Exascale Quantification of Uncertainties for Technology and Science Simulation

D7.3 Report on UQ results and overall user experience

Document information table

Contract number:	800898
Project acronym:	ExaQUte
Project Coordinator:	CIMNE
Document Responsible Partner:	str.ucture
Deliverable Type:	REPORT
Dissemination Level:	Private
Related WP & Task:	WP7, Task 7.4 & 7.6
Status:	Final version

Authoring

Prepared by:				
Authors	Partner	Modified Page/Sections	Version	Comments
Sami Bidier	str.ucture	§ 1, § 4, § 5	1.0	
Ustim Khristenko	TUM	§ 2	1.0	
Riccardo Tosi	CIMNE	§ 1, § 3	1.0	

Change Log

Versions	Modified Page/Sections	Comments
V1.0	First version	

Approval

Approved by:				
	Name	Partner	Date	OK
Task leader	Roland Wüchner	TUM		
WP leader	Alexander Michalski	str.ucture	25.02.21	✓
Coordinator	Riccardo Rossi	CIMNE	26.02.21	✓

Executive summary

This deliverable report focuses on the main Uncertainty Quantification (UQ) results obtained within the EXAscale Quantification of Uncertainties for Technology and Science Simulation (ExaQUte) project. Details on the turbulent wind inlet generator, that enables the supply of random, yet steady, wind velocity boundary conditions during run-time, are given in section 2. This enables the developed UQ workflow, whose results are presented on the basis of the Commonwealth Advisory Aeronautical Council (CAARC) as described in Deliverable 7.1 [27]. Finally, the completed UQ workflow and the results are evaluated from an application-driven wind engineering point of view. Thereby, the significance of the developed methods and the obtained results are discussed and their applicability in practical wind-engineering applications is tested through a complete test-run of the UQ workflow.

Contents

List of Figures

Nomenclature / Acronym list

Acronym	Meaning
CAARC	Commonwealth Advisory Aeronautical Council
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
CVaR	Conditional Value at Risk
CWE	Computational Wind Engineering
DE	Discretization Error
ExaQUte	EXAscale Quantification of Uncertainties for Technology and Science Simulation
HPC	High Performance Computing
Kratos	Kratos Multiphysics
MC	Monte Carlo
MLMC	Multilevel Monte Carlo
pdf	probability density function
QoI	Quantity of Interest
SE	Statistical Error
UQ	Uncertainty Quantification
VaR	Value at Risk

1 Introduction

Computational Wind Engineering (CWE) is the research field which studies the effects of wind on structures, such as long span bridges or tall buildings. Traditionally, wind tunnels were exploited to perform experiments. However, the development of High Performance Computing (HPC) resources and computational fluid dynamics (CFD) capabilities of the last decades makes possible to apply numerical techniques to simulate the wind effect on buildings. Due to a vast number of highly uncertain parameters in the complex field of wind engineering, such as the turbulence intensity of the wind profile, the wind profile itself, varying wind directions or surface conditions, to name only a few, classical deterministic strategies are often limited in their trustworthy description of the problem at hand. In this regard, UQ treats and describes the propagation of uncertain fields in computational investigations. In the following sections, the UQ results within the ExaQUte project are demonstrated and reported. Furthermore, the applicability and significance of the developed methods to engineering applications is studied and commented on.

2 Mann model and the on-the-fly wind generator

2.1 Random field model for the velocity fluctuations

Let us consider a free space turbulent velocity field $\mathbf{u}(\boldsymbol{x}) = \langle \mathbf{u}(\boldsymbol{x}) \rangle + \tilde{\mathbf{u}}(\boldsymbol{x}) \in \mathbb{R}^3$, $\boldsymbol{x} \in \mathbb{R}^3$, where $\langle \mathbf{u} \rangle$ is the mean velocity field and $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \tilde{u}_3)$ corresponds to the zero-mean turbulent fluctuations. A common approach is to assume the model for $\tilde{\mathbf{u}}$ to be *Gaussian* random field. In this case, it is entirely determined from its two-point correlation tensor

$$R_{ij}(\boldsymbol{r}, \boldsymbol{x}, t) = \langle u_i(\boldsymbol{x}, t) u_j(\boldsymbol{x} + \boldsymbol{r}, t) \rangle.$$

When $R(\boldsymbol{r}, \boldsymbol{x}, t) = R(\boldsymbol{r}, t)$ depends only on the separation vector $\boldsymbol{r}$, the model is said to be spatially *homogeneous*. Alternatively, when $R(\boldsymbol{r}, \boldsymbol{x}, t) = R(\boldsymbol{r}, \boldsymbol{x})$ is independent of the time variable t, the model is said to be temporally *stationary*.

Frequently, it is convenient to consider the Fourier transform of the field $\tilde{\mathbf{u}}$. In such cases, we express the field in terms of a generalized Fourier–Stieltjes integral,

$$\tilde{\mathbf{u}}(\boldsymbol{x}) = \int_{\mathbb{R}^3} \mathrm{e}^{\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x}} \, \mathrm{d}\boldsymbol{Z}(\boldsymbol{k}) \,, \tag{1}$$

where $\boldsymbol{Z}(\boldsymbol{k})$ is a three-component measure on $\mathbb{R}^3$, and $\boldsymbol{k} = (k_1, k_2, k_3)$ is the wavenumber vector. The validity of this expression follows from the Wiener–Khinchin theorem [20]. Likewise, in the homogeneous setting, we may consider the Fourier transform of the covariance tensor, otherwise known as the *velocity-spectrum tensor*,

$$\Phi_{ij}(\boldsymbol{k}) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \mathrm{e}^{-\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{r}} \, R_{ij}(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}.$$

Consider three-dimensional additive white Gaussian noise [14, 19] in the physical and frequency domains, denoted $\boldsymbol{\xi}(\boldsymbol{x})$ and $\hat{\boldsymbol{\xi}}(\boldsymbol{k})$, respectively, such that

$$\boldsymbol{\xi}(\boldsymbol{x}) = \int_{\mathbb{R}^3} \mathrm{e}^{\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x}} \, \hat{\boldsymbol{\xi}}(\boldsymbol{k}) \, \mathrm{d}\boldsymbol{k} = \int_{\mathbb{R}^3} \mathrm{e}^{\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x}} \, \mathrm{d}\boldsymbol{W}(\boldsymbol{k}),$$

where $\boldsymbol{W}(\boldsymbol{k})$ is three-dimensional Brownian motion. We assume $\mathrm{d}\boldsymbol{Z}(\boldsymbol{k}) = \boldsymbol{G}(\boldsymbol{k})\,\mathrm{d}\boldsymbol{W}(\boldsymbol{k}) = \boldsymbol{G}(\boldsymbol{k})\hat{\boldsymbol{\xi}}(\boldsymbol{k})\,\mathrm{d}\boldsymbol{k}$, where $\boldsymbol{G}(\boldsymbol{k})^*\boldsymbol{G}(\boldsymbol{k}) = \Phi(\boldsymbol{k})$.

A standard form of the spectral tensor used in isotropic stationary and homogeneous turbulence models is

$$\Phi_{ij}(\boldsymbol{k}) = (4\pi)^{-1}k^{-2}E(k)P_{ij}(\boldsymbol{k})\,. \tag{2}$$

where $k = |\boldsymbol{k}|$ is the magnitude of the wavenumber vector, $E(k)$ is called the *energy spectrum function* and $P_{ij}(\boldsymbol{k}) = \delta_{ij} - \frac{k_i k_j}{k^2}$ is commonly referred to as the *projection tensor*. One common empirical model for $E(k)$, suggested by [35], is given by the expression

$$E(k) = c_0^2\,\varepsilon^{2/3}k^{-5/3}\left(\frac{kL}{(1+(kL)^2)^{1/2}}\right)^{17/3}. \tag{3}$$

Here, ε is the viscous dissipation of the turbulent kinetic energy, L is a length scale parameter, and $c_0^2 \approx 1.7$ is an empirical constant.

2.2 Shear flow and Mann's model

Given the velocity field $\mathbf{u} = \langle\mathbf{u}\rangle + \tilde{\mathbf{u}}$, let us define the average total derivative of the turbulent fluctuations $\tilde{\mathbf{u}}$ as follows:

$$\frac{\bar{D}\tilde{u}_i}{\bar{D}t} = \frac{\partial\tilde{u}_i}{\partial t} + \langle u_j\rangle\frac{\partial\tilde{u}_i}{\partial x_j}\,.$$

The rapid distortion equations [see, e.g., 17, 26, 33] are a linearization of the Navier–Stokes equations in free space when the turbulence-to-mean-shear time scale ratio is arbitrarily large. They can be written

$$\frac{\bar{D}\tilde{u}_i}{\bar{D}t} = -\tilde{u}_i\frac{\partial\langle u_j\rangle}{\partial x_i} - \frac{1}{\rho}\frac{\partial p}{\partial x_i}, \qquad \frac{1}{\rho}\Delta p = -2\frac{\partial\langle u_i\rangle}{\partial x_j}\frac{\partial\tilde{u}_j}{\partial x_i}\,, \tag{4}$$

where ρ and p stand for the mass density and the hydrostatic pressure respectively.

Under a uniform shear mean velocity gradient, $\langle u_i(\boldsymbol{x})\rangle = x_j\partial\langle u_i\rangle/\partial x_j$, where $\partial\langle u_i\rangle/\partial x_j$ is a constant tensor, a well-known form of these equations can be written out in Fourier space. In this case, the rate of change of each frequency $\boldsymbol{k}(t) = (k_1(t), k_2(t), k_3(t))$ is defined $\mathrm{d}k_i/\mathrm{d}t = -k_j\partial\langle u_j\rangle/\partial x_i$. We then have the following Fourier representation of the average total derivative of $\tilde{\mathbf{u}}$:

$$\frac{\bar{D}\tilde{u}_i}{\bar{D}t} = \int_{\mathbb{R}^3}\mathrm{e}^{\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x}}\left(\left(\frac{\partial}{\partial t} + \frac{\mathrm{d}k_j}{\mathrm{d}t}\frac{\partial}{\partial k_j}\right)\mathrm{d}Z_i(\boldsymbol{k},t)\right) = \int_{\mathbb{R}^3}\mathrm{e}^{\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x}}\left(\frac{\bar{D}\,\mathrm{d}Z_i(\boldsymbol{k},t)}{\bar{D}t}\right).$$

With this expression, the Fourier representation of (4) can be written

$$\frac{\bar{D}\,\mathrm{d}Z_j(\boldsymbol{k},t)}{\bar{D}t} = \frac{\partial u_\ell}{\partial x_k}\left(2\frac{k_j k_\ell}{k^2} - \delta_{j\ell}\right)\mathrm{d}Z_k(\boldsymbol{k},t)\,. \tag{5}$$

Exact solutions to (5) are well-known [see, e.g., 22, 33], given the initial conditions $\boldsymbol{k}_0 = (k_{10}, k_{20}, k_{30})$ and $\mathrm{d}\mathbf{Z}(\boldsymbol{k}_0, 0)$. In the scenario

$$\langle\mathbf{u}(\boldsymbol{x})\rangle = (u_0 + Sx_3)\boldsymbol{e}_1,$$

the solution can be written in terms of the evolving Fourier modes $\boldsymbol{k}(t)$ and non-dimensional time $\tau = St$, as follows:

$$\mathrm{d}\mathbf{Z}(\boldsymbol{k}, t) = \boldsymbol{D}_\tau(\boldsymbol{k})\, \mathrm{d}\mathbf{Z}(\boldsymbol{k}_0, 0),$$

where

$$\boldsymbol{D}_\tau(\boldsymbol{k}) = \begin{bmatrix} 1 & 0 & \zeta_1 \\ 0 & 1 & \zeta_2 \\ 0 & 0 & \zeta_3 \end{bmatrix}, \qquad \boldsymbol{k}_0 = \boldsymbol{T}_\tau \boldsymbol{k}, \qquad \boldsymbol{T}_\tau = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \tau & 0 & 1 \end{bmatrix}.$$

In the expression for $\boldsymbol{D}_\tau(\boldsymbol{k})$, the non-dimensional coefficients $\zeta_i = \zeta_i(\boldsymbol{k}, \tau)$, $i = 1, 2, 3$, are defined

$$\zeta_1 = C_1 - C_2 k_2/k_1, \quad \zeta_2 = C_1 k_2/k_1 + C_2, \quad \zeta_3 = k_0^2/k^2,$$

where $k_0 = |\boldsymbol{k}_0|$ and

$$C_1 = \frac{\tau k_1^2 (k_0^2 - 2k_{30}^2 + \tau k_1 k_{30})}{k^2 (k_1^2 + k_2^2)}, \qquad C_2 = \frac{k_2 k_0^2}{(k_1^2 + k_2^2)^{3/2}} \arctan\left(\frac{\tau k_1 (k_1^2 + k_2^2)^{1/2}}{k_0^2 - \tau k_{30} k_1} \right).$$

An important extension of the rapid distortion model involves replacing the constant distortion time τ by a wavenumber-dependent "eddy lifetime" $\tau(k)$; see, e.g., [22]. Such models are considered more realistic because, at some point, the shear from the mean velocity gradient will cause the eddies to stretch and eventually they will breakup within a size-dependent timescale. A widely used example is Mann's model [22], where the eddy lifetime τ is given by the formula

$$\tau(k) = \Gamma S^{-1} (kL)^{-2/3} \left[{}_2F_1(1/3, 17/6; 4/3; -(kL)^{-2}) \right]^{-1/2}, \tag{6}$$

where ${}_2F_1(a, b; c; x)$ is the hypergeometric function, and Γ is a parameter to be determined.

2.3 The on-the-fly wind generator

The representation (1) of the turbulent fluctuations $\tilde{\mathbf{u}}$ can be formally written as a convolution with Gaussian noise $\boldsymbol{\xi}$ in the physical domain:

$$\tilde{\mathbf{u}} = \mathcal{F}^{-1} \mathcal{G} \hat{\boldsymbol{\xi}} = \mathcal{F}^{-1} \mathcal{G} \mathcal{F} \boldsymbol{\xi}, \tag{7}$$

where $\mathcal{F}$ stands for the Fourier transform, the operator $\mathcal{G}$ corresponds to point-wise multiplication by $\boldsymbol{G}(\boldsymbol{k})$, the "square root" of the spectral tensor, such that $\boldsymbol{G}(\boldsymbol{k})^* \boldsymbol{G}(\boldsymbol{k}) = \Phi(\boldsymbol{k})$.

Experience shows that solving all-at-once for a contiguous section of synthetic wind, long enough to be fed into a complete CFD simulation, can be very costly. First of all, the cost of solving (7) scales at best linearly with the domain size. Second of all, storing the solution data may take up an necessary amount of computer resources. In fact, for a 600s CFD simulation, the entire synthetic wind field may require more than 10GB to store.

It turns out the much of this computational cost can be avoided simply by piecing together the random field block-by-block. This technique can easily be combined with inlet condition generation; see, e.g., Figure 1. In that setting, it allows for on-the-fly synthetic wind generation, since only a small block of wind pertaining to the given time step needs to be generated in order to progress the simulation. First, we explain how to perform the block-by-block generation, then we explain why it works.

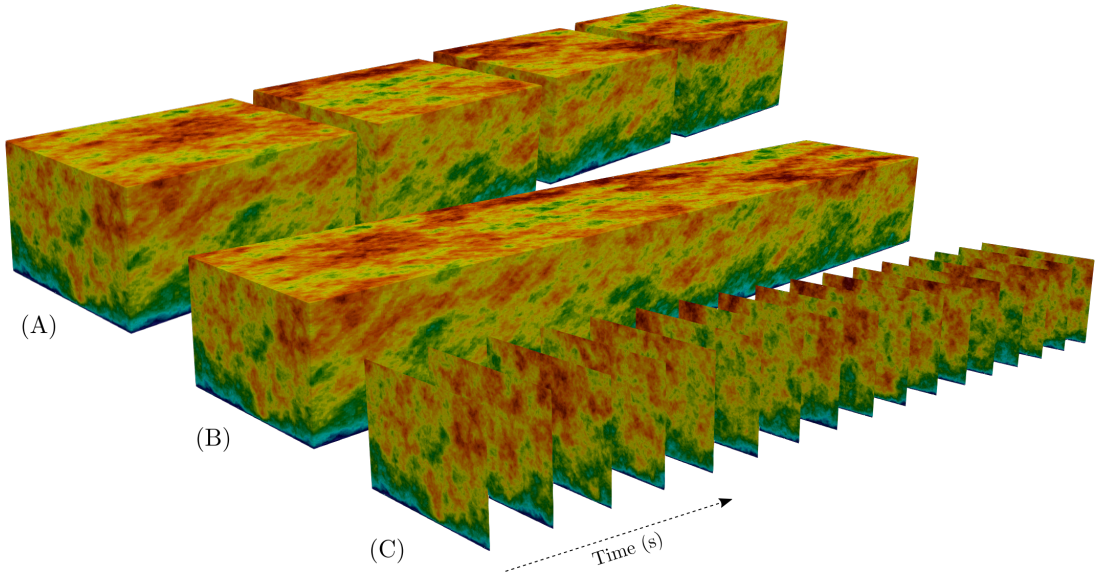

Figure 1: (A) Block-by-block generated wind. (B) All-at-once generated wind. (C) Inlet snapshots.

The technique proceeds as follows. Begin with a subdomain of the final wind field computation domain. This block (i.e., subdomain), should be large enough to contain non-overlapping buffer regions of the same size (e.g., three correlation lengths) as would be used for generating the contiguous field; cf. Figure 2. After seeding the load with additive white Gaussian noise, the resulting wind field, outside the buffer regions, will be accurately rendered.

In order to generate matching wind in a neighboring (but overlapping) block, it is required that the interface of the new, leftmost buffer region (cf. the dotted lines in 2) align with the opposite side of the old, rightmost buffer region interface and that common Gaussian noise be used in the overlap of the two regions, in both overlapping blocks. After seeding the remainder of the new block with new additive white Gaussian noise and solving (7), one arrives at a new block of wind matching its neighbor, up to their common interface, outside each others' buffer regions. Once the buffer regions are discarded, the remaining sections of wind field can be grouped together to form a large contiguous wind field, or processed into snapshots and fed into a CFD simulation; cf. 1 (C).

To understand why this technique works, note that the common noise in the overlap forces the two wind fields to match at their buffer region interfaces, at least up to the accuracy allowed by the size of the buffers. Moving away from the interface, the wind field changes at a fixed regularity, with the influence of the common overlap steadily diminishing.

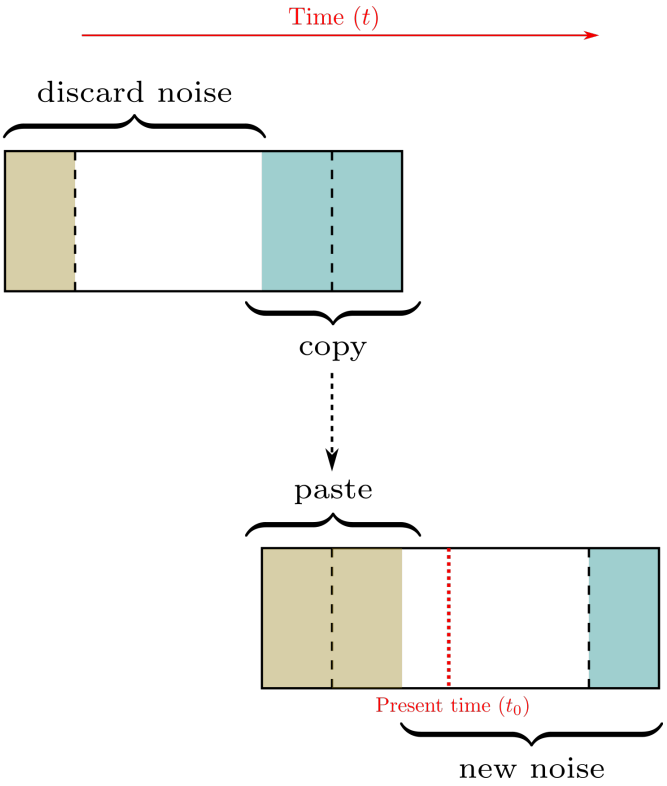

Figure 2: Technique to generate a contiguous wind field block-by-block.

3 UQ Demonstrator - CAARC

It is common knowledge that wind is a variable whose nature is intrinsically stochastic. Therefore, the effect of wind on structures cannot be exactly known, but should be predicted through specific methods and algorithms. The field of science which studies the propagation of uncertainties in computational and real-world problems is called UQ. In this section, we study the effect of wind on the CAARC building, which is a standard benchmark of CWE. We refer to [10, 15, 16, 28] for details about the CAARC building.

3.1 Software and HPC platform

Kratos Multiphysics (Kratos) [12, 13] is the solver software used to solve the CFD problem. Comprehensive comparisons of the solver with experimental and computational CWE results present in literature can be found in [3]. XMC [6] is used as hierarchical Monte Carlo (MC) library and PyCOMPSs [9, 21, 30] and HyperLoom [11] are the programming model for distributed computing. The integration between the different software has been an important part of the ExaQUte project, as reported in [1, 2, 4, 5, 31].

The codes and the specific example have already been presented in deliverable [5] and released in [24, 25]. The results we present next have also been presented in [32].

The results shown in this section have been obtained using the MareNostrum IV Supercomputer, located at the Barcelona Supercomputing Center. Its current peak performance is 11.15 Petaflops, ten times more than its previous version, MareNostrum III. The supercomputer is composed by 3,456 nodes, each of them with two Intel R®Xeon Platinum 8160 (24 cores at 2,1 GHz each). It has 384.75 TB of main memory, 100Gb In-

tel R®Omni-Path Full-Fat Tree Interconnection, and 14 PB of shared disk storage managed by the Global Parallel File System.

3.2 Problem formulation

The wind flow past the CAARC building can be model with the incompressible Navier-Stokes equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \quad & \text{on } \Omega\,,\ t\in[0,T] \\ \nabla\cdot\mathbf{u} = 0 \quad & \text{on } \Omega\,,\ t\in[0,T]\,, \end{aligned} \tag{8}$$

where $\mathbf{u}$ is the velocity, p is the pressure, ν is the kinematic viscosity, $\mathbf{f}$ is the body forces, Ω is the domain and $[0,T]$ is the time window.

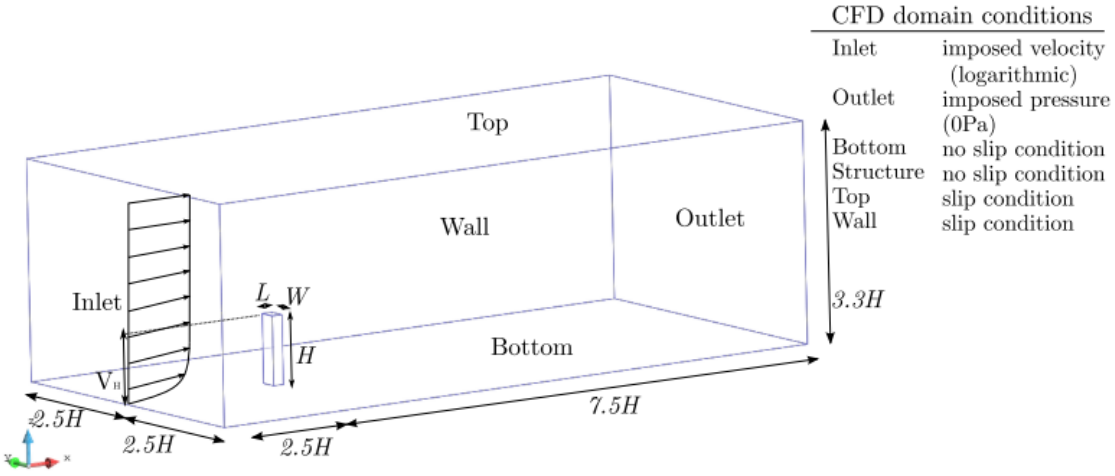

Figure 3: CAARC problem domain. $H = 180\,\text{m}$, $W = 45\,\text{m}$ and $L = 30\,\text{m}$.

The problem domain is presented in figure 3. The wind velocity mean profile is logarithmic and modeled as in [23], with a $40\,\text{m}\,\text{s}^{-1}$ velocity at the reference height of 180 m. The roughness height is 2.0 m, which is typical of centers of very large cities [18]. For this reason, the mean profile is kept constant and only wind fluctuations are considered to be stochastic. The Reynolds number is of the order of 10^8, computed with a characteristic length of 45 m and air density and viscosity.

The original idea was to solve the stochastic problem using the Multilevel Monte Carlo (MLMC) or the Continuation MLMC methods. However, we could not satisfy the hypotheses of these algorithms (see [7, 8]). For this reason, we apply the MC method to solve the problem. Specifically, the algorithm we exploit to solve the problem is the asynchronous MC, which has been developed within the project to run hierarchical MC methods in distributed computing. A simple introduction to the algorithm can be found here [2].

The Monte Carlo estimator for the expected value is

$$\mathbb{E}^{MC}[\mathrm{Q}_H] = \frac{\sum_{n=1}^{N} \mathrm{Q}_H(w^{(n)})}{N}\,, \tag{9}$$

where Q is the Quantity of Interest (QoI) we are interested in, N is the total number of MC realizations, w is the wind fluctuations random variable and H is a discretization parameter of the domain.

Convergence of the asynchronous MC algorithm is checked through the failure probability criteria, which reads

$$\mathbb{P}\left(\left|\mathbb{E}^{MC}[\mathrm{Q}_H] - \mathbb{E}[\mathrm{Q}]\right| \geq \varepsilon\right) \leq \phi\,. \tag{10}$$

In equation 10, $\varepsilon > 0$ is the absolute tolerance of the difference between the sampled estimator $\mathbb{E}^{MC}[\mathrm{Q}_H]$ and the true estimator $\mathbb{E}[\mathrm{Q}]$. $1-\phi \in (0,1)$ is the confidence on the the final statistical estimator. Equation 10 is normally evaluated through the evaluation of the Discretization Error (DE) and of the Statistical Error (SE) [29]. However, for single-level MC, the computation of the DE is not possible for our problem, due to the lack of an analytical solution. Therefore, convergence of the algorithm is evaluated only assessing the SE. Then, equation 10 becomes

$$\mathcal{C}_\phi \mathrm{SE} < \varepsilon\,, \tag{11}$$

where $\mathcal{C}_\phi$ is a confidence coefficient. We refer to [29] for details.

The QoI for which we assess convergence is the time-averaged drag force $\langle F_d \rangle_{T_{bt},T}$, averaged over the effective time window $[T_{bt}, T]$. The burn-in time T_{bt} is the time history we discard to remove the bias of initial conditions, and we set it to 32 s. The confidence $1-\phi$ is 99% and the tolerance ε is 65000, whose corresponding relative tolerance with respect to the time-averaged drag force expected value is of the order of 0.7%.

Other observables are computed: the drag force F_d, the base moment M_b and the time-averaged base moment $\langle M_b \rangle_{T_{bt},T}$, the pressure field on the building p and the time-averaged pressure field on the building $\langle p \rangle_{T_{bt},T}$. International units are used to measure quantities.

3.3 Results

The stochastic problem is solved for a number of wind realizations $N = 126$ and an effective time window $T - T_{bt} = 300\,\mathrm{s}$. The final error $\mathcal{C}_\phi$SE we obtain is 52755, whose relative value $\frac{\mathcal{C}_\phi \mathrm{SE}}{\mathbb{E}^{MC}[\langle F_d \rangle_{T_{bt},T}]}$ is smaller than 1%.

In table 1 we report the sampled expected value and the sampled standard deviation of $\langle F_d \rangle_{T_{bt},T}$, $\langle M_b \rangle_{T_{bt},T}$, F_d and M_b. The standard deviation of a variable Q_H is calculated as

$$\sigma[\mathrm{Q}_H] \approx \sigma^{MC}[\mathrm{Q}_H] = \sqrt{\frac{\sum_{n=1}^{N} (\mathrm{Q}_H(w^{(n)}) - \mathbb{E}^{MC}[\mathrm{Q}_H])^2}{N-1}}\,. \tag{12}$$

We can observe that, as expected, time-averaged quantities present the same expected value as their corresponding standard values. On the other hand, standard deviation values are smaller for time-averaged quantities, since the intermediate time-averaging process damps oscillations and peaks.

Similar conclusions can be drawn for the time-averaged pressure field $\langle p(x) \rangle_{T_{bt},T}$ and the pressure field $p(x)$. Observing figures 4 and 5, we can readily observe that the mean values are the same, while standard deviations are not.

It is known that a risk measure as $\mathbb{E}[\mathrm{Q}] \pm \sigma[\mathrm{Q}]$ is not optimal for many physical variables, especially when Q presents a non-symmetric probability density function (pdf) [34].

	$\mathbb{E}[\cdot]$	$\sigma[\cdot]$
$\langle F_d \rangle_{T_{bt},T}$	9417766	254555
$\langle M_b \rangle_{T_{bt},T}$	-36001	720874
F_d	9417766	2365324
M_b	-36001	9434439

Table 1: Expected value and standard deviation of time-averaged drag force $\langle F_d \rangle_{T_{bt},T}$, time-averaged base moment $\langle M_b \rangle_{T_{bt},T}$, drag force F_d and base moment M_b.

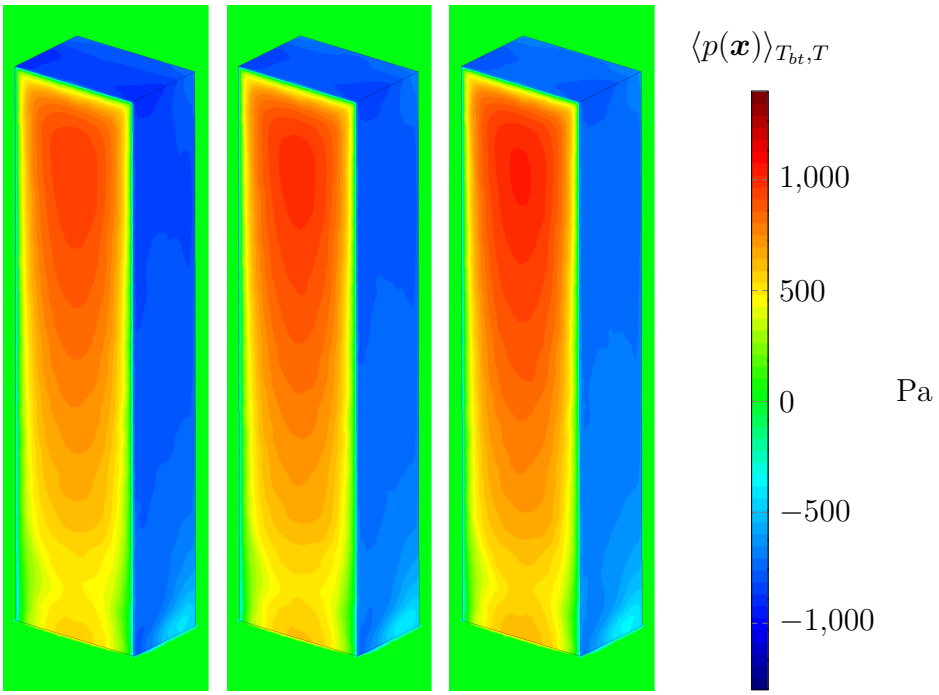

Figure 4: Statistical result of the time-averaged pressure field $\langle p(x) \rangle_{T_{bt},T}$. From left to right, $\mathbb{E}[\langle p(x) \rangle_{T_{bt},T}] - \sigma[\langle p(x) \rangle_{T_{bt},T}]$, $\mathbb{E}[\langle p(x) \rangle_{T_{bt},T}]$ and $\mathbb{E}[\langle p(x) \rangle_{T_{bt},T}] + \sigma[\langle p(x) \rangle_{T_{bt},T}]$.

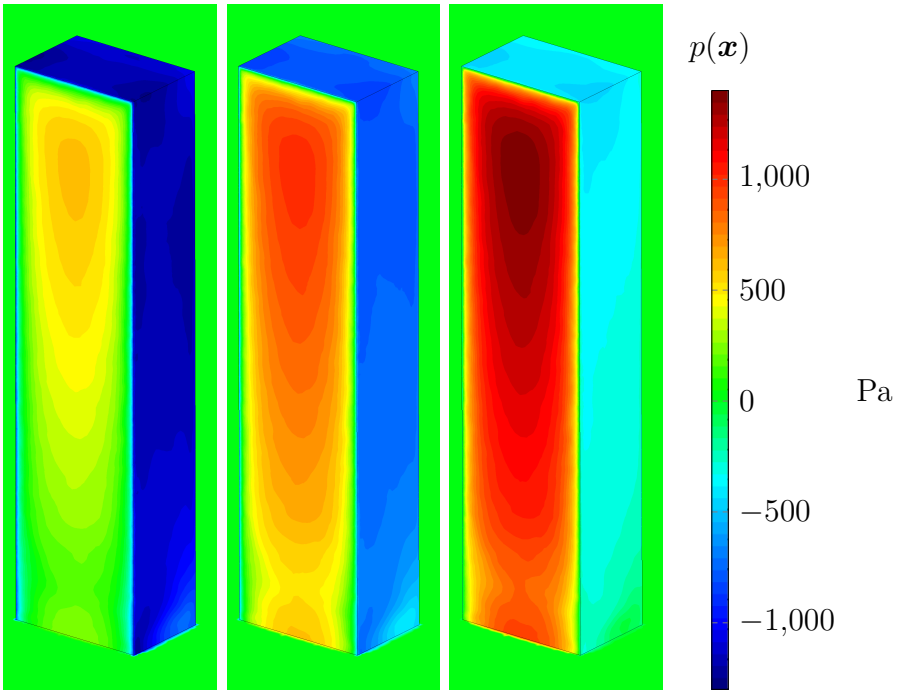

Figure 5: Statistical result of the pressure field $p(x)$. From left to right, $\mathbb{E}[p(x)] - \sigma[p(x)]$, $\mathbb{E}[p(x)]$ and $\mathbb{E}[p(x)] + \sigma[p(x)]$.

For this reason, it is interesting to estimate other statistical estimators as the Value at Risk (VaR) or the Conditional Value at Risk (CVaR). The CVaR is computed as

$$\text{CVaR}_\alpha(\text{Q}) = \frac{1}{1-\alpha}\int_\alpha^1 q_\beta(\text{Q})d\beta\,, \tag{13}$$

where $q_\alpha(\text{Q})$ is the α-quantile of a random variable Q [34]. We report in table 2 the CVaR results for the time-averaged drag force and the drag force. Once more, we observe the difference between standard physical quantities and their physical counterparts.

	CVaR	α
$\langle F_d \rangle$	9845804	0.9
F_d	13871810	0.9

Table 2: CVaR analysis of time-averaged drag force $\langle F_d \rangle_{T_{bt},T}$ and drag force F_d. Results for $N = 126$, $T - T_{bt} = 300\,\text{s}$ and $\alpha = 0.9$.

4 Significance of UQ results for wind engineering applications

Up to date, the scale of industrial applications of CFD simulations with highly uncertain parameters, such as those presented in Sections 2 and 3, is still very small. This is mainly due to two factors. Firstly, the amount of computational resources necessary to conduct

multiple simulations for one problem in a certain project is limited. Secondly, and most important, the modelling infrastructure to rapidly set-up and conduct these simulations in the short time frame of an industrial application project is generally not available. Conflicting to these difficulties stands the frequently formulated client request to estimate the accuracy and to quantify the error of the conducted CFD simulations. From the industrial application point of view, the UQ results presented here thus greatly contribute to more conclusive CFD simulation applications. In the following the significance of the UQ results are specifically assessed.

- **On-the-fly wind generator:** The generation of turbulent wind inlet data is a crucial step in civil engineering applications of CFD methods. The developed method to integrate the generation of the uncertain wind data directly during the run represent a large improvement compared to currently applied wind generation methods. As already mentioned in Section 2, the generation of the turbulent wind data prior to the actual CFD run is not only demanding in terms of storage, but also implies the knowledge of the sufficient total simulation time that is needed for trustworthy results in terms of UQ. This usually requires extensive user experience. In this regard, the possibility to generate varying turbulent wind data in a distributed manner for UQ runs combined with the in-silico evaluation of statistical results offers huge potential for optimised time- and data-storage saving CWE applications, also for non-experienced users.

- **UQ demonstrators using MC methods:** By means of the exemplary CAARC-related MC simulations, the presented UQ demonstrator shows the effectiveness and advantages of the developed workflow. The currently employed CFD simulation workflow, for example at str.ucture GmbH, for general wind engineering problems consists of multiple deterministic CFD runs with varying or randomly changing initial conditions and a subsequent statistical evaluation of the results in order to assess the highly uncertain conditions. This is generally done in a semi-automated parametric workflow. In contrast to this classical approach, the developed workflow within ExaQUte employs the usage of the PyCOMPSs and the XMC environments to steer, realise and evaluate a large number of simulation realisations based on a single modelling set-up in the same order of time. Despite the larger computational costs, this workflow is highly effective. Furthermore, and most important, it additionally supplies the user with a confidence criteria and a statistical error estimation with respect to the uncertain variables of the defined problem. To the authors best knowledge, making this uncertainty quantification available to end users of real wind engineering problems is unprecedented and presents an import step towards the broader acceptance and trustworthiness of computer-aided wind engineering. In this regard, it should be noted that the confidence on the statistical estimator and thus the confidence coefficient $\mathcal{C}_\phi$ given in (11) and the absolute tolerance remain as the critical parameters that should be discussed and determined ideally in a dialogue between the applying engineer and the respective client at hand.

5 Overall user experience and evaluation

In order to evaluate the user experience of the ExaQUte UQ workflow, all developed software and Computer Aided Engineering (CAE) packages were supplied to str.ucture

GmbH. An additional UQ example was set-up to test the usability of the developed methods for industrial wind engineering applications. Therefore, local installations of the packages on the in-house cluster were done and subsequently used, as well as existing installations on the project partner's (IT4I) HPC cluster Salomon. In the following, the conducted steps to set up the MC run are presented and evaluated on the basis of the chosen example, a high-rise building with a parametrically triangulated facade, compare Figure 6.

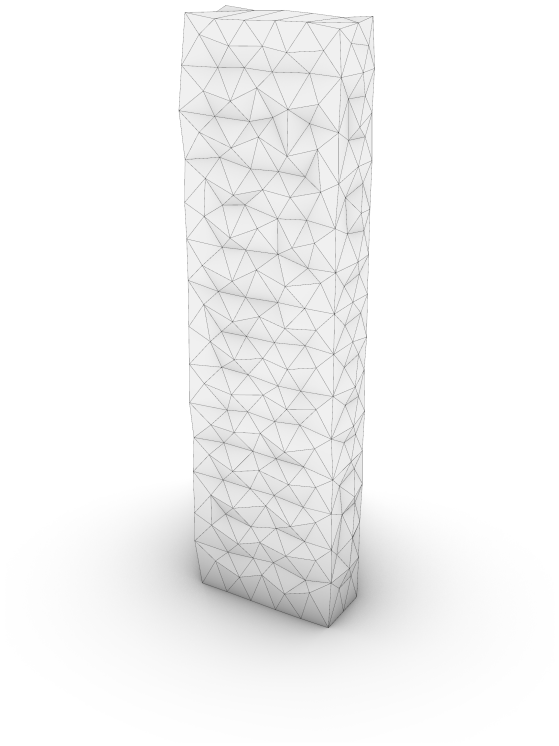

Figure 6: Visualization of the original CAD model of the chosen high-rise building.

Pre-processing: In a first step, the building geometry is imported into the graphical user interface GiD 15.0 in order to be able to define all geometrical and mesh-specific characteristics as well as the initial and boundary values for the CFD computations in the pre-processing step of the Kratos environment. The CAD import to GiD, in this case directly from the existing Rhinoceros 3D file, provided a seamless transition into the Kratos environment. The chosen definition of mesh sizes and mesh refinement zones in the regions of computational interest are visualised in Figure 7. Initial conditions for the wind inlet velocity u_x are set according to the Eurocode 1 (National Annex Germany) for an inner city region as $u_x(z) = 0.56\, u_b\, (z/10)^{0.3}$ with a basic velocity of $u_b = 25.0\, m/s$ (wind zone 2). The turbulence intensity of the wind profile is defined as the uncertain parameter for the MC computations. The remaining boundary conditions chosen in accordance to the CAARC problem formulation as given in 3. Furthermore, the computational domain is adjusted to fit the problem set-up. Before moving forward towards UQ computations, the model formulation and integrity are successfully tested using a deterministic CFD

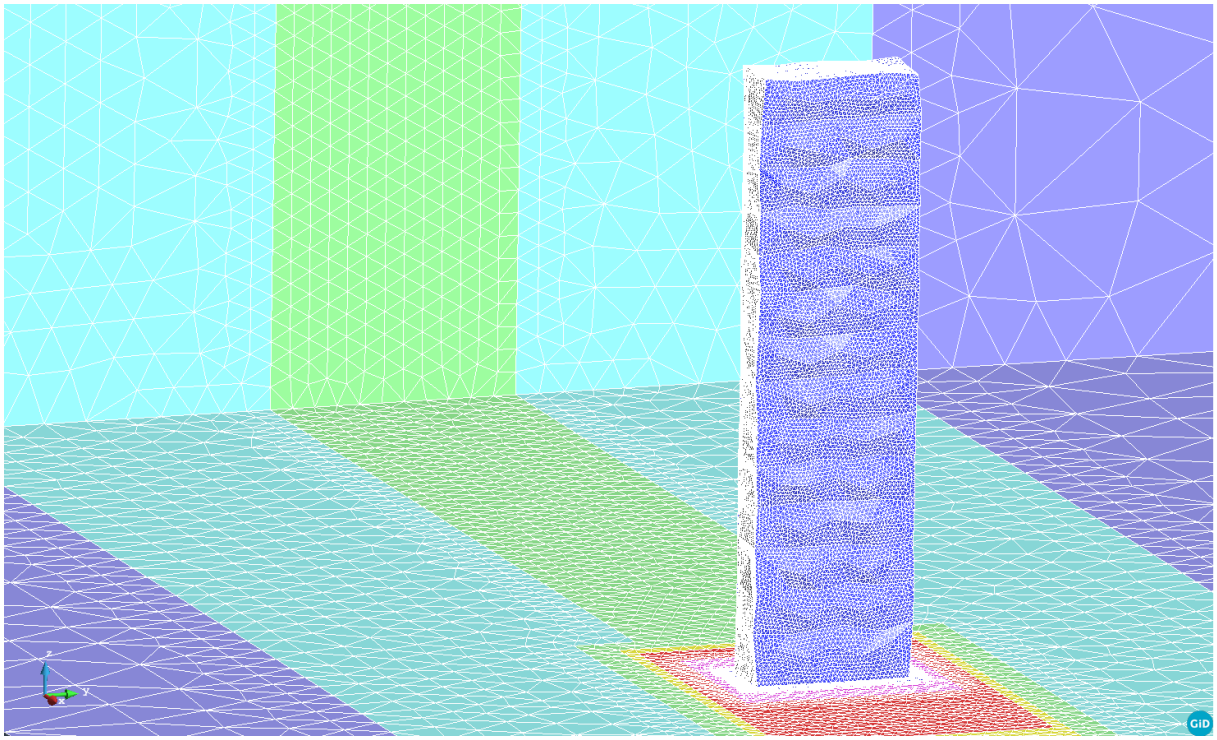

Figure 7: Model import to GiD and meshing within the Kratos pre-processing.

simulation on the in-house cluster.

UQ set-up and test run: In the second step, the UQ formulation is set-up based on the openly accessible example repository of the MC/MLMC application of Kratos[1] [25]. Specifically, the example of the turbulent inlet wind engineering CAARC problem, which is generally comparable to the UQ demonstrator given in Section 3, is used with XMC and PyCOMPSs as the Monte Carlo library and the tool for distributed computing, respectively. The necessary input definitions need to be supplied via predefined function calls to Kratos and XMC, respectively. For the present case these implied, for example, the choice of the QoI (adopted from the CAARC problem as the time averaged drag force), further observable quantities (here the averaged pressure field on the building surfaces) and the number N of MC realisations. The problem definitions require basic understanding of the Python programming language and a short introduction to parallel computing architectures. However, especially the usage of PyCOMPSs as a job distribution tool for the stochastic runs enabled a quick and out-of-the-box usage of the developed UQ methodology for the chosen high-rise example. Due to lack of remaining HPC time, only a small number of $N = 5$ realisations with an academical time interval of $10\,s$ were realised on the IT4I Salomon cluster. Nevertheless, it can safely be stated that the developed methodology and workflow is applicable to common wind-engineering problems.

Post-processing: The statistical results are readily supplied as, e. g., GiD post-processing files after the completed workflow. As an example, Figure 8 shows the averaged pressure field on the high-rise building for the previously described test run. Various post-processing formats are available from single runs, that also allow the user to export

[1] https://github.com/KratosMultiphysics/Examples/tree/master/multilevel_monte_carlo/use_cases

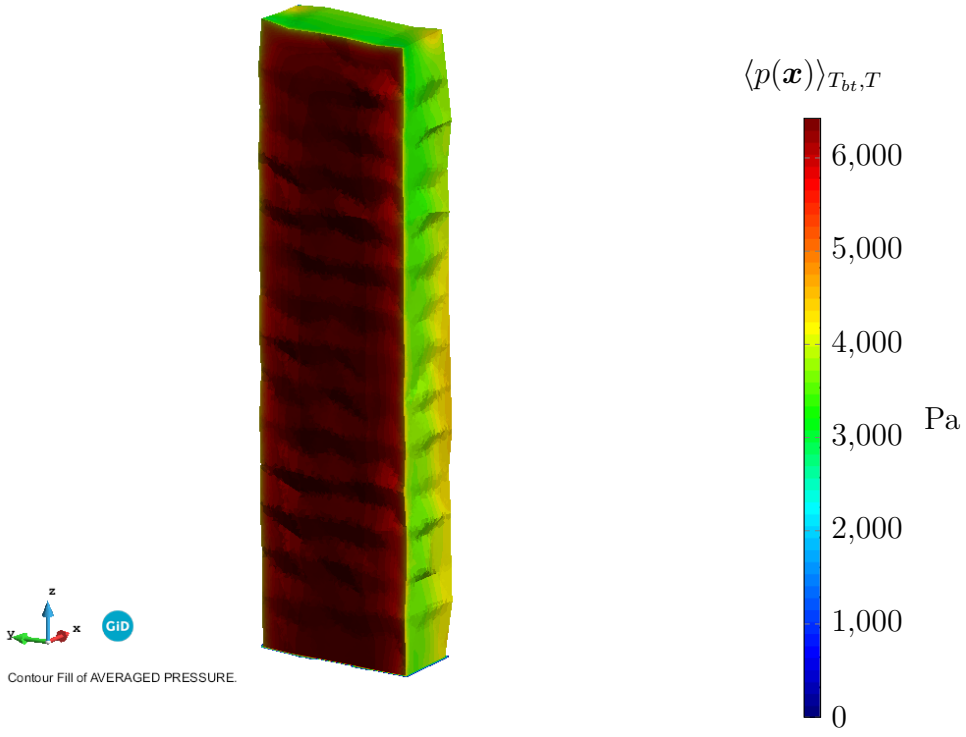

Figure 8: Averaged pressure field on the building surface for the MC test run.

results towards existing post-processing workflow. In this regard, Figure 9 shows the streamline evolution of a single run around the high-rise building.

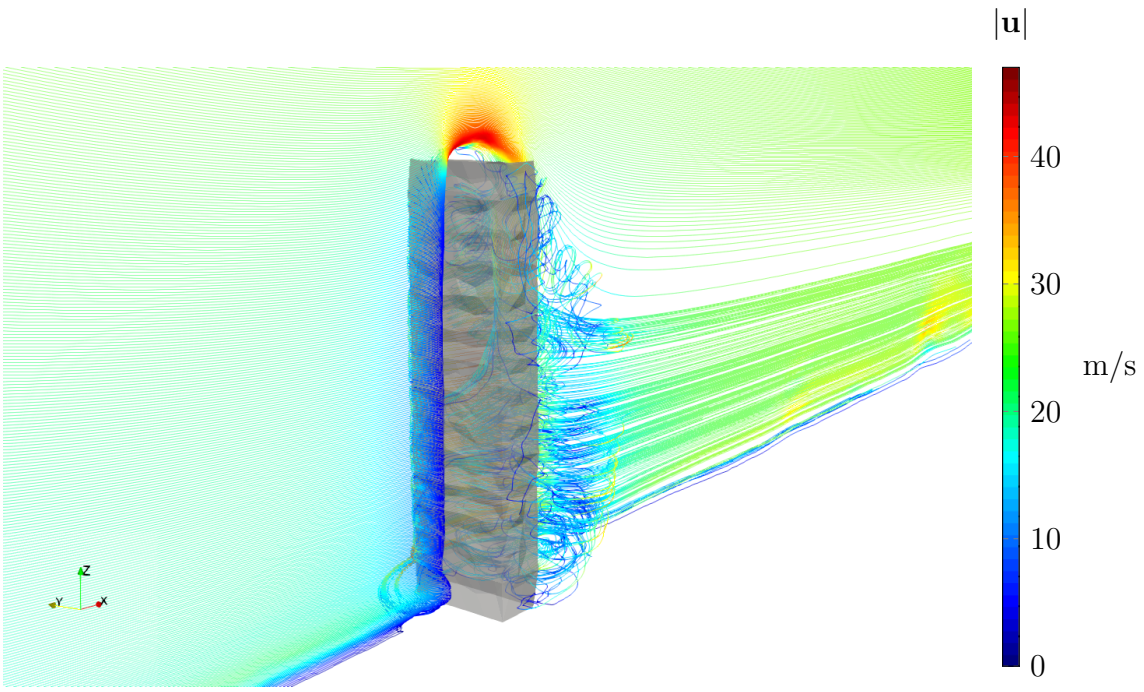

Figure 9: Exported single run results towards the currently employed CFD workflow at str.ucture GmbH.

Overall, the completed test-run with the developed UQ workflow proves the applicability of the presented methods to be used in wind-engineering applications, where the propagation of uncertain parameters is unknown to the user.

References

[1] R. Amela, Q. Ayoul-Guilmard, S. Ganesh, R. Tosi, R. M. Badia, F. Nobile, R. Rossi, and C. Soriano. D5.2 Release of ExaQUte MLMC Python engine. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2019. URL `https://doi.org/10.23967/exaqute.2021.2.024`.

[2] R. Amela, R. M. Badia, S. Böhm, R. Tosi, C. Soriano, and R. Rossi. D4.2 Profiling report of the partner ' s tools , complete with performance suggestions. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2019. URL `https://doi.org/10.23967/exaqute.2021.2.023`.

[3] A. Apostolatos, R. Rossi, and C. Soriano. D7.2 Finalization of "deterministic" verifcation and validation tests. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2020. URL `https://doi.org/10.23967/exaqute.2021.2.006`.

[4] Q. Ayoul-Guilmard, R. M. Badia, J. Ejarque, M. Núñez, S. Ganesh, F. Nobile, C. Soriano, C. Roig, R. Rossi, and R. Tosi. D1.3 First public Release of the solver. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2020. URL `https://doi.org/10.23967/exaqute.2021.2.007`.

[5] Q. Ayoul-Guilmard, S. Ganesh, F. Nobile, R. M. Badia, J. Ejarque, B. Keith, A. Kodakkal, M. Núñez, C. Roig, R. Rossi, R. Tosi, and C. Soriano. D1.4 Final public Release of the solver. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2020. URL `https://doi.org/10.23967/exaqute.2021.2.009`.

[6] Q. Ayoul-Guilmard, S. Ganesh, F. Nobile, R. Rossi, R. Tosi, R. M. Badia, and R. Amela. XMC, 2020. URL `https://doi.org/10.5281/zenodo.3235832`.

[7] Q. Ayoul-Guilmard, S. Ganesh, M. Núñez, R. Tosi, F. Nobile, R. Rossi, and C. Soriano. D5.3 Report on theoretical work to allow the use of MLMC with adaptive mesh refinement. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2020. URL `https://doi.org/10.23967/exaqute.2021.2.002`.

[8] Q. Ayoul-Guilmard, S. Ganesh, M. Núñez, R. Tosi, F. Nobile, R. Rossi, and C. Soriano. D5.4 Report on MLMC for time dependent problems. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2020. URL `https://doi.org/10.23967/exaqute.2021.2.005`.

[9] R. M. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent. {COMP} Superscalar, an interoperable programming framework. *SoftwareX*, 3–4, 2015. doi:10.1016/j.softx.2015.10.004.

[10] A. L. Braun and A. M. Awruch. Aerodynamic and aeroelastic analyses on the CAARC standard tall building model using numerical simulation. *Computers and Structures*, 87(9-10):564–581, may 2009. ISSN 00457949. doi:10.1016/j.compstruc.2009.02.002. URL `http://dx.doi.org/10.1016/j.compstruc.2009.02.002`.

[11] V. Cima, S. Böhm, J. Martinovič, J. Dvorsk\'y, K. Janurová, T. V. Aa, T. J. Ashby, and V. Chupakhin. HyperLoom: A Platform for Defining and Executing Scientific Pipelines in Distributed Environments. In *Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms*, pages 1–6. ACM, 2018.

[12] P. Dadvand, R. Rossi, and E. Oñate. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of Computational Methods in Engineering*, 17(3):253–297, 2010. ISSN 11343060. doi:10.1007/s11831-010-9045-2.

[13] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. R. Idelsohn, and E. Oñate. Migration of a generic multi-physics framework to HPC environments. *Computers and Fluids*, 80(1):301–309, 2013. ISSN 00457930. doi:10.1016/j.compfluid.2012.02.004.

[14] T. Hida, H.-H. Kuo, J. Potthoff, and L. Streit. *White noise: an infinite dimensional calculus*, volume 253. Springer Science & Business Media, 2013.

[15] J. D. Holmes and T. K. Tse. International high-frequency base balance benchmark study. *Wind and Structures, An International Journal*, 18(4):457–471, 2014. ISSN 12266116. doi:10.12989/was.2014.18.4.457.

[16] S. Huang, Q. S. Li, and S. Xu. Numerical evaluation of wind effects on a tall steel building by CFD. *Journal of Constructional Steel Research*, 2007. ISSN 0143974X. doi:10.1016/j.jcsr.2006.06.033.

[17] J. C. Hunt and D. J. Carruthers. Rapid distortion theory and the 'problems' of turbulence. *Journal of Fluid Mechanics*, 212(2):497–532, 1990. ISSN 14697645. doi:10.1017/S0022112090002075.

[18] J. JCSS. Probabilistic model code. *Joint Committee on Structural Safety*, 2001.

[19] H.-H. Kuo. *White noise distribution theory*. CRC press, 2018.

[20] G. J. Lord, C. E. Powell, and T. Shardlow. *An introduction to computational stochastic PDEs*, volume 50. Cambridge University Press, 2014.

[21] F. Lordan, E. Tejedor, J. Ejarque, R. Rafanell, J. Álvarez, F. Marozzo, D. Lezzi, R. Sirvent, D. Talia, and R. M. Badia. ServiceSs: An Interoperable Programming Framework for the Cloud. *Journal of Grid Computing*, 12(1):67–91, 2014. ISSN 15707873. doi:10.1007/s10723-013-9272-5.

[22] J. Mann. The spatial structure of neutral atmospheric surface-layer turbulence. *Journal of fluid mechanics*, 273:141–168, 1994.

[23] J. Mann. Wind field simulation. *Probabilistic Engineering Mechanics*, 13(4):269–282, 1998. ISSN 02668920. doi:10.1016/s0266-8920(97)00036-2. URL `https://ac-els-cdn-com.recursos.biblioteca.upc.edu/S0266892097000362/1-s2.0-S0266892097000362-main.pdf?`

_tid=35c100fe-45d6-41b7-8534-b299ed7855a3{&}acdnat=1531413687_51eb28e70aa14ade51f9d2c0ef662d38.

[24] V. Mataix Ferrándiz, P. Bucher, R. Rossi, R. Zorrilla, J. Cotela Dalmau, J. Maria, M. A. Celigueta, and G. Casas. KratosMultiphysics/Kratos: KratosMultiphysics 8.1, 2020. URL https://doi.org/10.5281/zenodo.3234644.

[25] V. Mataix Ferrándiz, R. Tosi, I. de Pouplana, R. Zorrilla, L. Gracia, S. Warnakulasuriya, M. Núñez, B. Chandra, A. Ghantasala, P. Bucher, J. Cotela, A. Franci, F. Arrufat, K. B. Sautter, S. Latorre, J. González-Usúa, A. Geiser, P. Dadvand, M. Maso, D. Baumgaertner, M. A. Celigueta, R. Kikkeri Nagaraja, S. Wenczowski, B. Saridar, C. Roig, M. Zidan, and G. Casas. KratosMultiphysics/Examples: Kratos Examples 8.1, 2020. URL https://doi.org/10.5281/zenodo.4292799.

[26] M. R. Maxey. Distortion of turbulence in flows with parallel streamlines. *Journal of Fluid Mechanics*, 124:261–282, 1982. ISSN 14697645. doi:10.1017/S0022112082002493.

[27] A. Michalski, D. Andersson, R. Rossi, and C. Soriano. D7.1, Delivery of geometry and computational model. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2021. URL https://doi.org/10.23967/exaqute.2021.2.020.

[28] E. D. Obasaju. Measurement of forces and base overturning moments on the CAARC tall building model in a simulated atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 1992. ISSN 01676105. doi:10.1016/0167-6105(92)90361-D.

[29] M. Pisaroni, F. Nobile, and P. Leyland. A Continuation Multi Level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible inviscid aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 326:20–50, 2017. ISSN 00457825. doi:10.1016/j.cma.2017.07.030.

[30] E. Tejedor, Y. Becerra, G. Alomar, A. Queralt, R. M. Badia, J. Torres, T. Cortes, and J. Labarta. PyCOMPSs: Parallel computational workflows in Python. *International Journal of High Performance Computing Applications*, 31(1):66–82, 2017. ISSN 17412846. doi:10.1177/1094342015594678.

[31] R. Tosi, R. Amela, R. Badia, M. Núñez, R. Rossi, and C. Soriano. D1.2 First release of the softwares. Technical report, Open Access Repository of the ExaQUte project: Deliverables, 2019. URL https://doi.org/10.23967/exaqute.2021.2.011.

[32] R. Tosi, M. Núñez, B. Keith, J. Pons-Prats, B. Wohlmuth, and R. Rossi. Scalable dynamic asynchronous monte carlo framework applied to wind engineering problems. International Conference on Uncertainty Quantification & Optimisation, 2020.

[33] A. Townsend. *The structure of turbulent shear flow*. Cambridge university press, 1980.

[34] R. Tyrrell Rockafellar and J. O. Royset. Engineering Decisions under Risk Averseness. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 1(2):04015003, mar 2015. ISSN 23767642. doi:10.1061/AJRUA6.0000816.

[35] T. Von Kármán. Progress in the statistical theory of turbulence. *Proceedings of the National Academy of Sciences of the United States of America*, 34(11):530, 1948.