Model correlations, variances and expectations for $t_0 = 46\,\mathrm{s}$

Models	0	1	2	3	4	5	Variance	Expectation
0	1	0.997	0.469	0.595	0.492	0.605	$8.51 \cdot 10^9$	$1.16 \cdot 10^7$
1		1	0.475	0.611	0.483	0.606	$1.26 \cdot 10^9$	$6.93 \cdot 10^6$
2			1	0.335	0.442	0.471	$1.51 \cdot 10^9$	$5.88 \cdot 10^6$
3				1	0.234	0.245	$1.53 \cdot 10^9$	$5.76 \cdot 10^6$
4					1	0.373	$1.65 \cdot 10^9$	$5.96 \cdot 10^6$
5						1	$2.50 \cdot 10^9$	$6.17 \cdot 10^6$

Table 6: Model correlations, variances and expectations for $t_0 = 28\,\mathrm{s}$

Models	0	1	2	3	4	5	Variance	Expectation
0	1	0.917	−0.787	−0.743	−0.651	−0.639	$1.15 \cdot 10^8$	$7.15 \cdot 10^6$
1		1	−0.597	−0.517	−0.541	−0.475	$4.43 \cdot 10^8$	$8.37 \cdot 10^6$
2			1	0.779	0.483	0.55	$7.27 \cdot 10^9$	$7.76 \cdot 10^6$
3				1	0.495	0.56	$8.02 \cdot 10^9$	$7.52 \cdot 10^6$
4					1	0.586	$5.52 \cdot 10^9$	$7.08 \cdot 10^6$
5						1	$5.93 \cdot 10^9$	$7.08 \cdot 10^6$

STATISTICS FOR ADAPTED TRANSIENT TIME The same statistics as given previously in table 4 for the a priori transient duration of 36 s are now computed for the transient duration adapted from the correlation between model 0 and 5 for each QOI: 46 s for the drag force and 28 s for the base moment.

The statistics for the drag force are given in table 5. It is very similar to table 4a, with slightly higher correlations. The statistics for the base moment in table 6 show more heterogeneous differences: e.g. models 0 and 1 are more correlated to model 5 but less correlated between themselves.

SELECTION OF MODELS Beyond gaining general insight into the behaviour of the numerical models for this problem, the goal of this statistical study is to select suitable low-fidelity models for the MFMC estimator. Moreover, algorithm 2 will calibrate the MFMC estimator with respect to a single QOI, which has to be chosen. The surprising behaviour of the base moment marks it as unreliable target for the tuning of this estimator. The drag force will therefore be the quantity upon which the estimator is calibrated: henceforth $Q_l := \langle \boldsymbol{F}_l \cdot \boldsymbol{e}_1 \rangle_{46,600}$, where $\boldsymbol{F}_l$ comes from model l.

For various selections of low-fidelity models, we predict the cost to estimate $\mathbb{E}(Q_5)$ with a MFMC estimator calibrated according the 'separate calibration' strategy defined by (3.4) and (3.6). The target tolerance η is the statistical error of a MC estimator with 100 samples: $\eta := \sqrt{\mathbb{V}\mathrm{ar}(Q_5)/100}$. The factor of cost

reduction achieved by the MFMC estimator compared to the MC estimator of the same accuracy is defined as

$$\frac{\sum_l \check{s}_l \mathbb{E}_l}{100\,\mathbb{E}(Q_5)} =: \text{cost factor} \tag{3.12}$$

where the sum is over the levels, each comprising two models (barring the first level); $\mathbb{E}_l$ is the cost of level l; $\check{s}_l$ is the unrounded, optimal sample size of level l as defined in (3.6) (i.e. $\check{m}_l = \lceil \check{s}_l \rceil$). For example, for the model selection $(0, 2, 5)$, the cost factor is

$$\frac{\check{s}_0\,\mathbb{E}(Q_0) + \check{s}_1(\mathbb{E}(Q_0) + \mathbb{E}(Q_2)) + \check{s}_2(\mathbb{E}(Q_2) + \mathbb{E}(Q_5))}{100\,\mathbb{E}(Q_5)}.$$

The unrounded[8] samples sizes and cost factors are given in table 7, with a different model selection on each row. An immediate observation is that only three model selections achieve a cost reduction: $(0, 1, 5)$, $(1, 5)$ and $(0, 5)$. This is coherent with table 5, which already suggested that models 0 and 1 were the best low-fidelity models by a fair margin. Given that these two models are highly correlated but have quite different costs, it is not surprising either that model 0 performs better alone. With the current models available, there is no benefit to using multiple low-fidelity models.

Let us next make a similar comparison with the 'coupled calibration' strategy summarised by (3.8)–(3.9). First, let us recall assumption 2. For each model pair (i, j) we compute

$$\Upsilon(i,j) := \mathbb{C}\mathrm{orr}\big(Q_i, Q_j\big)^2 \frac{\mathbb{E}\big(Q_j\big) + \mathbb{E}(Q_i)}{\mathbb{E}\big(Q_{\min\{i,j\}}\big)}.$$

Therefore, $\Upsilon(i, j) > 1$ if and only if assumption 2 is true. Table 8 shows the values of Υ for all models pairs. It gives interesting insights into the results from table 7: the model pairs which performed best in 7 are those without any model pairing (i, j) with $\Upsilon(i, j) < 1$. E.g. compare the estimators 'MFMC–3' with estimators 'MFMC–2', where the model pairs $((2, 4), (4, 5))$ were replaced with the pair $(2, 5)$; likewise, the low values on the second diagonal relate to the poor performance of estimator MFMC–5. For MFMC estimators with two models ('MFMC–1'), unsurprisingly, a higher value of Υ is associated with better performance. Another conclusion from table 8 is that the only low-fidelity models that can be considered for coupled calibration with model 5 are models 0, 1 and 2.

[8] Since the reference accuracy of $\mu_{100}(Q_5)$ is arbitrary, unrounded sample sizes are more informative on how they would scale up for higher accuracy ($\mu_{1000}(Q_5)$, $\mu_{10\,000}(Q_5)$, et cætera (etc.)).

Table 7: Cost reduction with separate calibration (fixed statistical error)

Estimator	Unrounded sample sizes						Cost factor
	0	1	2	3	4	5	
MC						100	1.0
MFMC–5	10.26	0.38	5.57	13.44	52.29	108.60	1.921
MFMC–4a	10.23		6.14	13.48	52.38	108.59	1.921
MFMC–4b	67.52	2.51	36.68		52.15	105.86	1.825
MFMC–3a	67.04		40.24		52.10	105.56	1.815
MFMC–3b	188.67	7.03	102.50			91.79	1.197
MFMC–2a	185.08		111.10			90.58	1.166
MFMC–2b	478.82	17.86				69.34	0.787
MFMC–1a					78.22	104.72	1.795
MFMC–1b				60.61		104.33	1.429
MFMC–1c			154.04			90.89	1.179
MFMC–1d		294.16				72.33	0.856
MFMC–1e	470.43					68.79	0.755

Table 8: Assessment Υ of assumption 2 for the drag. >1 required.

Models	0	1	2	3	4	5
0	2	3.89	2.165	6.995	8.183	29.673
1		2	0.911	2.778	2.856	10.471
2			2	0.351	0.916	2.234
3				2	0.15	0.317
4					2	0.48
5						2

Table 9: Cost reduction with coupled calibration (fixed statistical error)

Low-fidelity model	Unrounded sample sizes		Cost factor
	Level 0	Level 1	
		100	1.0
2	68.08	90.46	1.080
1	238.92	71.79	0.831
0	424.43	68.52	0.747

For each model pair $(0, 5)$, $(1, 5)$ and $(2, 5)$, the MFMC estimator is calibrated for the same tolerance η as previously. The cost reduction factor is also computed following (3.12) as for table 7, with the unrounded sample sizes from (3.8). The results are given in table 9 and prompt two observations. First, all model pairs perform better with coupled calibration than with separate calibration (confer (cf.) table 7). Here as well, a high value of Υ is associated with better performance. Secondly, the model pair $(2, 5)$ does not perform better than the MC reference given in the first line, despite the fact that $\Upsilon(2, 5) = 2.234 > 1$; both are roughly equal in cost. This is because assumption 2 is only sufficient under the approximation that $\mathbb{C}(Q_2) + \mathbb{C}(Q_5) \approx \mathbb{C}(Q_5)$, whereas $\mathbb{C}(Q_2) + \mathbb{C}(Q_5) = 1.11\,\mathbb{C}(Q_5)$. This is likely to happen for values of Υ close to (or lower than) 2, a value achieved e.g. with the same model as both high-fidelity and low-fidelity.

The conclusion of this statistical study is the selection of (I) the drag force as the QoI for which to tune the MFMC estimator; (II) 46 s as the transient time for the computation of the time averages of the QoI; (III) model 0 as the single low-fidelity model to be used in the MFMC estimator. Next, section 3.2.5 will demonstrate the application of algorithm 2 to this MFMC estimation.

Comparison with previous study A briefer yet comparable study was carried out in deliverable 5.4, § 5.1. The fluid problem was the turbulent bi-dimensional flow over a rectangle from deliverable 5.3, § 3: an horizontal cut of the CAARC standard tall building from Braun and Awruch 2009. This problem was one of the intermediary benchmarks of the ExaQUte project (see deliverable 5.3, fig. 6) leading to the turbulent three-dimensional problem presented in section 3.2.1 and also featured in deliverable 7.4. As such, it was a simpler problem in several respects: (I) the spatial domain was bi-dimensional; (II) the Reynolds number was lower (10^5 versus 10^8); (III) the stochastic inlet boundary condition used a simplified model. The spatial dimensions were similar but were scaled down, and the bi-dimensional mesh was refined from a metric based on the time-averaged velocity field alone. Regarding point (III), the wind at the inlet was modelled as a random velocity

constant in time and space, following the normal distribution $\mathcal{N}(2.0\,\mathrm{m\,s^{-1}}, 0.02)$. The wind model used in the current study uses a space-dependent random profile as well as time-dependent fluctuations, based on Mann 1998; the reference velocity is also higher at $40\,\mathrm{m\,s^{-1}}$. This wind model is improved from measured data in deliverable 7.4.

The only QoI considered was the drag force, averaged from 140 s to 300 s. The extension of the final time to 600 s is part of the scaling up of the problem size, along with the mesh sizes (10^4 versus 10^6 nodes). The reduction of the transient time is partially due to the research on initial conditions led to that effect, reported in deliverable 3.3, § 3. The base moment was included in the current study since it is the QoI considered in deliverable 7.4; deliverable 6.4, problem 7, § 5.2.1 and problem 8, § 5.3.

The study in deliverable 5.4, § 5.1, p. 28 considered a different MFMC estimator, recalled in section 3.1.1. As in the current study, the numerical models were obtained from different discretisations. The correlations were only slightly better. However, the ratio of cost from most to least expensive models was much more favourable: 260 versus 80; this may owe more to the temporal discretisation than to the spatial one – cf. table 3 here and table 2 from ibid.

Remarkably, both studies discarded all but the coarsest low-fidelity model as inefficient. This may be partially explained by the type of low-fidelity models: sequences of adaptively-refined meshes were initially chosen for MLMC methods, whose requirements are different from MFMC methods (e.g. assumption 1 versus assumption 2). The resulting numerical models have proven to be too close in cost to the high-fidelity model. However, there is an additional reason: although the choice of low-fidelity models is usually a compromise between high correlation and low cost, in both studies the least expensive model showed the highest correlation to the high-fidelity model. This is more surprising and could be blamed on the chaotic behaviour of a turbulent flow: even a small change in mesh refinement shows phenomena largely different, to the point that a spatial resolution closer to that of the high-fidelity model is no guarantee of better correlation; in this study, the intermediate spatial resolutions appeared noisier than the coarsest ones. Another consideration is the nature of the QoI – here quantities integrated in space and time. These observations suggest that a multi-fidelity approach would perform better with low-fidelity models of a radically different type: e.g. less descriptive physical models, reduced-order models or even trained oracles. Certainly, the chosen models should be much less expensive than those considered in this study.

3.2.5 Demonstration of adaptive MFMC algorithm

Algorithm 2 is now applied to the fluid problem described in section 3.2.1. As selected by the study in section 3.2.4, the high-fidelity model returns Q_5 and the only

Table 10: Iterations of the adaptive multi-fidelity algorithm

Iteration	Tolerance	Sample sizes		Error	Total cost (s)
		Level 0	Level 1		
1			50		$198 \cdot 10^6$
2	5850	310	50	5992	$209 \cdot 10^6$
3	4815	436	75	4724	$316 \cdot 10^6$
4	4170	522	90	4125	$379 \cdot 10^6$

low-fidelity model selected returns Q_0, where $Q_l := \langle \boldsymbol{F}_l \cdot \boldsymbol{e}_1 \rangle_{46,600}$; the algorithm returns an estimation of $\mathbb{E}(Q_5)$. The study in section 3.2.1 is treated as a first iteration: the data (videlicet (viz.) covariances and costs) obtained on both models from this first iteration is used for the first calibration of the sample sizes. We choose a sequence of three values for the calibration tolerance $\tilde{\eta}$: 5850, 4815 and 4170. From the current knowledge of Q_0 and Q_5, these values are expected to lead to the optimal sample sizes $(310, 50)$, $(465, 75)$ and finally $(582, 100)$. The final target tolerance η is 4170 and is therefore expected to be satisfied at iteration 4. If it is not, the calibration tolerance $\tilde{\eta}$ will then be multiplied by 0.9 at every subsequent iteration, to ensure that the sample sizes keep increasing. For perspective, an *absolute* statistical error $\mathbb{E}\big((\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(Q_5) - \mathbb{E}(Q_5))^2\big)^{1/2} = 4170$ amounts to a *relative* statistical error $\mathbb{E}\big((\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(Q_5) - \mathbb{E}(Q_5))^2\big)^{1/2} / |\mathbb{E}(Q_5)| = 6.76 \cdot 10^{-4}$, according to the current estimation of $\mathbb{E}(Q_5)$.

This algorithm is run on the supercomputer Karolina as detailed in section 3.2.3. The algorithm was allocated 25 computing nodes for a total 3200 processors. The progression of the algorithm across iterations is summarised on table 10. This table is read chronologically from top to bottom and also from left to right: on a given row, the calibration tolerance yields the calibrated sample sizes; after computations of the new samples, the error is estimated from the current state of knowledge; finally, the current total cost of the algorithm is updated. Obviously, the new samples sizes are tuned based on the knowledge from the data from the previous iteration. A first observation is that the final tolerance of 4170 on the statistical error has been satisfied at the end of iteration 4, as expected. Secondly, the final sample sizes are noticeably smaller than expected: $(522, 90)$ versus $(582, 100)$. To understand this difference, let us look at the state of knowledge at the end of the algorithm.

There are eventually 90 samples from model 5 and 612 from model 0. The statistics estimated from these samples are given in table 11 for both the drag force (i.e. Q_0 and Q_5) and the base moment. Comparing to table 5 reveals $\mathbb{V}\mathrm{ar}(Q_5)$ had been overestimated, which explains why the eventual sample sizes were smaller

Table 11: Final statistics of numerical models

Model	Drag force		Base moment		Cost (s)
	Variance	Correlation	Variance	Correlation	
0	$10.20 \cdot 10^9$	0.603	$136.83 \cdot 10^6$	-0.608	$52.81 \cdot 10^3$
5	$2.23 \cdot 10^9$		$5.19 \cdot 10^9$		$3.85 \cdot 10^6$

than anticipated. On the other hand, $\mathbb{V}\mathrm{ar}(Q_0)$ had been underestimated; this is likely why iteration 2 failed to achieve the target tolerance of 5850 for which the sample sizes $(310, 50)$ had been chosen. The estimation of the correlation, however, has not changed significantly. The statistics of the base moment use a transient duration of 28.0 s and should therefore be compared to table 6. The variance of the high-fidelity model had been overestimated here too, as well as the correlation: the base moment no longer shows better correlation than the drag force. The slight increase in average costs from table 3 to table 11 could be attributed to the fragmentation of Kratos simulations across several computing nodes; the additional communication has been observed to increase the computational time.

This simulation is meant to exhibit and discuss the performance of algorithm 2. A first natural point of reference is the cost that would have been paid if, after the study in 3.2.4, we had opted for a direct MC estimation of $\mathbb{E}(Q_5)$ with the same target tolerance. The initial estimation $\mathbb{V}\mathrm{ar}(Q_5) \approx 2.50 \cdot 10^9$ would have led to using $\lceil 2.50 \cdot 10^9 / 4170^2 \rceil = 144$ samples of Q_5, hence a total cost of $144 €(Q_5) = 562 \cdot 10^6$ s $= 156 \cdot 10^3$ h. Instead, algorithm 2 cost $379 \cdot 10^6$ s $= 105 \cdot 10^3$ h, which is a reduction by a factor 0.674. This improvement can be attributed to two distinct choices: (I) the MFMC estimator instead of the MC one; (II) the continuation algorithm with a decreasing sequence of calibration tolerances.

To isolate the performance of the MFMC estimation, let us compare instead the cost of algorithm 2 to the cost to achieve the same statistical error of 4125 with a MC estimator tuned based on the *current* knowledge summarised in table 11. It would require $\lceil 2.23 \cdot 10^9 / 4125^2 \rceil = 131$ samples, for a total cost of $504 \cdot 10^6$ s. Consequently, the cost reduction factor attributed purely to the MFMC method is 0.751.

Conversely, the benefit of the continuation algorithm can be quantified by considering two references. First, the sample sizes that would have been used by targeting immediately the final tolerance are $(582, 100)$. The associated cost would have been $582 €(Q_0) + 100(€(Q_0) + €(Q_5)) = 421 \cdot 10^6$ s $= 117 \cdot 10^3$ h, which means that the continuation strategy reduced the cost by a factor 0.9; obviously, this factor depends entirely on the reliability of initial estimations. Secondly, one may remark that the final samples sizes $(522, 90)$ were chosen based on the knowledge

from iteration 3, and wonder if they are overestimated. From the current knowledge (i.e. from table 11), the sample sizes calibrated for the final tolerance 4170 would be $(\lceil 519.59 \rceil, \lceil 88.24 \rceil) = (520, 89)$. The corresponding cost of $375 \cdot 10^6$ s is 0.989 times the realised cost; it is therefore a very close match, as could be expected by the closeness of the final error estimation to the target tolerance. Of course, this comparison is only reliable insofar as the statistics in table 11 are accurate.

The raw results of this numerical experiment and the study in section 3.2.4 can be found in the data set publically available as Ayoul-Guilmard, Rosa M. Badia et al. 2021.

The conclusions from this practical demonstration echo and confirm those from the statistical study in section 3.2.4. The algorithm shows a noticeable cost reduction and appears robust to unreliable initial estimations; the software stack developed demonstrated its ability to manage heavy simulations in a distributed environment. The most straightforward way to improve the performance evinced here is to design low-fidelity models with a lower cost with a higher correlation to the next model in the hierarchy, as there is room for improvement in both directions. For example, a low-fidelity model with the same correlation but a negligible cost would yield a cost reduction factor of 0.637 purely for the MFMC method (versus 0.751 here). The same performance would be achieved with a correlation of 0.7 an unchanged cost. Ideas to lower the cost of such fluid simulations in the context of uncertainty quantification have been proposed and investigated throughout the ExaQUte project (see e.g. list in deliverable 5.4, § 5). Approaches to use shorter durations are studied in deliverable 3.3 and in deliverable 3.4 (based of the research of Krasnopolsky 2018 and Makarashvili et al. 2017); deliverable 6.4, § 3.3 also proposed to make the duration a discretisation parameter, to vary it across the sequence of models as the spatial resolution. The time step sizes could be made to vary, e.g. larger in the transient period (see deliverable 3.3, § 3). Even simplified or radically different fluid models could be chosen based on the QoI to provide inexpensive low-fidelity models. The improvement of the correlation between models is less straightforward and would likely entail gaining a better understanding of the cause of the loss of correlation, e.g. through a careful consideration of the input uncertainty.

4 Conclusion

4.1 Hierarchical Monte Carlo methods for wind engineering

Two different types of hierarchical Monte Carlo methods were presented, for two distinct goals. Each one featured numerical experiments using the open-source software stack describe in introduction, thus demonstrating its ability to scale up to large-scale simulations in a distributed computing environment.

First, a multi-level Monte Carlo (MLMC) method for the estimation of the conditional value at risk (CVaR) was proposed, which can readily yield other statistics (exempli gratia (e.g.) quantile). Its accuracy is controlled by a posteriori error estimations, and it achieves a complexity similar to that of MLMC for the expectation. An algorithm was given to calibrate adaptively the estimator so as to achieve the desired accuracy with minimal cost. This method was then illustrated on two different problems: (I) the Van der Pol oscillator, a non-linear oscillator sometimes used as simplified model for vortex shedding; (II) the potential flow around an airfoil, also used as a benchmark is other aspects of the ExaQUte project (e.g. shape optimisation). Both experiments showed the method to be reliably accurate. Its cost efficiency was shown to be as good or better than predicted. The airfoil experiment also featured a comparison between a hierarchy of meshes fixed a priori, and one refined adaptively.

Secondly, a multi-fidelity Monte Carlo (MFMC) method was proposed to address the case of turbulent flows, whose chaotic behaviour renders MLMC approaches inefficient. A strategy to calibrate optimally the estimator for a fixed hierarchy of models was proposed, along with a necessary condition for the efficiency of a single low-fidelity model. An algorithm to tune this calibration adaptively was also described. Then, this method was illustrated on one of the target benchmarks of the ExaQUte project: a tri-dimensional turbulent flow around a tall building. A detailed statistical study investigated the quality of different levels of discretisation as low-fidelity models. Finally, the performance of the adaptive algorithm was shown on a large-scale simulation, and found to be as good or better than predicted. The limitations of the low-fidelity models considered was discussed, along with possible improvements to achieve better cost reduction.

4.2 Algorithmic extensions of Monte Carlo methods

The research in this work package provided both better understanding and methodological improvements on the use of hierarchical Monte Carlo (MC) methods – MLMC in particular – to simulations of fluid flow with uncertain boundary conditions.

Adaptive mesh refinement, unsteady flow solutions and wind engineering applications were the respective focus of deliverable 5.3, deliverable 5.4 and deliverable 5.5. A notable contribution is the accurate and efficient estimation of the CVaR (and other related statistics such as quantiles) with an adaptive MLMC algorithm. The most prominent difficulty was the chaotic behaviour of quantity of interest (QoI) for turbulent flows, which defeats MLMC approaches. An alternative was proposed in the form of a MFMC approach and was found to be beneficial, even though various points to be improved were identified; insights and suggestions were given regarding the design and choice of suitable numerical models.

Finally, one of the primary goals of this work package was to provide first the ExaQUte project then the larger research community with open tools to use these methods. Those were implemented in the XMC library, integrated with the solver Kratos Multiphysics and interfaced with both COMPSs and Quake via the common ExaQUte application programming interface (API) for distributed programming. All these tools are publicly available, with open code and licenses.

4.3 Beyond the ExaQUte project

Numerous promising research directions have been identified throughout the project. Plans to combine more effectively adaptive mesh refinement with MLMC algorithms have been outlined in deliverable 5.3. Several ways in which the different temporal discretisation and spans could be leveraged in a MLMC approach were discussed in deliverable 5.4; related investigations are also reported in deliverable 3.3. The novel approach for CVaR estimation described here comes with ideas of possible improvement, such as the adaptation of the interpolation interval to bring the error estimation closer to the exact error. Following convincing results on the inadvisability of MLMC with turbulent flows, the MFMC approach studied in this report shows encouraging results; it is a more recent research topic with many possible improvements. We predict that investigation into more efficient low-fidelity models will be rewarding in terms of cost reduction; many possibilities have been suggested and some are even studied in ibid. and deliverable 3.4. Furthermore, the hierarchical structure of the proposed MFMC estimator is similar enough to that of a MLMC estimator that developments made for the latter – e.g. CVaR estimation – could be adapted to the former. Similarly, the research reported in deliverable 6.5 on the use of MLMC for the estimation of sensitivities in optimisation under uncertainties (OUU) could be considered for MFMC.

Regarding the software output of ExaQUte, further development of XMC is planned for the near future at EPFL, with a focus on stabilisation, broader adoption and usability. More generally, every partner has declared its intention to keep maintaining their respective tools. Their exploitation plans can be found in deliverable 8.5.

References

General

Babuška, Ivo, Fabio Nobile and Raúl Tempone (2007). 'A stochastic collocation method for elliptic partial differential equations with random input data'. In: *SIAM Journal on Numerical Analysis* 45.3, pp. 1005–1034.

Badia, Rosa M., J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes and R. Sirvent (Dec. 2015). 'COMP Superscalar, an interoperable programming framework'. In: *SoftwareX* 3–4. DOI: 10.1016/j.softx.2015.10.004.

Beyhaghi, Pooriya, Shahrouz Alimohammadi and Thomas Bewley (Feb. 2018). 'A multiscale, asymptotically unbiased approach to uncertainty quantification in the numerical approximation of infinite time-averaged statistics'. In: *arXiv e-prints*. arXiv: 1802.01056.

Bierig, Claudio and Alexey Chernov (2016). 'Approximation of probability density functions by the multilevel Monte Carlo maximum entropy method'. In: *Journal of Computational Physics* 314, pp. 661–681.

Braun, Alexandre Luis and Armando Miguel Awruch (May 2009). 'Aerodynamic and aeroelastic analyses on the CAARC standard tall building model using numerical simulation'. In: *Computers & Structures* 87.9, pp. 564–581. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2009.02.002.

BSC, Barcelona Supercomputing Centre, *COMP Superscalar* version 2.9.0. URL: https://compss.bsc.es/.

Bui, C., C. Dapogny and P. Frey (10th Nov. 2012). 'An accurate anisotropic adaptation method for solving the level set advection equation'. In: *International Journal for Numerical Methods in Fluids* 70.7, pp. 899–922. DOI: 10.1002/fld.2730.

Codina, Ramon, Santiago Badia, Joan Baiges and Javier Principe (2017). 'Variational Multiscale Methods in Computational Fluid Dynamics'. In: *Encyclopedia of Computational Mechanics Second Edition*. Chichester, UK: John Wiley & Sons, Ltd, pp. 1–28. ISBN: 9781119176817. DOI: 10.1002/9781119176817.ecm2117.

Collier, Nathan, Abdul-Lateef Haji-Ali, Fabio Nobile, Erik von Schwerin and Raúl Tempone (1st June 2015). 'A continuation multilevel Monte Carlo algorithm'. In: *BIT Numerical Mathematics* 55.2, pp. 399–432. ISSN: 1572-9125. DOI: 10.1007/s10543-014-0511-3.

Dadvand, P., R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S.R. Idelsohn and E. Oñate (July 2013). 'Migration of a generic multi-physics framework to'

HPC environments'. In: *Computers & Fluids* 80, pp. 301–309. DOI: 10.1016/j.compfluid.2012.02.004.

Dadvand, Pooyan, Riccardo Rossi and Eugenio Oñate (July 2010). 'An Object-oriented Environment for Developing Finite Element Codes for Multi-disciplinary Applications'. In: *Archives of Computational Methods in Engineering* 17.3, pp. 253–297. DOI: 10.1007/s11831-010-9045-2.

Dapogny, C., C. Dobrzynski and P. Frey (Apr. 2014). 'Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems'. In: *Journal of Computational Physics* 262, pp. 358–378. ISSN: 10902716. DOI: 10.1016/j.jcp.2014.01.005.

Dapogny, Charles, Cécile Dobrzynski, Pascal Frey and Algiane Froehly, *MMG* version 5.6.0, 5th Nov. 2021. INRIA. LIC: GNU Lesser General Public License. VCS: https://github.com/MmgTools/mmg, SWHID: ⟨swh:1:rev:889d408419b5c48833c249695987cf6ec699d399⟩.

Davari, M., R. Rossi, P. Dadvand, I. López and R. Wüchner (May 2019). 'A cut finite element method for the solution of the full-potential equation with an embedded wake'. In: *Computational Mechanics* 63.5, pp. 821–833. ISSN: 01787675. DOI: 10.1007/s00466-018-1624-3.

Dong, S., G. E. Karniadakis and C. Chryssostomidis (Mar. 2014). 'A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains'. In: *Journal of Computational Physics* 261, pp. 83–105. ISSN: 10902716. DOI: 10.1016/j.jcp.2013.12.042.

Ehsan, Fazl and Robert H. Scanlan (1990). 'Vortex-induced vibrations of flexible bridges'. In: *Journal of Engineering Mechanics* 116.6, pp. 1392–1411. DOI: 10.1061/(ASCE)0733-9399(1990)116:6(1392).

Felippa, Carlos A. (2004). 'Introduction to finite element methods'. Lecture notes (Aerospace engineering sciences department). University of Colorado at Boulder.

Frey, P. J. and F. Alauzet (2005). 'Anisotropic mesh adaptation for CFD computations'. In: *Computer Methods in Applied Mechanics and Engineering* 194.48-49, pp. 5068–5082. ISSN: 00457825. DOI: 10.1016/j.cma.2004.11.025.

Ghanem, Roger G. and Pol D. Spanos (1991). *Stochastic Finite Elements: A Spectral Approach*. Springer New York. ISBN: 978-1-4612-3094-6. DOI: 10.1007/978-1-4612-3094-6.

Giles, Michael B and Abdul-Lateef Haji-Ali (2019). 'Multilevel nested simulation for efficient risk estimation'. In: *SIAM/ASA Journal on Uncertainty Quantification* 7.2, pp. 497–525.

Giles, Michael B, Tigran Nagapetyan and Klaus Ritter (2015). 'Multilevel Monte Carlo approximation of distribution functions and densities'. In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1, pp. 267–295.

Giles, Michael B, Tigran Nagapetyan and Klaus Ritter (2017). 'Adaptive multilevel monte carlo approximation of distribution functions'. In: *arXiv preprint arXiv:1706.06869*.

Giles, Michael B. (2008). 'Multilevel Monte Carlo Path Simulation'. In: *Operations Research* 56.3, pp. 607–617. DOI: 10.1287/opre.1070.0496.

Gou, Wenhui (2016). *Estimating value-at-risk using multilevel Monte Carlo maximum entropy method*.

Heinrich, Stefan (20th Dec. 2001). 'Multilevel Monte Carlo Methods'. In: *Large-Scale Scientific Computing*. Ed. by Svetozar Margenov, Jerzy Waśniewski and Plamen Yalamov. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 58–67. ISBN: 978-3-540-45346-8. DOI: 10.1007/3-540-45346-6_5.

Joint committee on structural safety (May 2001). 'Wind load'. In: *Probabilistic model code*. Vol. 2.13. ISBN: 978-3-909386-79-6.

Keith, Brendan, Ustim Khristenko and Barbara Wohlmuth (2021). 'A fractional PDE model for turbulent velocity fields near solid walls'. In: *Journal of Fluid Mechanics* 916. ISSN: 14697645. DOI: 10.1017/jfm.2021.182.

Krasnopolsky, Boris I. (2018). 'An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles'. In: *Computer Physics Communications* 229, pp. 8–19. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2018.03.023.

Krumscheid, Sebastian and Fabio Nobile (2018). 'Multilevel Monte Carlo Approximation of Functions'. In: *SIAM/ASA Journal on Uncertainty Quantification* 6.3, pp. 1256–1293. DOI: 10.1137/17M1135566.

Le Maître, Olivier P. and Omar M. Knio (2010). *Spectral Methods for Uncertainty Quantification*. Springer Netherlands. ISBN: 978-90-481-3520-2. DOI: 10.1007/978-90-481-3520-2.

Lordan, Francesc, Enric Tejedor, Jorge Ejarque, Roger Rafanell, Javier Álvarez, Fabrizio Marozzo, Daniele Lezzi, Raül Sirvent, Domenico Talia and Rosa M. Badia (Sept. 2013). 'ServiceSs: An Interoperable Programming Framework for the Cloud'. In: *Journal of Grid Computing* 12.1, pp. 67–91. DOI: 10.1007/s10723-013-9272-5.

Makarashvili, Vakhtang, Elia Merzari, Aleksandr Obabko, Andrew Siegel and Paul Fischer (2017). 'A performance analysis of ensemble averaging for high fidelity turbulence simulations at the strong scaling limit'. In: *Computer Physics Communications* 219, pp. 236–245. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2017.05.023.

Mann, Jakob (1998). 'Wind field simulation'. In: *Probabilistic Engineering Mechanics* 13.4, pp. 269–282. ISSN: 0266-8920. DOI: 10.1016/S0266-8920(97)00036-2.

Melendo, A., A. Coll, M. Pasenau, E. Escolano and A. Monros, *GiD: The personal pre- and post-processor* version 15, 2021. CIMNE. URL: www.gidhome.com, (visited on 10/11/2021).

Nishida, Brian and Mark Drela (1995). 'Fully simultaneous coupling for three-dimensional viscous/inviscid flows'. In: *13th Applied Aerodynamics Conference*, pp. 355–361. DOI: 10.2514/6.1995-1806.

Pope, Stephen B. (Aug. 2000). *Turbulent Flows*. Cambridge University Press. ISBN: 9780521591256. DOI: 10.1017/CBO9780511840531.

Rockafellar, Tyrrell R. and Johannes O. Royset (Mar. 2015). 'Engineering Decisions under Risk Averseness'. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 1.2. ISSN: 2376-7642. DOI: 10.1061/ajrua6.0000816.

Tejedor, Enric, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M. Badia, Jordi Torres, Toni Cortes and Jesús Labarta (2017). 'PyCOMPSs: Parallel computational workflows in Python'. In: *International Journal of High Performance Computing Applications* 31.1, pp. 66–82. DOI: 10.1177/1094342015594678.

Tibshirani, Robert J and Bradley Efron (1993). 'An introduction to the bootstrap'. In: *Monographs on statistics and applied probability* 57, pp. 1–436.

Van der Pol, Balthasar (1920). 'A theory of the amplitude of free and forced triode oscillations'. In: *Radio Review* 1, pp. 701–710.

Virtanen, Pauli et al. (2020). 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

ExaQUte

Amela, Ramon, Quentin Ayoul-Guilmard, Rosa M. Badia, Sundar Ganesh, Fabio Nobile, Riccardo Rossi and Riccardo Tosi, *XMC* version 2.0.0, 10th Nov. 2020. DOI: 10.5281/zenodo.3235832.

Amela, Ramon, Quentin Ayoul-Guilmard, Rosa Maria Badia, Sundar Ganesh, Fabio Nobile, Riccardo Rossi and Riccardo Tosi (30th May 2019). *Release of ExaQUte MLMC Python engine*. Deliverable 5.2. Version 1.0. ExaQUte consortium.

Apostolatos, Andreas (30th Apr. 2020). *Finalization of deterministic verifcation and validation tests*. Deliverable 7.2. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.006.

Ayoul-Guilmard, Quentin, Rosa M. Badia, Jorge Ejarque, Sundar Ganesh, Anoop Kodakkal, Fabio Nobile, Marc Núñez, Jordi Pons-Prats, Javier Principe, Riccardo Rossi, Cecilia Soriano and Riccardo Tosi (30th Nov. 2021). *ExaQUte Data*. Data set. Version 1. ExaQUte consortium. Zenodo. DOI: 10.5281/zenodo.5729258.

Ayoul-Guilmard, Quentin, Sundar Ganesh, Anoop Kodakkal, Fabio Nobile and Marc Núñez (14th Dec. 2020). *Report on stochastic optimisation for unsteady problems*. Deliverable 6.4. Version 1.0. ExaQUte consortium. 56 pp.

— (30th Nov. 2021). *Report on stochastic optimisation for wind engineering*. Deliverable 6.5. Version 1.0. ExaQUte consortium.

Ayoul-Guilmard, Quentin, Sundar Ganesh, Sebastian Krumscheid and Fabio Nobile (2021). 'Quantifying uncertain system outputs via the multilevel Monte Carlo method — distribution and robustness measures'. In preparation.

Ayoul-Guilmard, Quentin, Sundar Ganesh and Fabio Nobile (27th July 2020). *Report on stochastic optimisation for simple problems*. Deliverable 6.3. Version 1.3. ExaQUte consortium. 43 pp.

Ayoul-Guilmard, Quentin, Sundar Ganesh, Fabio Nobile, Marc Núñez, Riccardo Rossi and Riccardo Tosi (13th Nov. 2020a). *Report on MLMC for time-dependent problems*. Deliverable 5.4. Version 1.0. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.005.

— (30th May 2020b). *Report on theoretical work to allow the use of MLMC with adaptive mesh refinement*. Deliverable 5.3. Version 1.0. ExaQUte consortium. 31 pp.

Ayoul-Guilmard, Quentin, Sundar Ganesh, Fabio Nobile, Marc Núñez and Riccardo Tosi (30th Nov. 2021). *Report on the application of MLMC to wind engineering*. Deliverable 5.5. Version 1.0. ExaQUte consortium.

Badia, Rosa Maria, Stanislav Böhm and Jorge Ejarque (30th Nov. 2021). *Framework development and release*. Deliverable 4.5. Version 1.0. ExaQUte consortium.

Bidier, Sami, Ustim Khristenko, Riccardo Tosi, Roland Wuchner, Alexander Michalski, Riccardo Rossi and Cecilia Soriano (28th Feb. 2021). *Report on UQ results and overall user experience*. Deliverable 7.3. ExaQUte consortium. DOI: 10.23967/exaqute.2021.9.002.

Bidier, Sami, Anoop Kodakkal and Ustim Khristenko (30th Nov. 2021). *Final report on stochastic optimization results*. Deliverable 7.4. Version 1.0. ExaQUte consortium.

Böhm, Stanislav, Jakub Beránek and Martin Surkovsky, *Quake* 30th Nov. 2021. IT4I. URL: https://code.it4i.cz/boh126/quake, (visited on 30/11/2021).

Böhm, Stanislav and Jorge Ejarque, *ExaQUte API* 30th Nov. 2021. IT4I, BSC. URL: https://github.com/ExaQUte-project/exaqute-api, (visited on 30/11/2021).

Cirrottola, Luca and Algiane Froehly (Nov. 2019). *Parallel unstructured mesh adaptation using iterative remeshing and repartitioning*. Research Report RR-9307. INRIA Bordeaux, équipe CARDAMOM.

— (30th Nov. 2021a). *Final release of the mesh generation/adaption capaibilities*. Deliverable 2.5. Version 1.0. ExaQUte consortium.

Cirrottola, Luca and Algiane Froehly, *ParMMG* version 1.4.0, 5th Nov. 2021. INRIA. LIC: GNU Lesser General Public License. VCS: https://github.com/MmgTools/ParMmg, SWHID: ⟨swh:1:rev:be8d5242abaeeafe3d6fd70f5a35064b94b518e1⟩.

Drzisga, Daniel, Mario Teixeira Parete and Roland Wüchner (30th July 2018). *Release of ExaQUte API for MLMC*. Deliverable 5.1. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.026.

Ganesh, Sundar, Quentin Ayoul-Guilmard and Fabio Nobile (30th May 2019). *Report on the calculation of stochastic sensitivities*. Deliverable 6.2. Version 1.1. ExaQUte consortium. 18 pp.

Kodakkal, Anoop and Cecilia Soriano (30th Nov. 2021). *Report on exploitation activities*. Deliverable 8.5. Version 1.0. ExaQUte consortium.

Mataix, Vicente et al., *Kratos Multiphysics* version 9.0, Nov. 2021. DOI: 10.5281/zenodo.3234644,

Nobile, Fabio, Rosa Maria Badia, Jorge Ejarque, Luca Cirrottola, Algiane Froehly, Brendan Keith, Anoop Kodakkal, Marc Núñez, Carlos Roig, Riccardo Tosi, Riccardo Rossi, Cecilia Soriano, Sundar Ganesh and Quentin Ayoul-Guilmard (30th Nov. 2020). *Final public release of the solver*. Deliverable 1.4. Version 1.0. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.009.

Núñez, Marc, Iñigo López, Joan Baiges and Riccardo Rossi (Jan. 2022). 'An embedded approach for the solution of the full potential equation with finite elements'. In: *Computer Methods in Applied Mechanics and Engineering* 388, p. 114244. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.114244.

Tosi, Riccardo, Ramon Amela, Rosa M Badia and Riccardo Rossi (2021). 'A Parallel Dynamic Asynchronous Framework for Uncertainty Quantification by Hierarchical Monte Carlo Algorithms'. In: *Journal of Scientific Computing* 89.1, p. 28. ISSN: 1573-7691. DOI: 10.1007/s10915-021-01598-6.

Tosi, Riccardo, Marc Núñez, Ramon Codina, Jordi Pons-Prats, Javier Principe and Riccardo Rossi (30th Nov. 2021a). *Report of ensemble-based parallelism for turbulent flows and release of solvers*. Deliverable 3.3. ExaQUte consortium.

— (30th Nov. 2021b). *Report on the calibration of parallel methods for transient problems in wind engineering*. Deliverable 3.4. ExaQUte consortium.

Aliases

COMPSs	Barcelona Supercomputing Centre BSC, *COMP Superscalar* version 2.9.0. URL: https://compss.bsc.es/.
deliverable 1.4	Fabio Nobile, Rosa Maria Badia, Jorge Ejarque, Luca Cirrottola, Algiane Froehly, Brendan Keith, Anoop Kodakkal, Marc Núñez, Carlos Roig, Riccardo Tosi, Riccardo Rossi, Cecilia Soriano, Sundar Ganesh and Quentin Ayoul-Guilmard

(30th Nov. 2020). *Final public release of the solver*. Deliverable 1.4. Version 1.0. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.009.

deliverable 2.5 Luca Cirrottola and Algiane Froehly (30th Nov. 2021a). *Final release of the mesh generation/adaption capaibilities*. Deliverable 2.5. Version 1.0. ExaQUte consortium.

deliverable 3.3 Riccardo Tosi, Marc Núñez, Ramon Codina, Jordi Pons-Prats, Javier Principe and Riccardo Rossi (30th Nov. 2021a). *Report of ensemble-based parallelism for turbulent flows and release of solvers*. Deliverable 3.3. ExaQUte consortium.

deliverable 3.4 Riccardo Tosi, Marc Núñez, Ramon Codina, Jordi Pons-Prats, Javier Principe and Riccardo Rossi (30th Nov. 2021b). *Report on the calibration of parallel methods for transient problems in wind engineering*. Deliverable 3.4. ExaQUte consortium.

deliverable 4.5 Rosa Maria Badia, Stanislav Böhm and Jorge Ejarque (30th Nov. 2021). *Framework development and release*. Deliverable 4.5. Version 1.0. ExaQUte consortium.

deliverable 5.1 Daniel Drzisga, Mario Teixeira Parete and Roland Wüchner (30th July 2018). *Release of ExaQUte API for MLMC*. Deliverable 5.1. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.026.

deliverable 5.2 Ramon Amela, Quentin Ayoul-Guilmard, Rosa Maria Badia, Sundar Ganesh, Fabio Nobile, Riccardo Rossi and Riccardo Tosi (30th May 2019). *Release of ExaQUte MLMC Python engine*. Deliverable 5.2. Version 1.0. ExaQUte consortium.

deliverable 5.3 Quentin Ayoul-Guilmard, Sundar Ganesh, Fabio Nobile, Marc Núñez, Riccardo Rossi and Riccardo Tosi (30th May 2020a). *Report on theoretical work to allow the use of MLMC with adaptive mesh refinement*. Deliverable 5.3. Version 1.0. ExaQUte consortium. 31 pp.

deliverable 5.4 Quentin Ayoul-Guilmard, Sundar Ganesh, Fabio Nobile, Marc Núñez, Riccardo Rossi and Riccardo Tosi (13th Nov. 2020b). *Report on MLMC for time-dependent problems*. Deliverable 5.4. Version 1.0. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.005.

deliverable 5.5 Quentin Ayoul-Guilmard, Sundar Ganesh, Fabio Nobile, Marc Núñez and Riccardo Tosi (30th Nov. 2021). *Report on the application of MLMC to wind engineering*. Deliverable 5.5. Version 1.0. ExaQUte consortium.

deliverable 6.2 Sundar Ganesh, Quentin Ayoul-Guilmard and Fabio Nobile (30th May 2019). *Report on the calculation of stochastic sensitivities*. Deliverable 6.2. Version 1.1. ExaQUte consortium. 18 pp.

deliverable 6.3 Quentin Ayoul-Guilmard, Sundar Ganesh and Fabio Nobile (27th July 2020). *Report on stochastic optimisation for simple problems*. Deliverable 6.3. Version 1.3. ExaQUte consortium. 43 pp.

deliverable 6.4 Quentin Ayoul-Guilmard, Sundar Ganesh, Anoop Kodakkal, Fabio Nobile and Marc Núñez (14th Dec. 2020). *Report on stochastic optimisation for unsteady problems*. Deliverable 6.4. Version 1.0. ExaQUte consortium. 56 pp.

deliverable 6.5 Quentin Ayoul-Guilmard, Sundar Ganesh, Anoop Kodakkal, Fabio Nobile and Marc Núñez (30th Nov. 2021). *Report on stochastic optimisation for wind engineering*. Deliverable 6.5. Version 1.0. ExaQUte consortium.

deliverable 7.2 Andreas Apostolatos (30th Apr. 2020). *Finalization of deterministic verifcation and validation tests*. Deliverable 7.2. ExaQUte consortium. DOI: 10.23967/exaqute.2021.2.006.

deliverable 7.3 Sami Bidier, Ustim Khristenko, Riccardo Tosi, Roland Wuchner, Alexander Michalski, Riccardo Rossi and Cecilia Soriano (28th Feb. 2021). *Report on UQ results and overall user experience*. Deliverable 7.3. ExaQUte consortium. DOI: 10.23967/exaqute.2021.9.002.

deliverable 7.4 Sami Bidier, Anoop Kodakkal and Ustim Khristenko (30th Nov. 2021). *Final report on stochastic optimization results*. Deliverable 7.4. Version 1.0. ExaQUte consortium.

deliverable 8.5 Anoop Kodakkal and Cecilia Soriano (30th Nov. 2021). *Report on exploitation activities*. Deliverable 8.5. Version 1.0. ExaQUte consortium.

Kratos Multiphysics Vicente Mataix et al., *Kratos Multiphysics* version 9.0, Nov. 2021. DOI: 10.5281/zenodo.3234644,

MMG Charles Dapogny, Cécile Dobrzynski, Pascal Frey and Algiane Froehly, *MMG* version 5.6.0, 5th Nov. 2021. INRIA. LIC: GNU Lesser General Public License. VCS: https://github.com/MmgTools/mmg, SWHID: ⟨swh:1:rev:889d408419b5c48833c249695987cf6ec699d399⟩.

ParMMG Luca Cirrottola and Algiane Froehly, *ParMMG* version 1.4.0, 5th Nov. 2021. INRIA. LIC: GNU Lesser General Public

License. VCS: https://github.com/MmgTools/ParMmg, SWHID: ⟨swh:1:rev:be8d5242abaeeafe3d6fd70f5a35064b94b518e1⟩.

Quake Stanislav Böhm, Jakub Beránek and Martin Surkovsky, *Quake* 30th Nov. 2021. IT4I. URL: https://code.it4i.cz/boh126/quake, (visited on 30/11/2021).

XMC Ramon Amela, Quentin Ayoul-Guilmard, Rosa M. Badia, Sundar Ganesh, Fabio Nobile, Riccardo Rossi and Riccardo Tosi, *XMC* version 2.0.0, 10th Nov. 2020. DOI: 10.5281/zenodo.3235832.

A Calibration of MFMC estimator

Let $L \in \mathbb{N}$ and $\boldsymbol{\alpha} \in \mathbb{R}_*^L$. Given $L+1$ real-valued random variables $(X_0, \dots, X_L) =: \boldsymbol{X} \in \mathrm{L}^2(\Omega, \mathbb{R}^{L+1})$, we recall the telescopic sum (3.2):

$$\begin{aligned}\mathbb{E}(X_L) &= \sum_{l=0}^{L} \left(\mathbb{E}\big(Y_{l,\alpha_{l-1}}\big) \prod_{k=l}^{L-1} \alpha_k \right) \\ &= \sum_{l=1}^{L} \mathbb{E}(X_l - \alpha_{l-1} X_{l-1}) \prod_{k=l}^{L-1} \alpha_k + \mathbb{E}(X_0) \prod_{k=0}^{L-1} \alpha_k, \end{aligned} \tag{A.1}$$

with the notation $\forall l \in [\![1, L]\!]$

$$Y_{l,\alpha_{l-1}} := X_l - \alpha_{l-1} X_{l-1}; \qquad Y_{0,\alpha_{-1}} := X_0.$$

If every expectation appearing in (A.1) is estimated with an independent MC estimator, the resulting estimator is

$$\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X}) := \sum_{l=0}^{L} \mu_{m_l}(Y_{l,\alpha_{l-1}}) \prod_{k=l}^{L-1} \alpha_k, \tag{A.2}$$

with sample sizes $\boldsymbol{m} = (m_0, \dots, m_L) \in \mathbb{N}_*^{L+1}$. Let us study two different strategies to optimise the control coefficients $\boldsymbol{\alpha}$ and the sample sizes $\boldsymbol{m}$, starting with a simpler case.

A.1 Separate calibration

In this simplified strategy we will first optimise the control coefficients independently of the sample sizes. Then, with these control coefficients, we will seek the cost-optimal sample sizes for a given accuracy.

A.1.1 Calibrated control coefficients

In order to optimise the control coefficient independently from the sample sizes, let us make two simplifying assumptions:

(I) the expectation of the lowest-fidelity model $\mathbb{E}(X_0)$ is known;

(II) the sample sizes are equal on all levels: $\forall l \in [\![0, L]\!]$, $m_l = m_0$.

These two assumptions simplify the MFMC estimator $\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})$ from (A.2) into a MC estimation $\mu_{m_0}(Z_{\boldsymbol{\alpha}})$ of the random variable

$$Z_{\boldsymbol{\alpha}} := \sum_{l=1}^{L} Y_{l,\alpha_{l-1}} \prod_{k=l}^{L-1} \alpha_k + \mathbb{E}(X_0) \prod_{k=0}^{L-1} \alpha_k$$

whose variance can be written

$$\mathbb{V}\mathrm{ar}(Z_{\boldsymbol{\alpha}}) = \mathbb{V}\mathrm{ar}\left(\sum_{l=1}^{L} Y_{l,\alpha_{l-1}} \prod_{k=l}^{L-1} \alpha_k\right).$$

Let us look for the control coefficients which minimise the variance of this estimator:

$$\check{\boldsymbol{\alpha}} := \mathrm{argmin}\Big\{\mathbb{V}\mathrm{ar}\Big(\mu_{m_0}(Z_{\boldsymbol{a}})\Big) : \boldsymbol{a} \in \mathbb{R}_*^L\Big\}. \tag{A.3}$$

The variance can be expressed as a function of the control coefficients:

$$v(\boldsymbol{\alpha}) := \mathbb{V}\mathrm{ar}\Big(\mu_{m_0}(Z_{\boldsymbol{\alpha}})\Big) = \frac{1}{m_0}\,\mathbb{V}\mathrm{ar}\left(\sum_{l=1}^{L} Y_{l,\alpha_{l-1}} \prod_{k=l}^{L-1} \alpha_k\right) = \frac{1}{m_0}\sum_{l=1}^{L} \mathbb{V}\mathrm{ar}\Big(Y_{l,\alpha_{l-1}}\Big) \prod_{k=l}^{L-1} \alpha_k^2$$

with, $\forall l \in [\![1, L]\!]$

$$\mathbb{V}\mathrm{ar}\Big(Y_{l,\alpha_{l-1}}\Big) = \mathbb{V}\mathrm{ar}(X_l) + \alpha_{l-1}^2\,\mathbb{V}\mathrm{ar}(X_{l-1}) - 2\alpha_{l-1}\mathbb{C}\mathrm{ov}(X_l, X_{l-1}).$$

Let $l \in [\![1, L-1]\!]$; then

$$\begin{aligned}\frac{\partial v}{\partial \alpha_l}(\boldsymbol{\alpha}) &= \frac{2}{m_0}\alpha_l\left(\sum_{i=1}^{l} \mathbb{V}\mathrm{ar}\Big(Y_{i,\alpha_{i-1}}\Big) \prod_{j=0}^{i-1} \alpha_j\right) \prod_{k=l+1}^{L-1} \alpha_k^2 + \frac{2}{m_0}\alpha_l\,\mathbb{V}\mathrm{ar}(X_l) \prod_{k=l+1}^{L-1} \alpha_k^2 \\ &\qquad - \frac{2}{m_0}\mathbb{C}\mathrm{ov}(X_{l+1}, X_l) \prod_{k=l+1}^{L-1} \alpha_k^2 \\ &= \frac{2}{m_0}\left(\alpha_l\left(\mathbb{V}\mathrm{ar}(X_l) + \sum_{i=1}^{l} \mathbb{V}\mathrm{ar}\Big(Y_{i,\alpha_{i-1}}\Big) \prod_{j=0}^{i-1} \alpha_j\right) - \mathbb{C}\mathrm{ov}(X_{l+1}, X_l)\right) \prod_{k=l+1}^{L-1} \alpha_k^2.\end{aligned}$$

Thus

$$\frac{\partial v}{\partial \alpha_l}(\boldsymbol{\alpha}) = 0 \iff \alpha_l = \frac{\mathbb{C}\mathrm{ov}(X_{l+1}, X_l)}{\mathbb{V}\mathrm{ar}(X_l) + \sum_{i=1}^{l} \mathbb{V}\mathrm{ar}\Big(Y_{i,\alpha_{i-1}}\Big) \prod_{j=0}^{i-1} \alpha_j} \tag{A.4}$$

Since the right-hand side of (A.4) depends only on $\{\alpha_j : j \in [\![0, l-1]\!]\}$, $\check{\boldsymbol{\alpha}}$ is explicitly defined by the following recurrence relation[9]:

$$\forall l \in [\![0, L-1]\!], \quad \check{\alpha}_l := \frac{\mathbb{C}\mathrm{ov}(X_{l+1}, X_l)}{\mathbb{V}\mathrm{ar}(X_l) + \sum_{i=1}^{l} \mathbb{V}\mathrm{ar}\Big(Y_{i,\alpha_{i-1}}\Big) \prod_{j=0}^{i-1} \alpha_j}.$$

[9] Let us recall the following conventions: $\sum_{x\in\emptyset} x = 0$; $\prod_{x\in\emptyset} x = 1$.

A.1.2 Optimal sample sizes

Let $\eta \in]0, +\infty[$ and $\boldsymbol{m} \in \mathbb{N}_*^{L+1}$. Assumptions (I) and (II) made in section A.1.1 are withdrawn. We wish to choose the sample sizes $\boldsymbol{m}$ so that the statistical error of the MFMC estimator $\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})$ satisfies the tolerance η for a minimal cost. Therefore, we look for

$$\check{\boldsymbol{m}} := \operatorname{argmin}\left\{ €\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big) : \boldsymbol{m} \in \mathbb{N}_*^{L+1};\ \sqrt{\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big)} \leqslant \eta \right\}. \quad \text{(A.5)}$$

For all $l \in [\![0, L]\!]$, let us note

$$v_l := \mathbb{V}\mathrm{ar}\big(Y_{l,\check{\alpha}_{l-1}}\big) \prod_{k=l}^{L-1} \check{\alpha}_k^2 \quad \text{so that} \quad \mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big) = \sum_{i=0}^{L} \frac{v_i}{m_i}$$

and

$$€_l := €\big(Y_{l,\check{\alpha}_{l-1}}\big) \quad \text{so that} \quad €\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big) = \sum_{l=0}^{L} m_l\, €_l\,.$$

Consequently, to solve the constrained minimisation problem (A.5) we first look for

$$\boldsymbol{r} := \operatorname{argmin}\left\{ \sum_{l=0}^{L} n_l\, €_l : \sum_{i=0}^{L} \frac{v_i}{n_i} \leqslant \eta^2;\ \boldsymbol{n} \in \mathbb{R}_*^{L+1} \right\}.$$

To solve this constrained minimisation problem, we define the following Lagrangian function $\forall (\boldsymbol{n}, \lambda) \in \mathbb{R}^{L+1} \times \mathbb{R}$:

$$J(\boldsymbol{n}, \lambda) := \sum_{l=0}^{L} n_l\, €_l - \lambda \left(\eta^2 - \sum_{l=0}^{L} \frac{v_l}{n_l} \right).$$

Let us look the stationary points of J in $]0, +\infty[^{L+1} \times]0, +\infty[$. Let $(\boldsymbol{r}, \lambda) \in]0, +\infty[^{L+1} \times]0, +\infty[$;

$$\frac{\partial J}{\partial \lambda} J(\boldsymbol{r}, \lambda) = \eta^2 - \sum_{i=0}^{L} \frac{v_i}{r_i};$$

$\forall l \in [\![0, L]\!]$,

$$\frac{\partial J}{\partial n_l}(\boldsymbol{r}, \lambda) = €_l - \lambda \frac{v_l}{r_l^2}.$$

Therefore $(\boldsymbol{r}, \lambda)$ is a stationary point of J if and only if $\forall l \in [\![0, L]\!]$

$$\frac{\partial J}{\partial n_l}(\boldsymbol{r}, \lambda) = 0 \iff r_l = \sqrt{\frac{\lambda v_l}{€_l}},$$

and

$$\frac{\partial J}{\partial \lambda}(\boldsymbol{r},\lambda)=0 \iff \quad \eta^2=\sum_{i=0}^{L}\frac{v_i}{r_i}=\sum_{i=0}^{L}\sqrt{\frac{v_i\,€_i}{\lambda}}$$
$$\iff \quad \sqrt{\lambda}=\eta^{-2}\sum_{i=0}^{L}\sqrt{v_i\,€_i}.$$

Consequently, $\forall l\in[\![0,L]\!]$

$$r_l=\eta^{-2}\sqrt{\frac{v_l}{€_l}}\sum_{i=0}^{L}\sqrt{v_i\,€_i}.$$

Therefore the optimal sample sizes $\check{\boldsymbol{m}}$ as defined by (A.5) are

$$\forall l\in[\![0,L]\!],\quad \check{m}_l=\left\lceil \eta^{-2}\sqrt{\frac{v_l}{€_l}}\sum_{i=0}^{L}\sqrt{v_i\,€_i}\right\rceil. \tag{A.6}$$

Nevertheless, the control coefficients $\check{\boldsymbol{\alpha}}$ do not necessarily yield the lowest variance, due to the simplifying assumption made to derive them. An approach without such simplification will be presented in the next section to address this.

Remark 2 (Fixed budget). The approach above gives cost-minimal, error-constrained samples sizes. A converse reasoning yields similar results for error-minimal, cost-constrained sample sizes. Let $b\in\,]0,+\infty[$ be the computational budget, id est (i.e.) the upper bound on the cost. Solving

$$\hat{\boldsymbol{m}}:=\operatorname{argmin}\left\{\sqrt{\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big)}:\boldsymbol{m}\in\mathbb{N}_*^L;€\big(\mu_{\boldsymbol{m},\check{\boldsymbol{\alpha}}}(\boldsymbol{X})\big)\leqslant b\right\}$$

leads to seeking the stationary points of the Lagrangian function $\tilde{J}$ defined $\forall(\boldsymbol{n},\lambda)\in\mathbb{R}^{L+1}\times\mathbb{R}$ as

$$\tilde{J}(\boldsymbol{n},\lambda):=\sum_{l=0}^{L}\frac{v_l}{n_l}-\lambda\left(b-\sum_{i=0}^{L}n_i\,€_i\right).$$

The resolution follows the same reasoning as above. The final sample sizes are then

$$\forall l\in[\![0,L]\!],\quad \hat{m}_l=\left\lfloor b\sqrt{\frac{v_l}{€_l}}\sum_{i=0}^{L}\sqrt{v_i\,€_i}\right\rfloor.$$

Confer (cf.) (A.6). ◊

A.2 Coupled calibration

Here we seek to optimise jointly the control coefficients and sample sizes, whereas section A.1 derived optimal sample sizes for non-optimal control coefficients. The MFMC estimator is as defined in (A.2) and its variance is

$$\mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) = \sum_{l=0}^{L} \frac{\mathbb{V}\mathrm{ar}\big(Y_{l,\alpha_{l-1}}\big)}{m_l} \prod_{k=l}^{L-1} \alpha_k^2$$

We seek to calibrate the sample sizes $\boldsymbol{m}$ and control coefficients $\boldsymbol{\alpha}$ so as to minimise the cost of this estimator, while satisfying a tolerance $\eta \in]0, +\infty[$ on its statistical error. We define this optimal calibration as

$$(\breve{\boldsymbol{m}}, \breve{\boldsymbol{\alpha}}) := \mathrm{argmin}\big\{€\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(X)\big) : \mathbb{V}\mathrm{ar}\big(\mu_{\boldsymbol{m},\boldsymbol{\alpha}}(\boldsymbol{X})\big) \leqslant \eta^2;\ \boldsymbol{m} \in \mathbb{N}^{L+1};\ \boldsymbol{\alpha} \in \mathbb{R}^{L}\big\} \tag{A.7}$$

For the sake of brevity and legibility, we will use the following short-hand notation[10] $\forall k, l \in [\![0, L]\!]$

$$v_l := \mathbb{V}\mathrm{ar}(X_l), \qquad c_{k,l} := \mathbb{C}\mathrm{ov}(X_k, X_l),$$
$$€_l := €(Y_l), \qquad \rho_{l,k} := \frac{|c_{l,k}|}{\sqrt{v_l v_k}}.$$

Problem (A.7) is not as straightforward to solve as problems (A.3) and (A.5). Since section 3.2.5 makes use of only two models (i.e. $L := 1$), we will present here only the proof for this case. Consequently, we look for

$$(m_0, m_1, \alpha_0) := \mathrm{argmin}\Bigg\{ n_0 €_0 + n_1 €_1 : \frac{v_1 + a^2 v_0 - 2a c_{0,1}}{n_1} + \frac{a^2 v_0}{n_0} \leqslant \eta^2; \\ (n_0, n_1, a) \in]0, +\infty[\times [0, +\infty[\times \mathbb{R} \Bigg\}. \tag{A.8}$$

This leads us to seek the stationary points of the Lagrangian function $\tilde{J}$ defined $\forall (n_0, n_1, a, \lambda) \in \mathbb{R}_* \times \mathbb{R}_* \times \mathbb{R} \times \mathbb{R}$ as

$$J(n_0, n_1, a, \lambda) := n_0 €_0 + n_1 €_1 - \lambda \left(\eta^2 - \frac{v_1 + a^2 v_0 - 2a c_{0,1}}{n_1} + \frac{a^2 v_0}{n_0} \right)$$

[10] Nota bene (NB) the notation v_l is not to be confused with the short-hand notation used in section A.1.

whose partial derivatives are

$$\frac{\partial J}{\partial n_0}(n_0, n_1, a, \lambda) = €_0 - \frac{\lambda a^2 v_0}{n_0^2}; \tag{A.9}$$

$$\frac{\partial J}{\partial n_1} J(n_0, n_1, a, \lambda) = €_1 - \frac{\lambda}{n_1^2}(v_1 + a^2 v_0 - 2ac_{0,1}); \tag{A.10}$$

$$\frac{\partial J}{\partial a} J(n_0, n_1, a, \lambda) = 2\lambda\left(av_0\left(\frac{1}{n_1} + \frac{1}{n_0}\right) - \frac{c_{0,1}}{n_0}\right) \tag{A.11}$$

$$\frac{\partial J}{\partial \lambda} J(n_0, n_1, a, \lambda) = \frac{v_1 + a^2 v_0 - 2ac_{0,1}}{n_1} + \frac{a^2 v_0}{n_0} - \eta^2. \tag{A.12}$$

Let us seek the stationary points of J; for brevity, we will note $r_1 := m_1/m_0$. If $(m_0, m_1, \alpha_0, \lambda) \in]0, +\infty[\times]0, +\infty[\times \mathbb{R} \times]0, +\infty[$ is a stationary point of J, then from (A.11)

$$\alpha_0 = \frac{c_{0,1}}{v_0(1 + r_1)};$$

from (A.9)

$$m_0^2 = \frac{\lambda \alpha_0^2 v_0}{€_0} \implies m_0 = \frac{|c_{0,1}|}{(1 + r_1)}\sqrt{\frac{\lambda}{v_0\, €_0}};$$

from (A.12)

$$\begin{aligned}
\eta^2 &= \frac{v_1}{m_1} + v_0\alpha_0^2\left(\frac{1}{m_0} + \frac{1}{m_1}\right) - 2\alpha_0\frac{c_{0,1}}{m_1} \\
&= \frac{1}{m_0}\left(\frac{v_1}{r_1} + \frac{c_{0,1}^2}{v_0(1 + r_1)^2}\left(\frac{1}{r_1} + 1\right) - \frac{2c_{0,1}^2}{(1 + r_1)r_1 v_0}\right) \\
&= \sqrt{\frac{€_0\, v_0}{\lambda}}\left(\frac{v_1}{|c_{0,1}|}\left(1 + \frac{1}{r_1}\right) + \frac{|c_{0,1}|}{v_0(1 + r_1)}\left(\frac{r_1 + 1}{r_1}\right) - 2\frac{|c_{0,1}|}{r_1 v_0}\right) \\
&= \sqrt{\frac{€_0\, v_0}{\lambda}}\left(\frac{v_1}{|c_{0,1}|}\left(1 + \frac{1}{r_1}\right) - \frac{|c_{0,1}|}{r_1 v_0}\right) \\
&= |c_{0,1}|\sqrt{\frac{€_0}{\lambda v_0}}\frac{\rho_{0,1}^{-2}(r_1 + 1) - 1}{r_1} \\
\iff \sqrt{\lambda} &= \eta^{-2}|c_{0,1}|\sqrt{\frac{€_0}{v_0}}\frac{\rho_{0,1}^{-2}(r_1 + 1) - 1}{r_1}.
\end{aligned}$$

Consequently, the sample size m_0 must satisfy

$$\begin{aligned} m_0 &= \frac{|c_{0,1}|}{(1+r_1)\sqrt{v_0 \mathbb{V}_0}}\sqrt{\lambda} \\ &\geqslant \eta^{-2}|c_{0,1}|^2 \frac{\rho_{0,1}^{-2}(r_1+1)-1}{(1+r_1)r_1 v_0} \\ &= \frac{|c_{0,1}|^2}{\eta^2 r_1 v_0}\left(\rho_{0,1}^{-2} - \frac{1}{r_1+1}\right) \\ &= \frac{v_1}{\eta^2 r_1}\left(1 - \frac{\rho_{0,1}^2}{r_1+1}\right) \end{aligned}$$

Then, from (A.10):

$$\begin{aligned} \frac{m_1^2 \mathbb{V}_1}{\lambda} &= v_1 + \alpha_0^2 v_0 - 2c_{0,1}\alpha_0 \\ &= v_1 + \frac{m_0^2}{\lambda}\mathbb{V}_0 - 2\frac{m_0^2}{\lambda}\mathbb{V}_0(1+r_1) \\ &= v_1 - \frac{1+2r_1}{\lambda} m_0^2 \mathbb{V}_0 \end{aligned}$$

$$\iff \qquad v_1 = \frac{m_0^2}{\lambda}(r_1^2 \mathbb{V}_1 + 2r_1 \mathbb{V}_0 + \mathbb{V}_0)$$

$$\iff \qquad v_1(1+r_1)^2 = \frac{c_{0,1}^2}{v_0 \mathbb{V}_0}(r_1^2 \mathbb{V}_1 + 2r_1 \mathbb{V}_0 + \mathbb{V}_0)$$

$$\iff \qquad r_1^2 + 2r_1 + 1 = \rho_{0,1}^2\left(r_1^2\frac{\mathbb{V}_1}{\mathbb{V}_0} + 2r_1 + 1\right)$$

$$\iff \qquad 0 = r_1^2\left(\rho_{0,1}^2\frac{\mathbb{V}_1}{\mathbb{V}_0} - 1\right) + 2r_1(\rho_{0,1}^2 - 1) + \rho_{0,1}^2 - 1 \tag{A.13}$$

If $\rho_{0,1}^2 \mathbb{V}_1 = \mathbb{V}_0$, the unique solution to (A.13) is $r_1 = -0.5$, which is invalid; let us therefore assume $\rho_{0,1}^2 \mathbb{V}_1 \neq \mathbb{V}_0$.

Then polynomial equation (A.13) is of degree 2 and therefore admits real solutions if and only if

$$\begin{aligned} 0 &\leqslant 4(\rho_{0,1}^2 - 1)^2 - 4(\rho_{0,1}^2 - 1)\left(\rho_{0,1}^2\frac{\mathbb{V}_1}{\mathbb{V}_0} - 1\right) \\ &= 4\rho_{0,1}^2(1-\rho_{0,1}^2)\left(\frac{\mathbb{V}_1}{\mathbb{V}_0} - 1\right) \end{aligned}$$

This inequality is true since – by definitions – we know that $\rho_{0,1} \in [0,1]$ and $0 < €_0 < €_1$. This inequality can be assumed to be strict, since each of the three cases of equality precludes a MFMC approach with these models:

(I) if $\rho_{0,1} = 0$ then the models are uncorrelated;

(II) if $\rho_{0,1} = 1$ then a MC estimation of the least expensive model is more efficient;

(III) if $€_1 = €_0$ then the high-fidelity model X_1 has zero cost.

Therefore, (A.13) admits exactly two distinct real solutions: $r_1 \in \{r_1', r_1''\}$ with

$$r_1' := \frac{1 - \rho_{0,1}^2 - 2\rho_{0,1}\sqrt{(1-\rho_{0,1}^2)\left(\frac{€_1}{€_0} - 1\right)}}{2\left(\rho_{0,1}^2 \frac{€_1}{€_0} - 1\right)}, \tag{A.14a}$$

$$r_1'' := \frac{1 - \rho_{0,1}^2 + 2\rho_{0,1}\sqrt{(1-\rho_{0,1}^2)\left(\frac{€_1}{€_0} - 1\right)}}{2\left(\rho_{0,1}^2 \frac{€_1}{€_0} - 1\right)}. \tag{A.14b}$$

We want the smallest *strictly positive* solution, which leads to the question of their sign:

$$\begin{aligned}
& 1 - \rho_{0,1}^2 \geqslant 2\rho_{0,1}\sqrt{(1-\rho_{0,1}^2)\left(\frac{€_1}{€_0} - 1\right)} \\
\iff \quad & (1-\rho_{0,1}^2)^2 \geqslant 4\rho_{0,1}^2(1-\rho_{0,1}^2)\left(\frac{€_1}{€_0} - 1\right) \\
\iff \quad & 1 - \rho_{0,1}^2 \geqslant 4\rho_{0,1}^2\left(\frac{€_1}{€_0} - 1\right) \\
\iff \quad & 1 \geqslant \rho_{0,1}^2\left(4\frac{€_1}{€_0} - 3\right) \\
\iff \quad & \rho_{0,1}^2 \leqslant \frac{€_0}{4€_1 - 3€_0} \leqslant \frac{€_0}{€_1}
\end{aligned}$$

Consequently, there remain three cases.

(I) $\rho_{0,1}^2 \leqslant \frac{€_0}{4€_1 - 3€_0}$: no strictly positive solution.

(II) $\frac{€_0}{4€_1 - 3€_0} < \rho_{0,1}^2 < \frac{€_0}{€_1}$: $r_1 := r_1'$ is the only strictly positive solution.

(III) $\rho_{0,1}^2 > \frac{€_0}{€_1}$: $r_1 := r_1''$ is the only strictly positive solution.

This leads to assume that

$$\frac{\mathbb{C}\mathrm{ov}(X_0, X_1)^2}{\mathbb{V}\mathrm{ar}(X_0)\,\mathbb{V}\mathrm{ar}(X_1)} > \frac{€(X_0)}{€(X_0) + 4€(X_1)}. \tag{A.15}$$

Let us remark that the value of r_1 becomes infinitely large for $\rho_{0,1}^2$ close to $€_0 / €_1$; moreover, the interval $\left]\frac{€_0}{4€_1 - 3€_0}, \frac{€_0}{€_1}\right[$ is narrow in the general case where $€_0 \ll €_1$. There, a more desirable case – albeit not necessary – would be

$$\frac{\mathbb{C}\mathrm{ov}(X_0, X_1)^2}{\mathbb{V}\mathrm{ar}(X_0)\,\mathbb{V}\mathrm{ar}(X_1)} > \frac{€(X_0)}{€(X_0) + €(X_1)}.$$

We assume (A.15) henceforth and therefore (A.13) has exactly one strictly positive solution.

Since the formulation in $r_1 = m_1/m_0$ excludes '$m_0 = 0$' from the possible solutions, the cost of the resulting MFMC estimator is not guaranteed to be lower than a MC estimator of the same accuracy η. Let us make this comparison now: the sample size n of the MC estimator must verify the same constraint, therefore

$$\mathbb{V}\mathrm{ar}(\mu_n(X_1)) \leqslant \eta^2 \iff n \geqslant \frac{\mathbb{V}\mathrm{ar}(X_1)}{\eta^2}.$$

For simplification, let us make the approximation $€(X_1) \approx €_1$. Then the cost of the tuned MFMC estimator if lower than that of $\mu_n(X_1)$ if and only if

$$\begin{aligned}
& €_0\, m_0 + €_1\, m_0 r_1 \leqslant €(X_1)\frac{\mathbb{V}\mathrm{ar}(X_1)}{\eta^2} \approx €_1\,\frac{\mathbb{V}\mathrm{ar}(X_1)}{\eta^2} \\
\iff \quad & \left(1 - \frac{\rho_{0,1}^2}{r_1 + 1}\right)\left(\frac{€_0}{r} + €_1\right) \leqslant €_1 \\
\iff \quad & (r_1 + 1 - \rho_{0,1}^2)(€_0 + r_1\, €_1) \leqslant r_1(r_1 + 1)\, €_1 \\
\iff \quad & r_1(€_0 - \rho_{0,1}^2\, €_1) \leqslant -(1 - \rho_{0,1}^2)\, €_0\,.
\end{aligned}$$

This last inequality can be satisfied with $r_1 > 0$ if and only if $\rho_{0,1}^2 > €_0 / €_1$. Therefore, this is a necessary condition for the MFMC estimator such as calibrated here to have a lower cost than a MC estimator of the same accuracy. This leads us to formulate assumption 2 p. 47, and means that only r_1'' is an acceptable value for r_1. This assumption is not exactly sufficient because of the approximation $€(X_1) \approx €_1$, whereas in reality $€(X_1) < €_1$.

Finally, the solution to (A.8) is

$$\breve{m}_0 = \left\lceil \frac{v_1}{\eta^2 r_1''}\left(1 - \frac{\rho_{0,1}^2}{r_1'' + 1}\right)\right\rceil ; \qquad \breve{m}_1 = \left\lceil \frac{v_1}{\eta^2}\left(1 - \frac{\rho_{0,1}^2}{r_1'' + 1}\right)\right\rceil ;$$
$$\breve{\alpha}_0 = \frac{c_{0,1}}{v_0(1 + r_1'')} ;$$

with r_1'' as expressed in (A.14b) and under assumption 2.

List of Algorithms

List of Figures

List of Tables

Acronyms

API application programming interface

CAARC Commonwealth advisory aeronautical council

CFD computational fluid dynamics

CDF cumulative distribution function

HPC high-performance computing

MC Monte Carlo

MFMC multi-fidelity Monte Carlo

MLMC multi-level Monte Carlo

MPI message passing interface

MSE mean squared error

UQ uncertainty quantification

OUU optimisation under uncertainties

PDE partial differential equation

ODE ordinary differential equation

SDE stochastic differential equation

PDF probability density function

QoI quantity of interest

VaR value at risk

CVaR conditional value at risk

KDE kernel density estimation

CMLMC continuation multi-level Monte Carlo

Abbreviations

i.e. id est

e.g. exempli gratia

a.e. almost every

a.e. almost everywhere

a.s. almost surely

i.i.d. independent and identically-distributed

s.t. such that

iff. if and only if

ibid. ibidem

et al. et alii

etc. et cætera

cf. confer

viz. videlicet

NB nota bene