



Exascale Quantification of Uncertainties for  
Technology and Science Simulation

## D6.4 Report on stochastic optimisation for unsteady problems

### 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:	WP6 task 6.4
Status:	Final version



## Authoring

Prepared by EPFL, CIMNE, TUM				
Authors	Partner	Modified	Version	Comments
Quentin Ayoul-Guilnard	EPFL	All	1.0	Coordinator
Fabio Nobile		§ 3, A		Expert and reviewer
Sundar Ganesh		§ 4		
Marc Nuñez	CIMNE	§ 4		
Anoop Kodakkal	TUM	§ 5		

## 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	2020-12-11	✓
Coordinator	Cecilia Soriano	CIMNE	2020-12-14	✓
Coordinator	Riccardo Rossi	CIMNE	2020-12-14	✓

## Executive summary

This report brings together methodological research on stochastic optimisation and work on benchmark and target applications of the ExaQute project, with a focus on unsteady problems. A practical, general method for the optimisation of the conditional value at risk is proposed. Three different optimisation problems are described: an oscillator problem selected as a suitable trial and illustration case; the shape optimisation of an airfoil, chosen as a benchmark application in the project; the shape optimisation of a tall building, which is the challenging target application set for ExaQute. For each problem, the current developments and results are presented, the application of the proposed method is discussed, and the work to be done until the end of the project is laid out.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>General optimisation of a CVaR</b>	<b>8</b>
2.1	Optimisation problem and method . . . . .	8
2.2	Practical optimisation . . . . .	11
<b>3</b>	<b>Van der Pol oscillator with random forcing</b>	<b>15</b>
3.1	Optimisation problem . . . . .	15
3.2	Practical optimisation . . . . .	16
3.3	Future work . . . . .	18
<b>4</b>	<b>Potential flow around an airfoil</b>	<b>19</b>
4.1	Potential-flow problem . . . . .	19
4.2	Deterministic adjoint equations derivation . . . . .	25
4.2.1	Partial derivatives of the physical problem . . . . .	27
4.2.2	Partial derivatives of the lift coefficient . . . . .	28
4.2.3	Partial derivatives of volume and perimeter functionals . . . . .	29
4.2.4	Partial derivatives on airfoil geometry definitions . . . . .	30
4.3	Results on deterministic optimisation . . . . .	33
4.4	Prospects on optimisation under uncertainty . . . . .	33
<b>5</b>	<b>Unsteady wind around a building</b>	<b>37</b>
5.1	Drag of a twisted, tapered building . . . . .	37
5.1.1	Uncertainties in the incoming wind flow . . . . .	37
5.1.2	Details of numerical simulation . . . . .	39
5.2	Minimisation of the expected base moment . . . . .	41
5.2.1	Description of the optimisation problem . . . . .	41
5.2.2	Preliminary results on optimisation under uncertainties . . . . .	44
5.3	Prospects on CVaR optimisation . . . . .	46
5.4	Variation: parameterised facade . . . . .	47
<b>6</b>	<b>Conclusion</b>	<b>51</b>
	<b>References</b>	<b>53</b>
<b>A</b>	<b>Adjoint of first-order ODE with additive noise</b>	<b>55</b>
	<b>List of algorithms</b>	<b>57</b>
	<b>List of figures</b>	<b>57</b>

Acronyms	58
Abbreviations	58

# 1 Introduction

The ExaQUte project aims at developing efficient methods to design structures robust with respect to uncertain loading. These methods are primarily intended for applications in wind and civil engineering, such as the design of tall buildings to withstand uncertain wind conditions. Among the many scientific aspects involved, this document reports on the research that has been done in ExaQUte on optimisation under uncertainties (OUU). For unsteady problems, the major challenges include the turbulence of wind flows, which exhibit chaotic behaviour; the difficulty of estimating accurately the sensitivities of the objective function with respect to the design parameters, in chaotic regimes; and the tremendous computational time generally required for such optimisation constrained by partial differential equations (PDEs).

The project has chosen to investigate gradient-based optimisation methods, and favours adjoint calculus to estimate said gradient whenever possible, due to the generally high number of design parameters. Another important choice is the measure of risk associated to the objective function when robust optimisation under uncertain conditions is sought. The conditional value at risk (CVaR) has been selected as risk measure in this work, for its ability to control expected failure states. Lastly, the combined complexity of computational fluid dynamics (CFD) and stochastic optimisation highlights the need for efficient computational methods designed for high-performance computing (HPC); the ExaQUte project has chosen to focus on multi-level Monte Carlo (MLMC) methods to leverage distributed computing in uncertainty quantification. Although these will not be discussed in this report, parallelisation is also used at several other levels – exempli gratia (e.g.) PDE solver, mesh generation.

This document follows a series of previous research reports on optimisation: tools for deterministic shape optimisation ([deliverable 6.1](#)); the computation of stochastic sensitivities ([deliverable 6.2](#)); and the gradient-based methods for the optimisation of various risk measures, including the CVaR ([deliverable 6.3](#)). This document reports on the application of the methods and tools previously presented to several target applications, with a focus on unsteady (id est (i.e.) time-dependent) problems. Therefore, it features elements of OUU, CFD, and wind engineering; By the end of the project, these will be fully integrated into a global framework for robust design. The final results will be presented in deliverable 6.5.

In this document, we first present in § 2 the specific method that has been identified for the efficient optimisation of the CVaR of a random quantity of interest, including the details of a practical algorithm suitable for implementation. Then we illustrate in § 3 the adaptation of this method to a simple oscillator problem, which has been chosen to be a suitable trial case for the implementation of the optimisation toolkit. Section 4 presents a problem of engineering interest selected as

a benchmark with favourable properties for the ExaQUTE project: the optimisation of the shape of an airfoil. We describe the problem and the derivation of its sensitivities by adjoint calculus, present results of deterministic optimisation, and discuss the future application of the proposed method for OOU. Section 5 presents a challenging target application: the optimisation of the shape of a tall building amidst a turbulent wind flow. After the presentation of the problem, preliminary results of OOU are presented, where the lack of adjoint solutions is addressed with an alternative method for the estimation of sensitivities. We also lay out the next developments to be made on this optimisation problem, in the lights of the optimisation method proposed in this report. Lastly, we describe a variation of this problem proposed by our partner `str.ucture`, to be considered for its significant industrial interest.

## 2 General optimisation of a conditional value at risk

### 2.1 Optimisation problem and method

In this section, we recall from [deliverable 6.3](#) the formulation of a robust optimisation problem involving the CVaR of a given quantity of interest. We detail one particular optimisation algorithm that we propose to use in the coming months to tackle the problems of interest to the project, described in the subsequent sections. First we define the generic optimisation problem; then we describe the idealised, generic algorithm proposed; finally, we discuss a practical, cost-efficient implementation together with the approximations it entails.

**Definition** (Conditional value at risk). Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a probability space;  $\beta \in ]0, 1[$ ; and  $X \in L^1(\Omega, \mathbb{R})$  an integrable, real-valued, random variable. The value at risk (VaR) of significance<sup>1</sup>  $\beta$  of  $X$  is defined as

$$\text{VaR}_\beta(X) := \inf\{t \in \mathbb{R} : \mathbb{P}(X \leq t) \geq \beta\} \quad (2.1)$$

In this report, we assume that  $\mathbb{P}(X = \text{VaR}_\beta(X)) = 0$ ; see [Uryasev et al. 2010](#) for a more general context. We define the conditional value at risk (CVaR) of significance  $\beta$  of  $X$  as the following expectation, conditional on the VaR:

$$\text{CVaR}_\beta(X) := \mathbb{E}(X \mid X \geq \text{VaR}_\beta(X)) = \inf\{\mathbb{E}(\phi(X, s)) : s \in \mathbb{R}\}. \quad (2.2)$$

$$\text{with } \phi(X, s) := s + \frac{(X - s)^+}{1 - \beta}. \quad (2.3)$$

Assuming that the cumulative distribution function (CDF) of  $X$  is continuous, the infima in (2.1)–(2.2) are minima and  $\text{CVaR}_\beta(X) = \mathbb{E}(\phi(X, \text{VaR}_\beta(X)))$  (from [Rockafellar and Uryasev 2000](#), theorem 1).  $\diamond$

For multivariate functions such as  $\phi$ , we will denote respectively by  $\nabla_k$  and  $D_k$  the gradient and differential operators with respect to the  $k$ -th variable. Although the function  $(\cdot)^+$  in the expression of  $\phi$  is not differentiable at 0, it was shown in [deliverable 6.3](#) that  $\mathbb{E} \circ \phi$  is still Gâteaux-differentiable at  $(s, X)$  when  $\mathbb{P}(X = s) = 0$ . Moreover, for any  $\theta \in [0, 1]$ , a subgradient  $\bar{\nabla}^\theta$  of  $X \mapsto \mathbb{E}(\phi(X, s))$  is given by

$$\bar{\nabla}_1^\theta \phi(X, s) := \frac{1}{1 - \beta} (\mathbf{1}(X > s) + \theta \mathbf{1}(X = s)). \quad (2.4)$$

<sup>1</sup>Sometimes called ‘value at risk  $1 - \beta$ ’.



If  $\mathbb{P}(X = s) = 0$ , then

$$\nabla_1 \mathbb{E}(\phi(X, s)) = \bar{\nabla}_1^\theta \phi(X, s) \quad \text{almost surely (a.s.).}$$

Thus we extend the definition of the gradient operator  $\nabla$  into  $\bar{\nabla}^\theta$  by choosing a subgradient when the gradient is not defined; the choice of subgradient amounts to choosing  $\theta \in [0, 1]$ . This extension with subgradients was introduced in greater depth in [deliverable 6.3](#), p. 20.

Let  $Z$  denote the space of design variables, whose dimension is assumed finite. We consider the same context introduced in [ibid.](#): the system to be optimised satisfies a PDE dependent on the design  $z \in Z$  and affected by random effects from the aforementioned probability space. This constraint is written in abstract form as

$$F(u, z, \omega) = 0, \quad \text{for almost every (a.e.) } \omega \in \Omega, \quad (2.5)$$

where  $u$  is the state of the system for the given design  $z$  and random event  $\omega$ . We consider a real-valued quantity of interest  $Q$  which depends on  $u$  and  $z$  and assume it to be differentiable. We also assume that for any parameter value  $z \in Z$  there is a unique random solution  $u := \mathcal{M}(z)$  to the stochastic PDE (2.5), therefore our quantity of interest  $Q(z) := \tilde{Q}(\mathcal{M}(z), z)$  depends only on the design parameter. Additionally,  $Q$  is assumed to be convex.

We wish to find the optimal design  $z_*$  defined as

$$z_* := \operatorname{argmin}\{\operatorname{CVaR}(Q(z)) : z \in Z\}. \quad (2.6)$$

For reasons of practicality, we choose to consider problem 1 instead.

**Problem 1** (General CVaR optimisation with PDE constraint). We look for the optimal solution

$$(z_*, s_*) := \operatorname{argmin}\{J(z, s) := \mathbb{E}(\phi(Q(z), s)) : (z, s) \in Z \times \mathbb{R}\}, \quad (2.7)$$

where the constraint (2.5) is implicit in the definition of  $Q$ . According to (Rockafellar and Uryasev 2000, theorem 2),  $z_*$  satisfies also (2.6).  $\diamond$

We propose to solve this optimisation problem with a gradient-descent method based on algorithm 6 from [deliverable 6.3](#). The main challenges in implementing a gradient-descent method for (2.7) are:

- (I) the fact that  $\phi$  is not differentiable everywhere;
- (II) the computation of the derivative of the solution  $u$  to equation (2.5) with respect to the design parameter  $z$ ;
- (III) the choice of the step size to take in the descent direction.

We propose to address (i) with subgradients, using (2.4) for  $\nabla_1(\mathbb{E} \circ \phi)$ . We do not compute  $\nabla_1(\mathbb{E} \circ \phi)$  since the descent method will only be applied to the first variable of  $J$ . As explained below (see p. 11), we will compute directly an approximation of  $\operatorname{argmin}\{J(z, s) : s \in \mathbb{R}\}$ .

To address (ii) we consider either adjoint calculus or finite differences. Using adjoint calculus, from deliverable 6.3, p. 18,

$$\begin{aligned} \bar{\nabla}_1^\theta J(z, s) &= \mathbb{E}(g(z, s)) \\ \text{with } g(z, s) &:= D_2 F^*(u, z, \cdot)(y) \end{aligned} \quad (2.8)$$

where  $y$  satisfies

$$D_1 F^*(u, z, \cdot)(y) = \frac{D_1 \tilde{Q}(u, z)}{1 - \beta} (\mathbb{1}(Q(z) > s) + \theta \mathbb{1}(Q(z) = s)). \quad (2.9)$$

Nota bene (NB) although (2.9) is often named ‘adjoint’ or ‘dual’ equation to the ‘primal’ equation (2.5), its definition depends not only on (2.5) but also on the optimisation problem (e.g. the chosen risk measure).

Another approach to estimate  $\bar{\nabla}_1^\theta J(z, s)$  is by the method of *finite differences*. Let us illustrate it with a centred scheme (or ‘stencil’). For the sake of legibility, we choose  $s \in \mathbb{R}$  and note the first partial application of  $J$ :  $J_1 := z \mapsto J(z, s)$ . If  $J_1$  is convex, we can estimate a subdifferential of it at  $z \in Z$  in direction  $y \in Z$  as

$$\Delta_{h,y} J_1(z) := \frac{J_1(z + hy) - J_1(z - hy)}{2h}$$

for a given  $h \in ]0, +\infty[$ . If  $J_1$  is differentiable at  $z$ , then

$$\Delta_{h,y} J_1(z) = D_1 J(z, s)(y) + \mathcal{O}(h^2).$$

Denoting  $\mathbf{e} := \{\mathbf{e}_i : i \in \llbracket 1, \dim Z \rrbracket\}$  an orthonormal basis of  $Z$ , we can estimate a subgradient  $\Delta_h J_1(z)$  of  $J_1$  by its coordinates in  $\mathbf{e}$ :

$$\forall i \in \llbracket 1, \dim Z \rrbracket, \quad \mathbf{e}_i \cdot \Delta_h J_1(z) = \Delta_{h,\mathbf{e}_i} J_1(z).$$

If  $J_1$  is smooth,  $\Delta_h J_1$  gives a second-order approximation of  $\nabla J_1$ , i.e.

$$\|\nabla J_1(z) - \Delta_h J_1(z)\| \in \mathcal{O}(h^2);$$

higher orders of accuracy can be achieved by a stencil with more points. However, the number of evaluations of  $J_1$  required to build the approximation  $\Delta_h J_1$  is linearly proportional to  $\dim Z$ . This is a strong argument in favour of adjoint-based derivation, since  $\dim Z$  is often large in shape optimisation. We focus on the adjoint approach henceforth; finite differences are discussed in more details in § 5.3, p. 46.

With respect to (III), we seek to choose the step size  $\gamma$  adaptively so as to provide the maximal reduction of the value of the objective function. Given a design parameter  $z \in Z$  and a descent direction  $p \in Z$ , we define the optimal step size as

$$\gamma^*(z, p) := \operatorname{argmin}\{J_1(z + cp) : c \in ]0, +\infty[ \}. \quad (2.10)$$

This minimisation problem will be solved approximately by building an interpolation of the function  $c \mapsto J_1(z + cp)$ . This will be detailed in § 2.2.

Finally, we choose to estimate convergence by evaluating the relative size of the increments in  $(z, s)$ , and comparing it to a tolerance  $\eta \in ]0, +\infty[$  chosen a priori. I.e. we stop at iteration  $k \in \mathbb{N}$  if

$$S(z_k, z_{k-1}, s_k, s_{k-1}) := \frac{\|z_k - z_{k-1}\|_Z}{\|z_k\|_Z} + \frac{|s_k - s_{k-1}|}{|s_k|} \leq \eta,$$

assuming  $\|z_k\|_Z \neq 0 \neq |s_k|$ .

The steps described above are summarised as algorithm 1. This algorithm is called ‘ideal’ in the sense that it assumes that we can solve exactly (2.5), (2.9) and (2.10). In practice, approximations are necessary.

---

ALGORITHM 1: Ideal gradient-descent with exact var

---

- 1 INPUT:  $z_0, \eta, \theta$
  - 2 INITIALISE:  $k := 1, s_0 := 1$
  - 3 WHILE  $S(z_k, z_{k-1}, s_k, s_{k-1}) > \eta$  DO
  - 4     Compute  $Q(z_k)$  by solving primal problem (2.5) for  $z_k$
  - 5     Find the var:  $s_{k+1} := \operatorname{argmin}\{\mathbb{E}(\phi(Q(z_k), s)) : s \in \mathbb{R}\}$
  - 6     Compute the subgradient  $\bar{\nabla}_1^\theta J(z_k, s_{k+1})$
  - 7     Set the descent direction  $p := -\bar{\nabla}_1^\theta J(z_k, s_{k+1}) / \|\bar{\nabla}_1^\theta J(z_k, s_{k+1})\|$
  - 8     Find the optimal step size  $\gamma_k := \gamma^*(z_k, p)$  according to (2.10)
  - 9     Update design  $z_{k+1} := z_k + \gamma_k p$
  - 10     $k := k + 1$
  - 11 RESULT:  $z_k, s_k$
- 

## 2.2 Practical optimisation

To estimate the var in algorithm 1, line 5, we use the MLMC method described previously in deliverable 6.3, pp. 22–23 and proposed by Krumscheid and Nobile 2018. We recall only the gist of it below, and direct the reader to the previous

references for more details. Let  $L \in \mathbb{N}$  and  $\mathbf{N} \in \mathbb{N}^{L+1}$ : for any level  $l \in \llbracket 0, L \rrbracket$  we consider an approximate quantity of interest  $Q_l \approx Q$ . This approximation typically comes from a discretisation of the underlying PDE, using  $M_l$  degrees of freedom. The Monte Carlo (MC) estimation of  $\mathbb{E}(Q_l)$  with  $m \in \mathbb{N}$  independent samples is denoted

$$\mu_m(Q_l) = \frac{1}{m} \sum_{i=1}^m Q_l^{(i)}$$

Considering now a sequence of discretisations  $(Q_l)_{l=0}^L$  with increasing numbers of degrees of freedom  $(M_l)_{l=0}^L \in \mathbb{N}^{L+1}$ , we can then define a MLMC estimator of  $\mathbb{E}$ :

$$\mathbb{E}(Q) \approx \mu_{\mathbf{m}}(Q) := \mu_{m_0}(Q_0) + \sum_{l=1}^L \mu_{m_l}(Q_l - Q_{l-1}),$$

where  $\mathbf{m} := (m_l)_{l=0}^L \in \mathbb{N}^{L+1}$  is a decreasing sequence of sample sizes. To approximate the parametric expectation  $\psi = s \mapsto \mathbb{E}(\phi(Q, s))$ :

1. we consider  $n$  points  $\mathbf{r} := (r_i)_{i=1}^n \in \mathbb{R}^n$ ;
2. we compute the MLMC estimations  $\psi(r_i) \approx \mu_{\mathbf{m}}(\phi(Q, r_i))$  for all  $i \in \llbracket 1, n \rrbracket$ ;
3. we interpolate the values  $\{(r_i, \mu_{\mathbf{m}}(\phi(Q, r_i))) : i \in \llbracket 1, n \rrbracket\}$  with a suitable interpolator (e.g. splines).

We denote  $\Phi_{\mathbf{m}, \mathbf{r}}(Q)$  such an interpolant approximating  $\psi$ . We can then estimate

$$\begin{aligned} \text{VaR}_\beta(Q) &= \text{argmin } \psi \approx \text{argmin } \Phi_{\mathbf{m}, \mathbf{r}}(Q) \\ \text{and } \text{cVaR}_\beta(Q) &= \min \psi \approx \min \Phi_{\mathbf{m}, \mathbf{r}}(Q). \end{aligned}$$

For any  $\epsilon \in ]0, +\infty[$ , a posteriori error estimators enable us to choose  $L$ ,  $\mathbf{m}$  and  $\mathbf{r}$  such that (s.t.)

$$\text{MSE}(\text{argmin } \Phi_{\mathbf{m}, \mathbf{r}}(Q)) := \mathbb{E}\left(\left|\text{VaR}_\beta(Q) - \text{argmin } \Phi_{\mathbf{m}, \mathbf{r}}(Q)\right|^2\right) \leq \epsilon.$$

NB MC estimation is merely a particular case of this method, for  $L := 0$ . Therefore, this and what follows apply as well to single-level estimation.

We use a similar MLMC estimator  $\mu_{\mathbf{m}'}$  on line 7 to estimate  $\overline{\nabla}_1^\theta J(z_k, s_{k+1})$ , with  $\mathbf{m}' \in \mathbb{N}^{L'+1}$  and  $L' \in \mathbb{N}$ . The computational cost of this algorithm is dominated by the time necessary for resolutions of the primal equation (2.5) and its dual (2.9). In order to curb this cost, one can choose to pick at line 6 random events that had been drawn independently at line 4–5, for which the solutions to the primal problems are already known. If more samples are required, new events are drawn independently.

Regarding the problem (2.10) for the optimal step size: an accurate resolution may require numerous evaluations of  $J$ , and the cost thereof may outweigh the

benefit of the optimal step size  $\gamma^*$ . However, even a rough approximation of  $\gamma^*$  is useful. Therefore, we propose to approximate  $c \mapsto J(z_k + cp)$  with a quadratic interpolation  $\mathcal{J}$  and minimise the interpolant instead:

$$\check{J} := c \mapsto J(z_k + cp) \approx c \mapsto a_2 c^2 + a_1 c + a_0 =: \mathcal{J}(\check{J})$$

hence we choose

$$\gamma(z, p) = -\frac{a_1}{2a_2} = \operatorname{argmin} \mathcal{J}(\check{J}) \approx \gamma^*(z, p)$$

From lines 5 and 6, we already know respectively  $\check{J}(0) = J(z_k, s_{k+1})$  and  $\bar{\nabla}^\theta \check{J}(0) = \bar{\nabla}_1^\theta J(z_k, s_{k+1})$ . We choose an additional point  $\tilde{c} \in \mathbb{R} \setminus \{0\}$  at which we compute  $\check{J}(\tilde{c})$ , then we solve

$$\begin{aligned} & \begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ \tilde{c}^2 & \tilde{c} & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \check{J}(0) \\ \bar{\nabla}^\theta \check{J}(0) \\ \check{J}(\tilde{c}) \end{pmatrix}, \\ \text{which yields } & \begin{cases} a_0 = \check{J}(0) \\ a_1 = \bar{\nabla}^\theta \check{J}(0) - \frac{2}{\tilde{c}-2} \left( \frac{\check{J}(\tilde{c}) - \check{J}(0)}{\tilde{c}} - \bar{\nabla}^\theta \check{J}(0) \right) \\ a_2 = \frac{1}{\tilde{c}-2} \left( \frac{\check{J}(\tilde{c}) - \check{J}(0)}{\tilde{c}} - \bar{\nabla}^\theta \check{J}(0) \right). \end{cases} \end{aligned}$$

We propose heuristically  $\tilde{c} := \gamma_{k-1}$ .

In practice, we approximate  $\check{J}(\tilde{c}) \approx \mu_{\mathbf{m}''}(\phi(Q(z_k + \tilde{c}p), s_{k+1}))$  with  $\mathbf{m}'' \in \mathbb{N}^{L''+1}$  and  $L'' \in \mathbb{N}$ . Since this happens at line 8 of iteration  $k$  of algorithm 1, we already have  $\mathbf{m}'$  samples<sup>2</sup> of  $Q(z_k)$  from which we can build a control variate:

$$\check{J}(\tilde{c}) \approx \mu_{\mathbf{m}''}(\phi(Q(z_k + \tilde{c}p), s_{k+1}) - \alpha \phi(Q(z_k), s_{k+1})) + \alpha \mu_{\mathbf{m}'}(\phi(Q(z_k), s_{k+1})).$$

Thus our estimation of  $\check{J}$  can achieve the same accuracy with fewer samples – i.e. faster – than a single MLMC estimator. The greater the correlation between  $\phi(Q(z_k + \tilde{c}p), s_{k+1})$  and  $\phi(Q(z_k), s_{k+1})$ , the greater the gain. This method of variance reduction has been presented in more details – including the choice of the value of the control coefficient  $\alpha$  – in deliverable 5.4, § 5.1. In the interest of cost efficiency, one may build  $\mu_{\mathbf{m}''}$  by using a subset<sup>3</sup> of the random events used

<sup>2</sup>I.e.  $\forall l \in \llbracket 0, L' \rrbracket$ , we have  $m'_l$  samples of  $Q_l(z_k) - Q_{l-1}(z_k)$  or, more accurately,  $\max\{m'_l, m_l\}$ .

<sup>3</sup>This practice is often called ‘sub-sampling’.

in  $\mu_{\mathbf{m}'}$ , as proposed in [deliverable 5.4](#), p. 23. By doing so, only evaluations of  $\phi(Q(z_k + \tilde{c}p), s_{k+1})$  need to be computed<sup>4</sup>. This requires to store all samples of the random events, as well as the corresponding values of the quantity of interest. This is reasonable if each of these is a random vector of small size. However, it may require extensive storage if some of these quantities are spatial fields or time processes.

The practical considerations above are summarised into [algorithm 2](#).

---

**ALGORITHM 2:** Practical gradient-descent for CVaR optimisation with MLMC

---

- 1 INPUT:  $z_0, \gamma_0, \eta, \epsilon, \epsilon', \theta$
  - 2 INITIALISE:  $k := 1, s_0 := 0$
  - 3 WHILE  $S(z_k, z_{k-1}, s_k, s_{k-1}) > \eta$  DO
  - 4     Compute  $\Phi_{\mathbf{m},r}(Q(z_k))$  s.t.  $\text{MSE}(\text{argmin } \Phi_{\mathbf{m},r}(Q(z_k))) \leq \epsilon_k$
  - 5     Estimate the VaR as  $s_{k+1} := \text{argmin } \Phi_{\mathbf{m},r}(Q(z_k))$
  - 6     Compute the subgradient  $\mu_{\mathbf{m}'}(g(z_k, s_{k+1}))$  s.t.  
        $\text{MSE}(\mu_{\mathbf{m}'}(g(z_k, s_{k+1}))) \leq \epsilon'_k$
  - 7     Set the descent direction  $p := -\mu_{\mathbf{m}'}(g(z_k, s_{k+1})) / \|\mu_{\mathbf{m}'}(g(z_k, s_{k+1}))\|$
  - 8     Evaluate  $\check{J}(\gamma_{k-1}) = J(z_k + \gamma_{k-1}p, s_k)$
  - 9     Build interpolation  $\mathcal{J}(\check{J})$
  - 10    Find the optimal step size  $\gamma_k := \text{argmin } \mathcal{J}(\check{J})$
  - 11    Update design  $z_{k+1} := z_k + \gamma_k p$
  - 12     $k := k + 1$
  - 13 RESULT:  $z_k, s_k$
- 

<sup>4</sup>This assumes  $L'' \leq L'$  and  $\forall l \in \llbracket 0, L'' \rrbracket, m_l'' < m_l'$ .

### 3 Van der Pol oscillator with random forcing

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). 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. Furthermore, the sensitivity of its solution with respect to some design parameters is amenable to computation by adjoints, as proposed above. Therefore, it is suitable for presenting an adaptation of the method described in § 2.

#### 3.1 Optimisation problem

Let us first introduce some notations. Let  $f : \mathbb{R} \times \Omega \rightarrow \mathbb{R}$  be a random process. We will denote respectively  $\dot{f}$  and  $\ddot{f}$  its first and second partial derivatives with respect to the first variable. The temporal average – or time average – of  $f$  over the interval  $[0, T]$ , with  $T \in [0, +\infty[$ , is the real-valued random variable defined for a.e.  $\omega \in \Omega$  as

$$\langle f \rangle_T(\omega) := \frac{1}{T} \int_{t=0}^T f(t, \omega) dt.$$

**Problem 2** (Randomly-forced Van der Pol oscillator). Let  $z, T \in ]0, +\infty[$  and  $u_0, \dot{u}_0, \tau \in \mathbb{R}$ . We denote  $W : [0, T] \times \Omega \rightarrow \mathbb{R}$  a standard Wiener process. We define the Van der Pol oscillator with stochastic forcing as the function  $u : [0, T] \times \Omega \rightarrow \mathbb{R}$  which satisfies

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

almost everywhere (a.e.) in  $\Omega$ . Setting  $\tau$  to 0 yields the canonical unforced Van der Pol oscillator, which is asymptotically periodic with a period and amplitude independent of the initial conditions  $(u_0, \dot{u}_0)$ .  $\diamond$

**Problem 3** (Optimisation of the Van der Pol oscillator). For any  $z \in \mathbb{R}$ , there is a unique solution to problem 2. Therefore, denoting  $u$  that solution, we can define the mapping

$$Q(z) := \langle (u - a)^2 \rangle_T$$

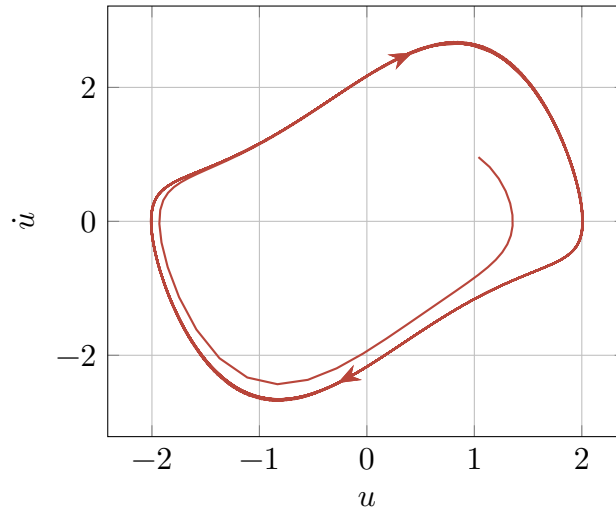


Figure 1: Example of trajectory of an unforced Van der Pol oscillator in the phase space

We wish to solve (2.7) in this particular case, i.e.

$$(z_*, s_*) := \operatorname{argmin} \left\{ J(z, s) := s + \frac{\mathbb{E}((Q(z) - s)^+)}{1 - \beta} : (z, s) \in Z \times \mathbb{R} \right\}.$$

◇

A constraint implicit in the definition of  $Q$  is problem 2, which is a particular case of (2.5). Indeed, problem 3 is exactly of the form of pb 1. Problem 2 was previously presented in deliverable 5.4, pp. 11–16; we refer the reader to it for more details.

### 3.2 Practical optimisation

We discuss here the adaptation of algorithm 2 to the optimisation problem 3. In the spirit of MLMC methods, for any level  $l \in \mathbb{N}$ , we choose a number of time steps  $N_l \in \mathbb{N}$  such that  $N_l < N_{l+1}$  and  $\lim_{l \rightarrow +\infty} N_l = +\infty$ . We introduce a discretisation  $\mathbb{T}_l$  of  $[0, T]$  into  $N_l \in \mathbb{N}$  segments:  $\mathbb{T}_l := \{t_n := nT/N_l : n \in \llbracket 0, N_l \rrbracket\}$ . We denote as follows the corresponding trapezoidal approximation of the temporal average:

$$\langle f \rangle_{\mathbb{T}_l} := \sum_{n=0}^{N_l-1} \frac{f(t_n) + f(t_{n+1})}{2N_l} \approx \langle f \rangle_T.$$



To solve problem 2 over  $\mathbb{T}_l$ , we use the Euler–Maruyama scheme: for any  $n \in \llbracket 1, N_l \rrbracket$ ,

$$\begin{aligned}\dot{u}(t_{n-1}, \cdot) &\approx \frac{N_l}{T}(u(t_n, \cdot) - u(t_{n-1}, \cdot)) \\ \ddot{u}(t_{n-1}, \cdot) &\approx \frac{N_l}{T}(\dot{u}(t_n, \cdot) - \dot{u}(t_{n-1}, \cdot)), \\ \text{and } \dot{W}(t_{n-1}, \cdot) &\approx \frac{N_l}{T}(W(t_n, \cdot) - W(t_{n-1}, \cdot)),\end{aligned}$$

where the Brownian increment is  $W(t_n, \cdot) - W(t_{n-1}, \cdot) = \xi_n \sqrt{T/N_l}$  with  $\xi_n \sim \mathcal{N}(0, 1)$ . The approximations  $u(t_n, \cdot) \approx u_n$  and  $\dot{u}(t_n, \cdot) \approx \dot{u}_n$  satisfy then

$$\begin{pmatrix} u_n \\ \dot{u}_n \end{pmatrix} = \begin{pmatrix} u_{n-1} \\ \dot{u}_{n-1} \end{pmatrix} + \frac{T}{N_l} \begin{pmatrix} z(1 - u_{n-1}^2) \dot{u}_{n-1} - u_{n-1} \\ \dot{u}_{n-1} \end{pmatrix} + \tau \xi_{n-1} \sqrt{\frac{T}{N_l}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (3.1)$$

We define the approximation of  $Q$  on the discretisation level  $l$  as

$$\begin{aligned}Q_l(z) &:= \langle (u - a)^2 \rangle_{\mathbb{T}_l}, \\ \text{with } \langle (u - a)^2 \rangle_{\mathbb{T}_l} &\approx \sum_{n=1}^{N_l} \left( \frac{(u_n - a)^2 + (u_{n-1} - a)^2}{2} \right) \frac{1}{N_l} \\ &\text{and } \begin{pmatrix} u_n \\ \dot{u}_n \end{pmatrix} \text{ satisfying (3.1).}\end{aligned}$$

The performance of MLMC methods depends on the behaviour of the quantity of interest  $Q_l$  with respect to the temporal discretisation  $\mathbb{T}_l$ ; we refer the reader to [deliverable 5.4](#), pp. 15–16 for further details.

Recalling (2.8)–(2.9), we express the sensitivity of the objective function for problem 3:

$$\begin{aligned}\bar{\nabla}_1^\theta J(z, s) &= \mathbb{E}(g(z, s)) \\ \text{with } g(z, s) &:= \langle (1 - u^2) \dot{u} \lambda \rangle_T.\end{aligned}$$

For the approximated quantity of interest  $Q_l$  introduced above, we denote the corresponding approximation

$$g(z, s) \approx g_l(z, s) := \langle (1 - u^2) \dot{u} \lambda \rangle_{\mathbb{T}_l}$$

The discrete random process  $\lambda$  above is the second component of the solution  $\boldsymbol{\lambda}$  to the *discrete adjoint problem*:  $\forall n \in \llbracket 1, N_l \rrbracket$ ,

$$\begin{aligned}\boldsymbol{\lambda}_{n-1} &= \boldsymbol{\lambda}_n + \frac{T}{N_l} \begin{pmatrix} 0 & -(2zu_n \dot{u}_n + 1) \\ 1 & z(1 - u_n^2) \end{pmatrix} \boldsymbol{\lambda}_n + \bar{\nabla}_1^\theta \phi(Q_l(z), s) \frac{(u_n - a)}{N_l} \begin{pmatrix} 2 \\ 0 \end{pmatrix} \\ \text{and } \boldsymbol{\lambda}_{N_l} &= \bar{\nabla}_1^\theta \phi(Q_l(z), s) \frac{(u_n - a)}{N_l} \begin{pmatrix} 1 \\ 0 \end{pmatrix}.\end{aligned}$$

The detailed derivation of the sensitivity of the objective function and the corresponding discrete adjoint problem is available in appendix A. NB the primal problem is solved *forward* in time while its adjoint is solved *backward*, hence their respective names: ‘forward problem’ and ‘backward problem’.

This concludes the adaptation of algorithm 2 to this particular optimisation problem. The practical considerations discussed in § 2 apply here as well: the re-use of samples of  $Q(z)$  from line 4 in line 6, and from this latter one in line 8, together with the use of a control-variate technique with sub-sampling. However, the normalisation of the descent direction at line 7 is pointless here, since the dimension of the parameter space is 1.

### 3.3 Future work

Although the Van der Pol oscillator is not among the target applications of the ExaQUTE project, it shares many of their characteristics while being simpler. It will be a good trial application for the optimisation routines and their interface with the XMC library (Ayoul-Guilmar, Ganesh, Nobile, Rossi et al. 2020). Therefore, the next stage will be to implement algorithm 2. We will discuss in the following sections the modifications to adapt it to applications of wind engineering, which will be the topic of deliverable 6.5.

Additionally, there are two improvements which we may attempt on this simpler optimisation problem. The first one is to make the time horizon a level parameter in the MLMC method, in addition to the time-step size: for any level  $l \in \mathbb{N}$ , we choose  $(N_l, T_l) \in \mathbb{N} \times ]0, T]$  and consider the Van der Pol oscillator over  $]0, T_l]$  with the discretisation in time  $\mathbb{T}_l := \{nT_l/N_l : n \in \llbracket 0, N_l \rrbracket\}$ . This could improve the performance of the MLMC method, provided the parameters  $(N_l, T_l)$  are chosen aptly.

The second point is to investigate an alternative objective function:

$$\tilde{J}(z, s) := \langle \text{cVaR}_\beta((u - a)^2) \rangle_T.$$

A first question is the pertinence of this choice for engineering purposes, compared to (2.6). If it has merits, there is the mathematical question of the reformulation of the optimisation problem, and gradient-descent method thereof.

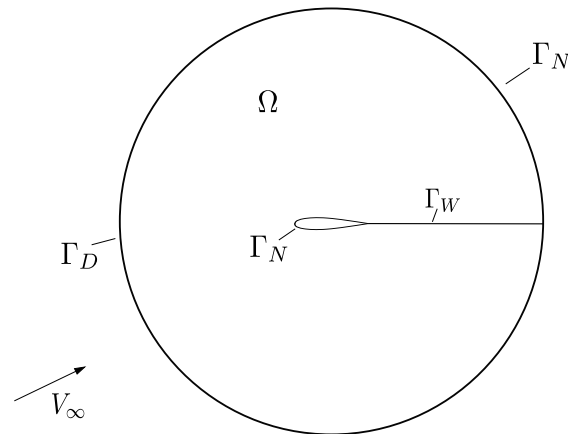


Figure 2: Simplified representation of the domain  $\mathbb{D}$  and its boundary  $\Gamma$

## 4 Potential flow around an airfoil

We consider here a well-known problem in aeronautics: to find the section of an airfoil to maximise its lift. The physics model used is that of the potential flow, described below. This is one of the applications chosen in the ExaQUte project to benchmark methods of OUU, and has been published along with the solver Kratos MultiPhysics (Ferrándiz, Bucher et al. 2020; Ferrándiz, Tosi et al. 2020). In this section we present the deterministic shape optimisation problem as well as the computation of sensitivities by adjoint calculus. Finally, we discuss the applicability to this test case of the OUU problem with CVaR minimisation presented above.

### 4.1 Potential-flow problem

Starting from Navier-Stokes equations, assuming that the flow is irrotational and inviscid, and assuming that the velocity can be written as the gradient of a potential such that  $v = \nabla \Phi$ , the mass conservation equation is reduced to:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla \Phi) = 0 \quad (4.1)$$

where the density,  $\rho$ , can be written in terms of far-field quantities and using the isentropic flow.

$$\frac{\rho}{\rho_\infty} = \left( 1 + \frac{\gamma - 1}{2} \frac{u_\infty^2}{a_\infty^2} \left( 1 - \frac{\nabla \Phi \cdot \nabla \Phi}{u_\infty^2} \right) \right)^{\frac{1}{\gamma - 1}} \quad (4.2)$$

This relation comes from introducing the potential flow assumption in Euler's momentum equation. 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 2, where the far-field boundary is represented by the outer circle. A line denoted by  $\Gamma_W$  represents the wake modelled in the problem. Depending on the flux passing through the domain, the imposed conditions are defined as  $\Gamma_D$  or  $\Gamma_N$ , which will be considered the inlet and the outlet of the domain, respectively. If  $\mathbf{n}$  is defined as the outer normal on the boundary, then  $\Gamma_N$  and  $\Gamma_D$  are defined as follows.

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

At the inlet, a Dirichlet condition is imposed by fixing an initial potential value as specified in (4.3). A Neumann condition is imposed in (4.4) 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: (4.5) imposes mass conservation across the wake; (4.6) imposes pressure equality between the upper and lower parts of the wake.

$$\Phi = \Phi_\infty \quad \text{on } \Gamma_D, \quad (4.3)$$

$$\mathbf{n} \cdot (\rho \nabla \Phi) = g \quad \text{on } \Gamma_N, \quad (4.4)$$

$$\mathbf{n} \cdot (\rho^+ \nabla \Phi^+ - \rho^- \nabla \Phi^-) = 0 \quad \text{on } \Gamma_W, \quad (4.5)$$

$$|\nabla \Phi^+|^2 - |\nabla \Phi^-|^2 = 0 \quad \text{on } \Gamma_W. \quad (4.6)$$

From (4.1), the weak form of the system can be derived by applying the Galerkin method and the divergence theorem:

$$\int_{\mathbb{D}} \rho \nabla \varphi \cdot \nabla \Phi d\mathbb{D} = \int_{\Gamma_N} \varphi \mathbf{n} \cdot (\rho \nabla \Phi) d\Gamma.$$

This weak form is discretized by linear finite elements: it is expressed in terms of the  $M$  nodal shape functions  $N_j$  and the  $M$  corresponding nodal values  $\Phi_j$ :

$$\forall i \in \llbracket 1, M \rrbracket, \quad \sum_{j=1}^M \int_{\mathbb{D}} \rho \nabla N_i \cdot \nabla N_j d\mathbb{D} \Phi_j = \int_{\Gamma_N} N_i g d\Gamma. \quad (4.7)$$

The right-hand side of the equation has been replaced by the Neumann boundary condition. The dependence of  $\rho$  on the potential makes the problem non-linear, so

the system is written in residual form  $R(\Phi) = 0$ . 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  $\Delta\Phi^k$ . For every element  $\mathbb{D}_e$  in the mesh with boundary  $\Gamma_{N_e} = \Gamma_N \cap \partial\mathbb{D}_e$ , its contribution to the residual and the Jacobian is computed, denoted by  $R_i^e$  and  $\mathfrak{J}_{i,j}^e$  respectively.

$$R_i^e(\Phi) = \sum_{j=1}^M \int_{\mathbb{D}_e} \rho \nabla N_i \cdot \nabla N_j d\mathbb{D}\Phi_j - \int_{\Gamma_{N_e}} N_i g d\Gamma,$$

$$\mathfrak{J}_{i,j}^e(\Phi) = \frac{\partial R_i^e(\Phi)}{\partial \Phi_j} = \int_{\mathbb{D}_e} \rho \nabla N_i \cdot \nabla N_j + 2 \frac{\partial \rho}{\partial |v|^2} (\nabla N_j \cdot \nabla \Phi) (\nabla N_i \cdot \nabla \Phi) d\mathbb{D}\Phi_j \quad (4.8)$$

Elemental contributions are assembled to the total system and the potential values  $\Phi^{k+1}$  are updated.

$$\mathfrak{J}(\Phi^k) \Delta\Phi^k = -R(\Phi^k),$$

$$\Phi^{k+1} = \Phi^k + \Delta\Phi^k.$$

In (4.8), the derivative of the density with respect to the local velocity is obtained from (4.2):

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

As it is well-known, the definition of a wake is required in potential flow methods in order to model lift. The wake forms a discontinuity in the domain that allows a discontinuity in the potential field. The intensity of this jump in the potential is related to the lift created by the geometry by means of the Kutta–Joukowski condition. A first approach to model this wake was presented by Eller 2012, where the wake is explicitly modelled in the mesh. This makes the mesh generation complex, as the wake is body-fitted in the domain.

In this deliverable, the approach from Davari et al. 2019 is used, where the wake is implicitly defined with a level-set function that cuts the elements that form the wake, easing mesh generation. The elements that are cut by the wake have duplicated degrees of freedom and use modified shape functions to deal with the split elements. The domain is therefore split into two by the wake. 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  $\mathbb{D}_{W,e}^+$ ,  $\mathbb{D}_{W,e}^-$ ,  $\Gamma_{W,e}^+ = \Gamma_N \cap \partial\mathbb{D}_{W,e}^+$ ,  $\Gamma_{W,e}^- = \Gamma_N \cap \partial\mathbb{D}_{W,e}^-$ , 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  $\Theta$ . The residuals and Jacobians derived from the full potential equation are rewritten for the elements that are intersected by the wake, denoted with  $R_{i,W}^{e,+}$ ,  $R_{i,W}^{e,-}$  and  $\mathfrak{J}_{i,j,W}^{e,+}$ ,  $\mathfrak{J}_{i,j,W}^{e,-}$  respectively. As detailed previously, the sign denotes if the equations are evaluated in the upper or lower side of the wake.

Thus, (4.7) can be rewritten into:

$$R_{i,W}^{e,+}(\Theta^+) = \sum_{j=1}^M \int_{\mathbb{D}_{W,e}^+} \rho^+ \nabla \hat{N}_i^+ \cdot \nabla \hat{N}_j^+ d\mathbb{D}\Theta_j^+ - \int_{\Gamma_{W,e}^+} \hat{N}_i^+ g d\Gamma \quad (4.9)$$

$$R_{i,W}^{e,-}(\Theta^-) = \sum_{j=1}^M \int_{\mathbb{D}_{W,e}^-} \rho^- \nabla \hat{N}_i^- \cdot \nabla \hat{N}_j^- d\mathbb{D}\Theta_j^- - \int_{\Gamma_{W,e}^-} \hat{N}_i^- g d\Gamma \quad (4.10)$$

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 (4.8), by computing  $\mathfrak{J}_{i,j,W}^{e,+} = \frac{\partial R_{i,W}^{e,+}(\Theta^+)}{\partial \Theta_j^+}$  and  $\mathfrak{J}_{i,j,W}^{e,-} = \frac{\partial R_{i,W}^{e,-}(\Theta^-)}{\partial \Theta_j^-}$ . Here,  $\Theta^+$  and  $\Theta^-$  are composed as:

$$\Theta_i^+ = \begin{cases} \Phi_i & \text{if } i^{\text{th}} \text{ node} \in \mathbb{D}^+ \\ \Psi_i & \text{if } i^{\text{th}} \text{ node} \in \mathbb{D}^- \end{cases} \quad (4.11)$$

$$\Theta_i^- = \begin{cases} \Psi_i & \text{if } i^{\text{th}} \text{ node} \in \mathbb{D}^+ \\ \Phi_i & \text{if } i^{\text{th}} \text{ node} \in \mathbb{D}^- \end{cases} \quad (4.12)$$

This division effectively duplicates the degrees of freedom on the wake elements in order to allow for a discontinuity in the potential, as shown in figure 4. Consider the element in figure 3a, 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  $\Phi_i$ ,  $\Psi_i$ . When evaluating the upper terms of the wake in figure 3b, (4.11) is used, assigning  $\Phi_i$  and  $\Psi_i$  accordingly. If the lower terms are evaluated in figure 3c, (4.12) is used instead.

Boundary conditions for both sides of the wake are imposed by a least-square finite-element approach, combining (4.5) and (4.6) in a single two-dimensional vector equation, by minimising the functional:

$$\Pi(\Theta^+, \Theta^-) = \frac{1}{2} \int_{\Gamma_W} \|\rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^-\|^2 d\Gamma = 0 \quad (4.13)$$

This expression will act as a constraint in the element formulation. The residual terms and its Jacobian are found by deriving the functional in (4.13), where derivatives 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

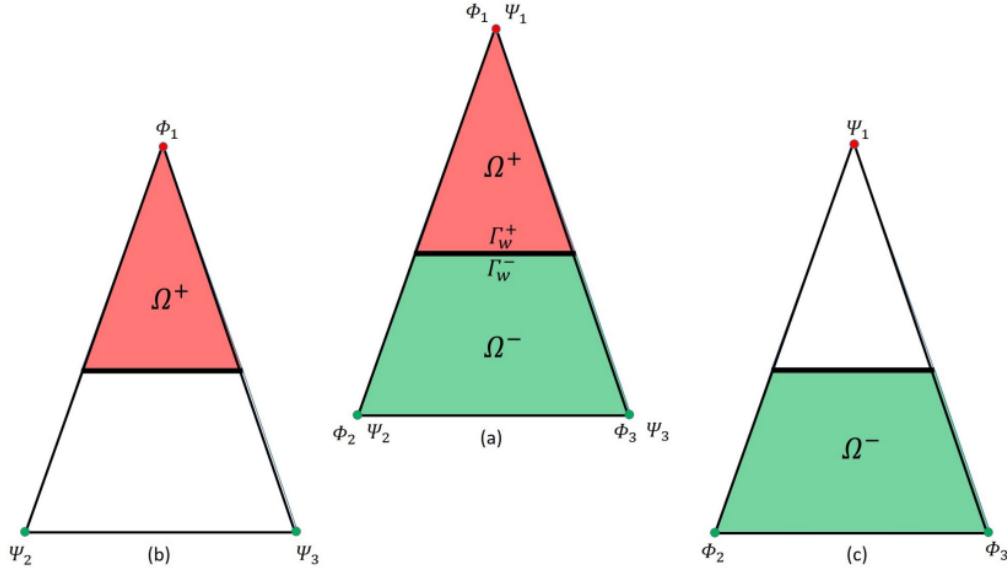


Figure 3: Scheme showing the cutting performed in a wake element from Davari et al. 2019. Wake elements have duplicated degrees of freedom  $\Phi_i, \Psi_i$ , which are divided depending on whether the integration is performed on the upper wake  $\mathbb{D}^+$  or the lower wake  $\mathbb{D}^-$ .

are considered. Here,  $u$  represents the velocity difference between the free-stream unperturbed flow  $V_\infty$ , and the actual velocity around the airfoil  $v$ . The elemental contributions of the residual for the constraint are written with the subscript  $B$  (e.g.  $R_{i,B}^{e,+}$ ) to refer to the terms enforcing the boundary conditions of the wake:

$$R_{i,B}^{e,+}(\Theta) = \frac{\partial \Pi(\Theta)}{\partial \Theta_i^+} = \int_{\Gamma_{W,e}^+} \nabla \hat{N}_i^+ (\rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^-) d\Gamma \quad (4.14)$$

$$R_{i,B}^{e,-}(\Theta) = \frac{\partial \Pi(\Theta)}{\partial \Theta_i^-} = - \int_{\Gamma_{W,e}^-} \nabla \hat{N}_i^- (\rho^+ \nabla \Theta^+ - \rho^- \nabla \Theta^-) d\Gamma \quad (4.15)$$

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

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

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

Since linear elements are considered, their gradients are constant and the boundary integrals can be replaced by volume integrals.

$$B_{i,j}^{e,+} = \frac{\partial R_{i,B}^{e,+}}{\partial \Theta_j^+}(\Theta) = \int_{\mathbb{D}_{W,e}^+} \rho^+ \nabla \hat{N}_i^+ \nabla \hat{N}_j^+ + 2 \frac{\partial \rho^+}{\partial \|v\|^2} (\nabla \hat{N}_j^+ \Theta_i^+) (\nabla \hat{N}_i^+ \Theta_i^+) \quad (4.16)$$

$$B_{i,j}^{e,-} = \frac{\partial R_{i,B}^{e,-}}{\partial \Theta_j^-}(\Theta) = - \int_{\mathbb{D}_{W,e}^-} \rho^- \nabla \hat{N}_i^- \nabla \hat{N}_j^- + 2 \frac{\partial \rho^-}{\partial \|v\|^2} (\nabla \hat{N}_j^- \Theta_i^-) (\nabla \hat{N}_i^- \Theta_i^-) \quad (4.17)$$

The terms in the system of equations obtained in (4.16) and (4.17) are equivalent to those defined by the full-potential equation in (4.8), only adapting the degrees of freedom according to (4.11) and (4.12). If this is expressed in matrix form for a triangular element with the same cutting as the element shown in figure 3, the residuals and Jacobians on the auxiliary degrees of freedom are written as:

$$B_{i,j}^e = (B_{i,j}^{e,+} \quad B_{i,j}^{e,-}) = \begin{pmatrix} B_{11}^+ & B_{12}^+ & B_{13}^+ & B_{11}^- & B_{12}^- & B_{13}^- \\ B_{21}^+ & B_{22}^+ & B_{23}^+ & B_{21}^- & B_{22}^- & B_{23}^- \\ B_{31}^+ & B_{32}^+ & B_{33}^+ & B_{31}^- & B_{32}^- & B_{33}^- \end{pmatrix}$$

$$R_{i,B}^e = \begin{pmatrix} R_{i,B}^{e,+} \\ R_{i,B}^{e,-} \end{pmatrix} = \begin{pmatrix} R_{2,B}^+ \\ R_{3,B}^+ \\ R_{1,B}^- \end{pmatrix}$$

These terms replace the terms in the auxiliary degrees of freedom of (4.9) and (4.10) and their Jacobians, acting as a constraint.

$$\mathfrak{J}_{i,j,W}^e = (\mathfrak{J}_{i,j,W}^{e,+} \quad \mathfrak{J}_{i,j,W}^{e,-}) = \begin{pmatrix} \mathfrak{J}_{11}^+ & \mathfrak{J}_{12}^+ & \mathfrak{J}_{13}^+ & 0 & 0 & 0 \\ B_{11}^+ & B_{12}^+ & B_{13}^+ & B_{11}^- & B_{12}^- & B_{13}^- \\ B_{21}^+ & B_{22}^+ & B_{23}^+ & B_{21}^- & B_{22}^- & B_{23}^- \\ B_{31}^+ & B_{32}^+ & B_{33}^+ & B_{31}^- & B_{32}^- & B_{33}^- \\ 0 & 0 & 0 & \mathfrak{J}_{21}^- & \mathfrak{J}_{22}^- & \mathfrak{J}_{23}^- \\ 0 & 0 & 0 & \mathfrak{J}_{31}^- & \mathfrak{J}_{32}^- & \mathfrak{J}_{33}^- \end{pmatrix}$$

$$R_{i,W}^e = \begin{pmatrix} R_{i,W}^{e,+} \\ R_{i,W}^{e,-} \end{pmatrix} = \begin{pmatrix} R_1^+ \\ R_{2,B}^+ \\ R_{3,B}^+ \\ R_{1,B}^- \\ R_2^- \\ R_3^- \end{pmatrix} \quad \Theta_i = \begin{pmatrix} \Theta_i^+ \\ \Theta_i^- \end{pmatrix} = \begin{pmatrix} \Phi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \Phi_5 \\ \Phi_6 \end{pmatrix}$$

These elemental contributions replace the residuals in (4.9) and (4.10) and its Jacobians, now with the constraint of the boundary conditions of the wake applied in the auxiliary degrees of freedom. In summary, the above approach allows to



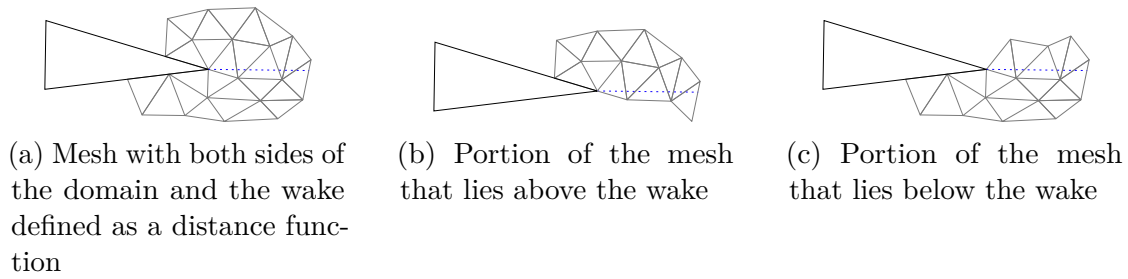


Figure 4: Representation of an embedded wake with body fitted geometry. The wake is defined implicitly through a distance function, and its intersection subdivides the mesh into two, shown in figures 4b and 4c

model a discontinuity on the potential field, while respecting the wake boundary conditions, without the need of explicitly model the wake in the mesh. This will be a key feature for optimisation, where the same mesh will be reused after deformation according to the geometry change in each optimisation step.

## 4.2 Deterministic adjoint equations derivation

The sensitivity calculation of the objective function  $J$  with respect to the design parameters  $z$  is now performed. In this problem,  $z$  is the coordinates of the nodes defining the airfoil shape. A snapshot of a mesh example is shown in figure 6.

The derivation of the adjoint equation follows the dual approach by Giles and Pierce 2000, and it starts with the definition of the sensitivities  $\frac{dQ}{dz}$ .

$$\frac{dQ}{dz} = \frac{\partial Q}{\partial z} + \frac{\partial Q}{\partial u} \frac{du}{dz} \quad (4.18)$$

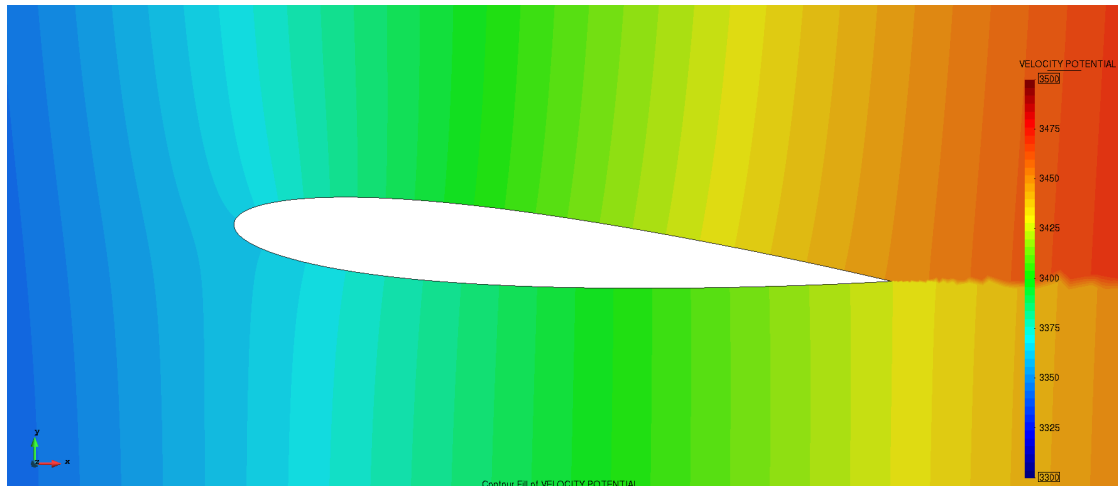
since the residual of the full potential equation need to be satisfied for all considered set of  $z$ , the derivative of the residual with respect to the geometry parameters is zero.

$$F(u, z) = 0 \quad \forall u, z$$

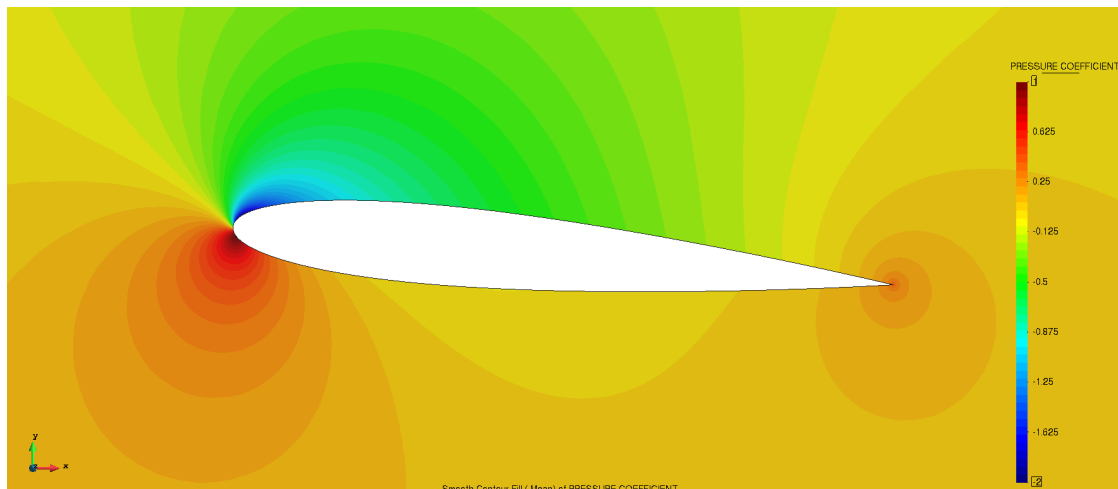
$$\frac{dF}{dz} = \frac{\partial F}{\partial z} + \frac{\partial F}{\partial u} \frac{du}{dz} = 0$$

This allows to find the rate of change of the primal variables with respect to the geometry parameters.

$$\frac{du}{dz} = -\frac{\partial F^{-1}}{\partial u} \frac{\partial F}{\partial z}$$



(a) Potential field on a NACA0012 airfoil at  $5^\circ$ . The discontinuity on the potential caused by the wake can be seen on the contour fill



(b) Pressure field on a NACA0012 airfoil at  $5^\circ$ . The pressure field is computed using the gradient of the potential field and Bernoulli's equation

Figure 5: Contour fills of the primal solution for a NACA0012 airfoil

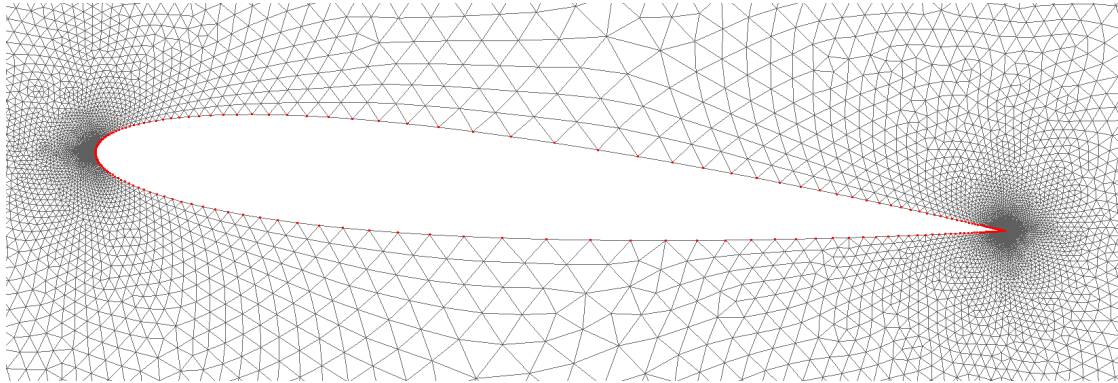


Figure 6: Mesh of a body fitted NACA0012 airfoil. The nodes marked with red define the shape of the airfoil and they are also the geometry parameters considered in the problem

which is replaced in the expression of the sensitivities in (4.18).

$$\frac{dQ}{dz} = \frac{\partial Q}{\partial z} - \frac{\partial Q}{\partial u} \frac{\partial F^{-1}}{\partial u} \frac{\partial F}{\partial z}$$

where the term  $\frac{\partial Q}{\partial u} \frac{\partial F^{-1}}{\partial u}$  can be seen as a the solution of the linear problem for some variable  $\lambda$  given by:

$$\frac{\partial Q}{\partial u} = - \left( \frac{\partial F}{\partial u} \right)^\top \lambda \quad (4.19)$$

whose solution  $\lambda$  is used to assemble the final sensitivities and the partial derivatives on  $z$  with:

$$\frac{dQ}{dz} = \frac{\partial Q}{\partial z} + \lambda^\top \frac{\partial F}{\partial z}.$$

Equation (4.19) is known as the ‘adjoint equation’.

#### 4.2.1 Partial derivatives of the physical problem

Next, the terms involving the adjoint equations are derived. The derivative of the residual of the problem with respect to the primal variables is the Jacobian of the problem defined in (4.8), also known as the left-hand side term, where the primal variable is the potential  $u = \Phi$ .

$$\frac{\partial F(u, z)}{\partial u} = \frac{\partial F(\Phi, z)}{\partial \Phi} = \mathfrak{J} \quad (4.20)$$

The derivative of the residual with respect to the geometry parameters  $z$  is computed by finite differences for some perturbation  $\varepsilon$ .

$$\frac{\partial F(u, z)}{\partial z} = \frac{\partial F(\Phi, z)}{\partial z} = \frac{F(\Phi, z + \varepsilon) - F(\Phi, z)}{\varepsilon}. \quad (4.21)$$

Evaluating the residual for a new set of geometry parameters,  $F(\Phi, z + \varepsilon)$ , does not require to compute the solution of the whole analysis, since the potential field remains constant. In this case, the same potential field  $\Phi$  obtained in the primal analysis is used to compute the Jacobians  $F(\Phi, z + \varepsilon)$ , but the coordinates  $z$  are different. This simply requires to recompute the shape function and shape function derivatives to evaluate the term  $F(\Phi, z + \varepsilon)$ , which is performed elementally. Note that these partial derivatives are also applicable to the wake boundary elements, where  $F$  is replaced by (4.14) and (4.15) and the partial derivative  $\frac{\partial F(u, z)}{\partial u}$  by the Jacobians in (4.16) and (4.17).

#### 4.2.2 Partial derivatives of the lift coefficient

Following the potential flow formulation, the lift force can be expressed in terms of the free-stream quantities by the Kutta–Joukowski theorem for streamlined bodies Anderson and Venkatakrisnan 1999.

$$L = \rho_{\infty} V_{\infty} \Gamma,$$

where  $\Gamma$  is the circulation of the flow and the subscript ‘ $\infty$ ’ denotes free-stream quantities. It can be shown that  $\Gamma$  can be expressed as the potential jump across the domain  $\Gamma = \Phi^+ - \Phi^-$  Nishida and Drela 1995. The lift force can be divided by a reference pressure to retrieve the lift coefficient.

$$C_l = \frac{L}{\frac{1}{2} \rho_{\infty} V_{\infty}^2 c} = \frac{2}{V_{\infty} c} (\Phi^+ - \Phi^-)$$

The derivative of the objective function with respect to the primal variables is computed analytically from the potential-jump expression of the lift coefficient.

$$\frac{\partial C_l}{\partial u} = \frac{\partial C_l}{\partial \Phi} = \frac{2}{V_{\infty} c} \times \begin{cases} 1 & \text{if } \Phi = \Phi^+ \\ -1 & \text{if } \Phi = \Phi^- \end{cases}$$

In order to compute the sensitivities of this objective function with respect to the geometry parameters,  $\frac{dC_l}{dz}$ , an adjoint problem is solved. In a deterministic case, these sensitivities are computed as:

$$\frac{dC_l}{dz} = \frac{\partial C_l}{\partial z} - \lambda^{\top} \frac{\partial F(u, z)}{\partial z} \quad (4.22)$$

Here  $F$  is the residual of the potential flow problem whose partial derivative are computed in (4.20) and (4.21).  $\lambda$  is the set of adjoint variables, which are computed by solving the adjoint equation:

$$-\left(\frac{\partial F(\Phi, z)}{\partial \Phi}\right)^\top \lambda = \frac{\partial C_l}{\partial \Phi}$$

Finally, the partial derivative of the objective function with respect to the geometry parameters is null in this case, since the expression of the lift coefficient for the potential jump does not depend explicitly on the geometry parameters.

$$\frac{\partial C_l}{\partial z} = \frac{\partial}{\partial z} \left( \frac{2}{V_\infty c} (\Phi^+ - \Phi^-) \right) = 0$$

Assembling the partial derivatives, the final sensitivities are computed using (4.22).

The values of the sensitivities can now be used to update the geometry and advance in the design space. However, using the sensitivities as they are would result in mesh irregularity after a few iterations. For this reason, the Vertex-Morphing method (Hojjat et al. 2014) is employed to compute the actual shape update at every iteration, which smooths the shape update in order to preserve mesh quality. This approach ensures the same solution is achieved on convex optimisation problems independently of the filter radius. However, for non-convex optimisation problems, which is the common case in engineering, this approach may lead to some local minimum depending on the filtering size chosen. For this problem, a filter radius that lies between half the chord and the thickness of the airfoil is chosen.

### 4.2.3 Partial derivatives of volume and perimeter functionals

The volume of the domain can be used either as an objective function, where typically the volume aims to be minimised, or as a constraint, where the geometry of study is restricted to not be greater or less than a certain value. The expression of the volume results from the integral of the domain, which is computed using the shape functions,

$$V = \int_{\mathbb{D}} d\mathbb{D}. \quad (4.23)$$

The volume gradient is computed by finite differences for some perturbation  $\varepsilon$ :

$$\frac{\partial V(z)}{\partial z} = \frac{V(z + \varepsilon) - V(z)}{\varepsilon}$$

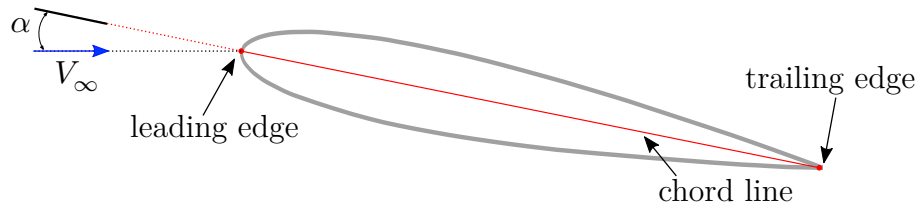


Figure 7: Scheme showing some of the main characteristic and attributes on an airfoil, which play an important role in the optimisation problem, showcasing the free stream velocity  $V_\infty$ , the angle of attack  $\alpha$ , and the chord line, defined by the leading and trailing edge

The perimeter can also be used as an objective function or constraint. The expressions and utilities used for the volume can be used to compute the perimeter, but applied on the boundary surfaces.

$$P = \int_{\Gamma_g} d\Gamma_g \quad (4.24)$$

where  $\Gamma_g$  refers to the boundary enclosing the geometry of the airfoil. The gradient of the perimeter is also computed by finite differences.

$$\frac{\partial P(z)}{\partial z} = \frac{P(z + \varepsilon) - P(z)}{\varepsilon}$$

#### 4.2.4 Partial derivatives on airfoil geometry definitions

In the context of airfoil optimisation, it is important to define some of the geometry aspects that come into play and that define the airfoil shape and condition. In the following figure, a highlight of the most important features of an airfoil is presented. The chord line is the line defined by both the leading edge and the trailing edge. The angle between the chord line and the free stream direction is known as the ‘angle of attack’.

Both the chord and the angle of attack will be enforced as constant, as a change of angle of attack or a different chord length changes drastically the airfoil nature. It is desired that the optimal shape obtained is some variation of the initial airfoil which keeps constant the initial angle of attack and the chord of the airfoil.

**Angle of attack** As stated above, the first geometry definition that will be considered is the angle of attack of the airfoil. The angle of attack is defined as the angle between the inflow velocity and the chord line. Let  $v$  be the inflow velocity and  $c$  the chord vector, defined as the position of the leading edge and the trailing

edge, with the trailing edge as origin. The angle of attack is then defined as the angle between these two vectors:

$$\alpha = \arccos\left(\frac{v \cdot c}{\|v\| \|c\|}\right) \quad (4.25)$$

The gradient of the angle of attack will only depend on the position of the trailing edge and the leading edge nodes. It can easily be computed by finite differences.

$$\frac{\partial \alpha}{\partial z} = \frac{\alpha(z + \varepsilon) - \alpha(z)}{\varepsilon}$$

**Chord line length** Another feature that is important to preserve across the design space is the chord length, that can be defined as the norm of the chord vector, represented by the leading and trailing edge nodes.

$$l_c = \|c\| \quad (4.26)$$

And its gradient is easily computed by finite differences:

$$\frac{\partial l_c}{\partial z} = \frac{l_c(z + \varepsilon) - l_c(z)}{\varepsilon}$$

**Optimisation problem** In this setup, the desired optimisation problem is to maximise lift while keeping constant the airfoil characteristics and volume.

**Problem 4** (Deterministic maximisation of the lift). For convenience, the optimisation problem will be defined as the minimisation of the negative of the lift coefficient:

$$z_\star := \operatorname{argmin}\{J(z) : z \in Z, F(u, z) = 0, \mathbf{K}(z) = 0\},$$

with  $J(z) = Q(z) := -C_l$ .

In this scenario, four constraints  $\mathbf{K}(z) = 0$  are applied, where the subscript ‘0’ denotes the initial values of the quantities to be constrained:

$$\begin{aligned} K_1(z) &= V_0 - \int_{\mathbb{D}} d\mathbb{D} \\ K_2(z) &= P_0 - \int_{\Gamma_g} d\Gamma_g \\ K_3(z) &= \alpha_0 - \arccos\left(\frac{v \cdot c}{\|v\| \|c\|}\right) \\ K_4(z) &= l_{c0} - \|c\| \end{aligned}$$

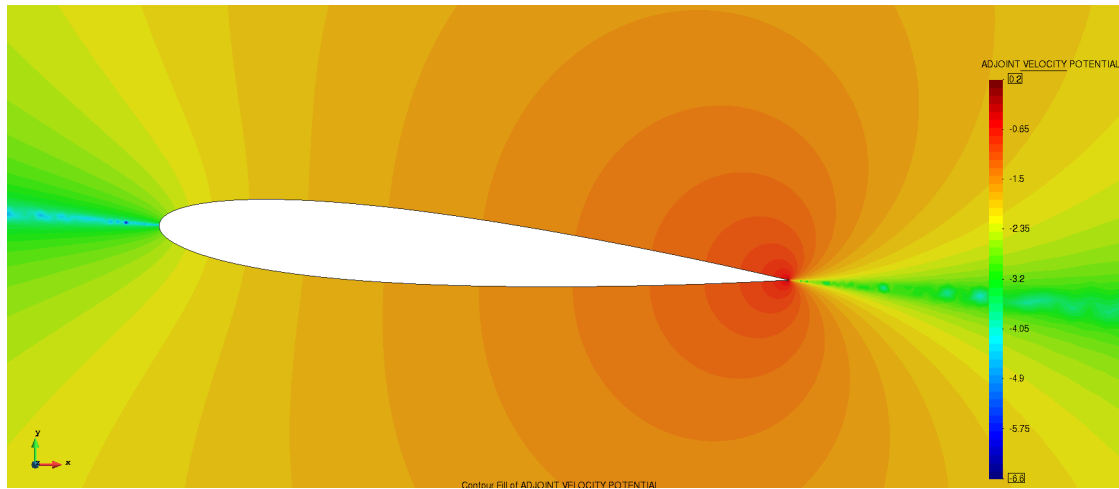
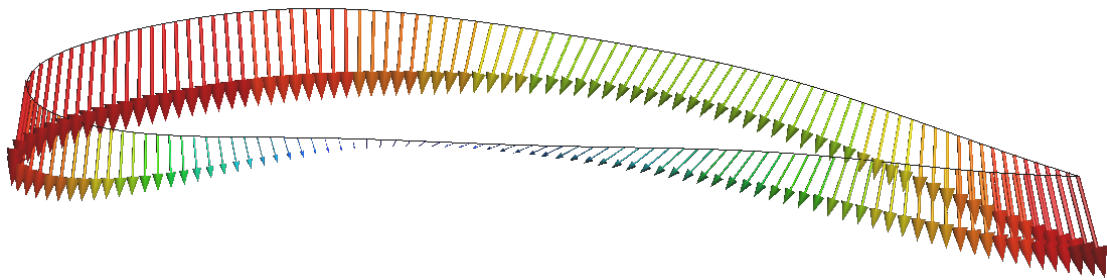
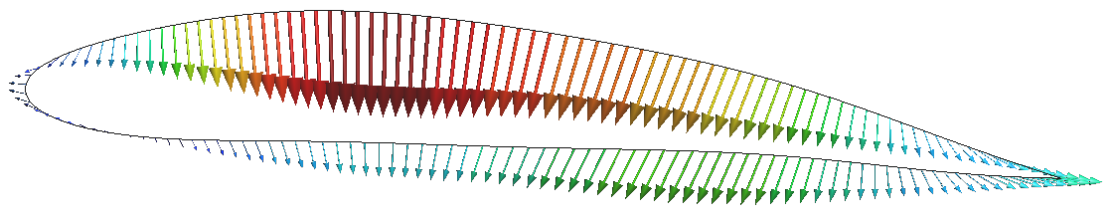


Figure 8: Adjoint field  $\lambda$  corresponding to the resolution of the adjoint problem with the lift coefficient as quantity of interest. This field was obtained by solving an extra linear problem comparable to the primal one, and it is an intermediate step towards obtaining the final sensitivities, as seen in (4.2)



(a) Shape update on the airfoil geometry on a gradient descent step



(b) Shape update on the airfoil geometry with the constraints activated on the trust region algorithm

Figure 9: Two snapshots of the optimisation problem showcased in the deliverable, corresponding to step 4 and step 6 of the optimisation loop. On 9a, the problem is driven by the objective function only, and the shape update is focusing on minimising it. On 9b, the constraints are activated and the shape update is trying to correct the current shape to satisfy the constraints



where the volume, perimeter, angle of attack and chord lengths have been substituted by (4.23), (4.24), (4.25) and (4.26) respectively.  $\diamond$

A trust-region optimisation algorithm is used to solve the problem. This algorithm works as a classical gradient-descent method if the constraints are satisfied. In case a constraint is not satisfied, the gradient is corrected using the gradient of the constraints to correct the shape update and respect the constraints.

### 4.3 Results on deterministic optimisation

An example of an optimisation problem using the lift sensitivities together with the geometric constraints defined in the previous section is shown next, for a NACA0012 profile with an angle of attack  $\alpha = 5^\circ$ . The problem has been solved using the Vertex Morphing method using a Gaussian filter with radius  $r_{\text{fil}} = 0.3c$ , i.e. 30% of the airfoil chord.

In figure 10a the evolution of the objective function with the number of optimisation steps is shown, reaching convergence of the algorithm after 121 iterations. The convergence criteria used is the relative change of the objective function of a given step with its previous one, with a tolerance set 0.1%. The shape update is controlled by a step size, that is reduced in half when the objective function oscillates. The initial step size was set to  $k = 0.1$ .

In figure 11, a comparison between the initial and optimal shape is shown, as well as the pressure distribution in both cases.

### 4.4 Prospects on optimisation under uncertainty

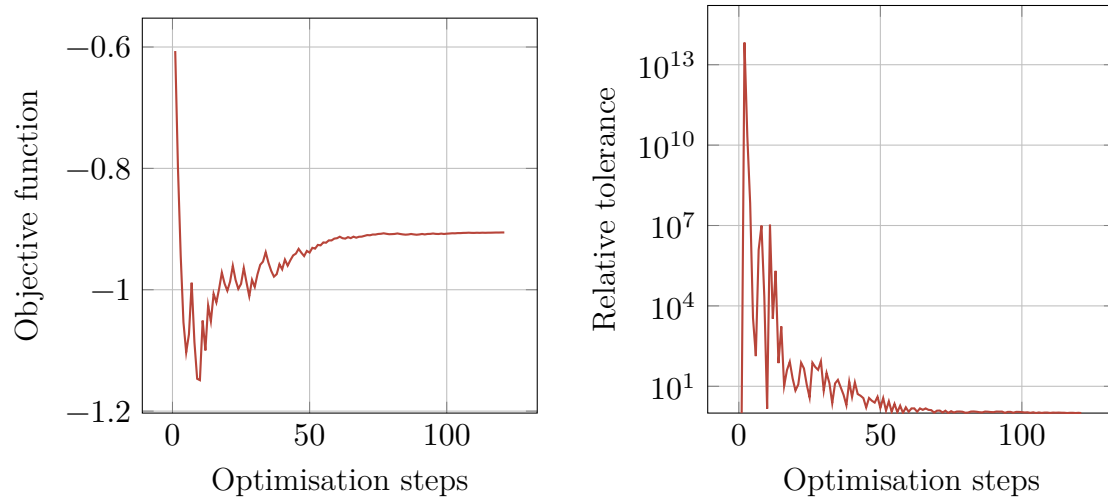
Let  $(\mathbf{V}_\infty, \Phi_\infty)$  be uncertain parameters, following a known distribution. The deterministic optimisation problem 4 becomes a problem of OUU, which we reformulate similarly to problem 1.

**Problem 5** (Airfoil shape for CVAR-optimal lift). We use the notations from problem 4. The lift coefficient  $C_l =: -Q(z)$  is now a random variable, and we wish to minimise  $\text{CVAR}_\beta(-C_l)$ . We define the optimal airfoil as  $z_*$  such that

$$(z_*, s_*) := \operatorname{argmin}\{J(z, s) := \phi(Q(z), s) : (z, s) \in Z \times \mathbb{R}, F(u, z) = 0, \mathbf{K}(z) = 0\}$$

$\diamond$

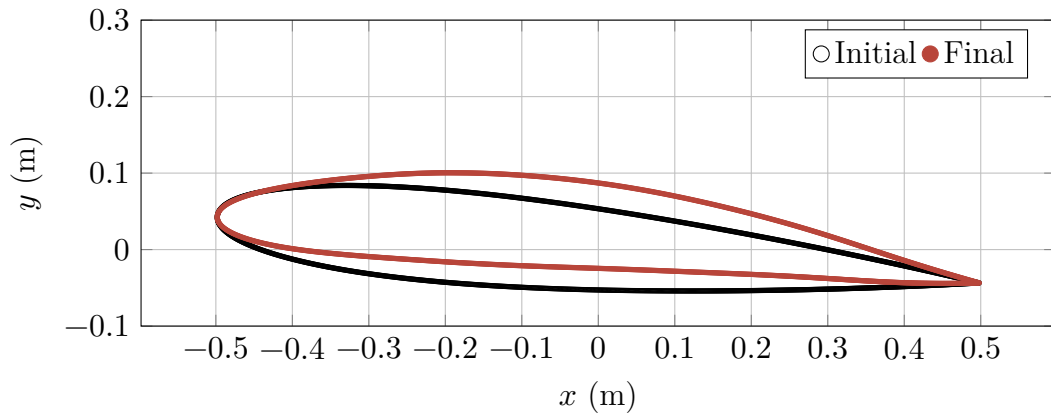
Like problem 3 for the Van der Pol oscillator (from § 3), this problem falls into the category of OUU problems represented by problem 5 and discussed in § 2, except for the constraints represented by  $\mathbf{K}$ . Let us note  $\mathbf{K}^{-1}(0) \subset Z$  the set of



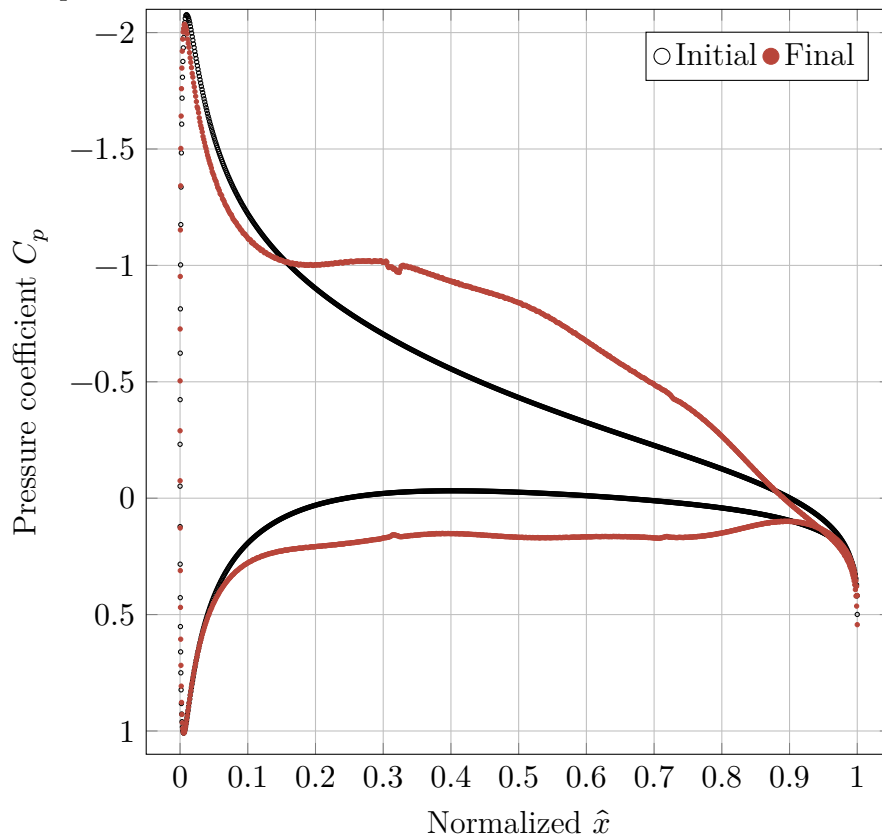
(a) Evolution of the objective function with the number of optimisation steps. The objective function is the negative of the lift coefficient

(b) Evolution of the stopping criteria through the optimisation loop, which is the relative tolerance of the objective function. When the relative change drops the threshold set by the user, the optimisation loop is stopped.

Figure 10: Evolution of the objective function and the relative tolerance in the optimisation problem showcased. Relative tolerance of 0.1 %; convergence reached after 121 iterations.



(a) Shape change on the initial against the final configuration after the end of the optimisation problem



(b) Pressure distribution on the initial and optimised airfoil, which produces the increase in lift.  $\hat{x}$  refers to the normalised position of the airfoil coordinated with the chord

Figure 11: Example of the results obtained on the resolution of an optimisation problem using the lift sensitivities, with volume, angle of attack and chord length as constraints. The optimal shape was achieved after 121 iterations.

admissible shapes and  $\Pi_{\mathbf{K}^{-1}(0)} : Z \rightarrow \mathbf{K}^{-1}(0)$  a projector onto that set. A simple approach to account for these constraints in algorithm 2 would be to replace l. 11 with

$$z_{k+1} := \Pi_{\mathbf{K}^{-1}(0)}(z_k + \gamma_k p).$$

The rest of algorithm 2 can be used verbatim to solve problem 5, since adjoint-based sensitivities are available. Instead of the time discretisation, the levels in the MLMC method would be defined by the spatial discretisation, i.e. the refinement of the mesh. After being implemented and tested on the simpler Van der Pol oscillator, algorithm 2 will be applied to this benchmark problem.



Figure 12: Absolute World twin-tower skyscraper complex. Mississauga, Ontario, Canada

## 5 Unsteady wind around a building

### 5.1 Drag of a twisted, tapered building

The past decades have seen increasingly-taller building structures built all around the world. Lateral loads like that of external wind force play a major role in deciding the structural system of these buildings. As the wind load attracted by the building increases with greater height, many mitigation strategies have been used in practice. Geometrical modification of the building shape is found to be effective in reducing the wind load on tall buildings. These include global geometrical modifications and local geometrical modifications. Various methods for global geometrical modifications exist: tapering, twisting, providing openings, providing stepping, et cætera (etc.). Herein, two such global geometrical modifications are applied to an elliptical tower. The twist along the axis and the tapering of the structure are the design parameters and are optimised to minimise the total force at the base of the building. Here, the geometry is motivated from the [Absolute World towers](#) shown in figure 12.

#### 5.1.1 Uncertainties in the incoming wind flow

Wind is associated with inherent uncertainties. As is common in the wind-engineering community, we model the natural wind effects in the atmospheric boundary layer (ABL) by decomposing the incoming wind velocity into its stationary *mean profile*  $\bar{u}$  and its unsteady *turbulent fluctuations*  $u'$ :  $u = \bar{u} + u'$ .

The mean wind profile is a contribution to the velocity field which changes over the span of several hours or days. On the contrary, the turbulent fluctuations in wind introduce short-term gusts with a life span ranging from few seconds to several minutes. However, the mean wind velocity and direction are also associated with uncertainties. The wind profile along the height is associated with the uncertainties in the terrain categories. During analysis, one needs to take into account the many uncertain parameters in the inlet conditions of a typical problem of CFD. Although the fluctuations will eventually be modelled and included in the simulations, they have been ignored in the work reported here. The wind inlet consists exclusively of the mean profile, which we proceed to describe below.

Most of the buildings reside entirely within the ABL. An ABL is characterised by constant shear stress in the vertical direction (Kaimal and Finnigan 1994). The ABL is *neutrally stable* at high wind speeds. Ground friction is majorly contributed from pressure drag – a force generated by pressure differences near the surface and is caused by wind flowing across obstacles at the surface. The ground friction depends largely on the type of the terrain and the size of the obstacles. The various terrain types are classified into terrain categories in codes and standards (Structural Eurocode 2005).

In this study, the local terrain type is assumed to be sufficiently characterised by a single *roughness length* parameter  $\bar{x}_3 > 0$ . The mean value of this parameter for each of the terrain categories can be found in the design codes. In the current study, the range of values of this parameter for each terrain category is taken from *Probabilistic model code* (Joint Committee on Structural Safety 2001). In addition, our mean-profile model incorporates the *friction velocity*  $u_*$ , which can be derived from the shear stress on the ground  $\tau_0$  by the simple formula  $\tau_0 = \rho u_*^2$ , and the *incidence wind angle*  $\theta$ .

Let  $\mathbb{D}$  denote a section of the ABL lying above a flat section of Earth's surface, parameterised by the Cartesian coordinates  $(x_1, x_2, x_3)$ . Under the assumptions of neutral stability and homogeneous roughness, the mean velocity  $\bar{u} = \bar{u}(x_3)$  can be modelled by the following logarithmic profile (Kareem and Tamura 2015):

$$\bar{u} = \frac{u_*}{\kappa} \ln \left( \frac{x_3}{\bar{x}_3} + 1 \right) e(\theta), \quad (5.1)$$

$$\text{where } e(\theta) := (\cos(\theta), \sin(\theta), 0) \in \mathbb{R}^3$$

is a unit normal vector denoting the mean wind direction, and  $\kappa \approx 0.41$  is the von Karmán constant.

Each of the parameters  $u_*$ ,  $\theta$ , and  $\bar{x}_3$  in (5.1) are random variables. It is often assumed (see *ibid.*) that the friction velocity  $u_*$ , averaged over all angles  $\theta$ , obeys a Weibull distribution  $\mathcal{W}(\lambda, k)$ , with *scale*  $\lambda$  and *shape*  $k$ . Likewise, the variation of the wind angle  $\theta$  is assumed uniform. In this study, we assume

Table 1: Fluid properties and building details

Quantity	Value	Unit
Density	1.225	$\text{kg m}^{-3}$
Viscosity	$1.507 \cdot 10^{-5}$	$\text{m}^2 \text{s}^{-1}$
Dynamic viscosity	$1.846 \cdot 10^{-5}$	$\text{N s m}^{-2}$
Reynolds number	$9.7 \cdot 10^7$	
Characteristic length	45	m
Height of the building	180	m
Major base diameter	45	m
Minor base diameter	30	m

that  $\bar{x}_3 \sim \mathcal{U}(x_3^L, x_3^U)$ , where  $x_3^L$  and  $x_3^U$  are positive constants inferred from the engineering code book *Probabilistic model code*. If  $u_*$ ,  $\theta$ , and  $\bar{x}_3$  are independent random variables, then this is enough to form a complete, parameterised statistical model for  $\bar{u}$ .

### 5.1.2 Details of numerical simulation

Table 1 lists the values used for the fluid properties and the building characteristics. The large Reynolds number in the current simulation makes the flow highly turbulent. Since natural wind is being considered in this work, the incompressible Navier–Stokes model used for the wind flow.

**Problem 6** (Incompressible Navier–Stokes flow). The incompressible Navier–Stokes equations state the conservation of mass and linear momentum in a fluid domain  $\mathbb{D}$ . The mass conservation reads

$$\nabla \cdot u = 0 \quad (5.2)$$

where  $u$  is the fluid velocity. The momentum conservation is expressed as

$$\rho \partial_t u + \rho u \cdot \nabla u - \nabla \cdot \sigma = f, \quad (5.3)$$

where  $\rho$  is the fluid density,  $\sigma$  is the stress tensor, and  $f$  is the external force acting on  $\mathbb{D}$ . Suitable initial and boundary conditions need to be prescribed for (5.2) and (5.3). The initial and boundary conditions for the problem are

$$\begin{aligned} u &= u_0 && \text{over } \mathbb{D} \times \{0\} \\ u &= u_D && \text{over } \Gamma_D \times ]0, T] \\ \sigma \cdot \mathbf{n} &= \mathbf{t}_r && \text{over } \Gamma_N \times ]0, T] \end{aligned}$$

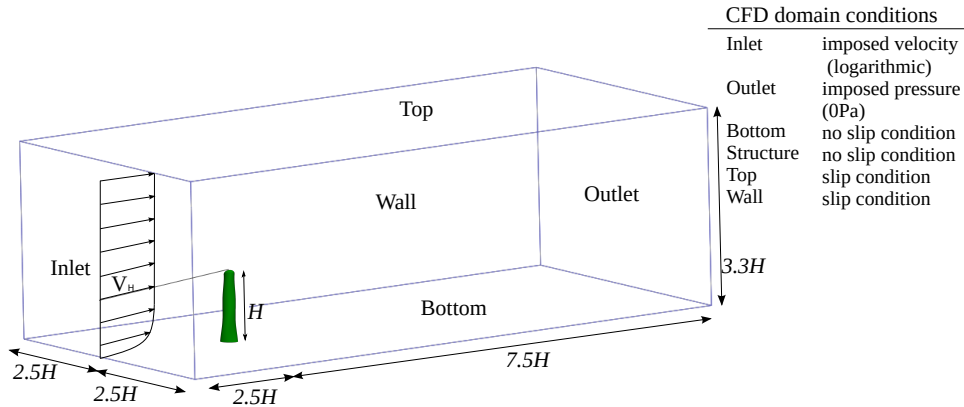


Figure 13: Details of the simulation domain and boundary conditions

where  $u_0$  is the initial velocity field;  $u_D$  is the imposed velocity on the Dirichlet boundary  $\Gamma_D$ ;  $\mathbf{n}$  is the outer normal vector;  $\mathbf{t}_r$  is the imposed traction acting along the Neumann boundary  $\Gamma_N$ .  $\diamond$

The wind flow around the building is simulated using CFD analysis with the open-source Kratos Multiphysics solver (Ferrándiz, Bucher et al. 2020). This involves a finite-element method (FEM) for flow problems based upon a variational multi-scale (VMS) formulation from Cotela-Dalmau et al. 2016. The fluid domain is modelled with *fractional step elements*. The computational domain is detailed in the figure 13, along with the boundary conditions. The blockage ratio<sup>5</sup> is less than 0.8%.

The geometry is meshed with different refinement zones and the details of the adopted meshes are shown in figure 14. The general mesh structure is preserved for all the geometries and optimisation steps. Within this work, the geometry subject to shape optimisation encounters shape updates driven by the gradient-descent optimisation algorithm employed. The embedded CFD approach as proposed in Zorrilla et al. 2019 is herein used in the preliminary results. However, for the simulations with a high Reynolds number, a very fine mesh resolution is required for the representation of the building geometry and hence a body-fitted approach with mesh generation is also being explored.

In the original work of *ibid.*, the constraints are generally formulated for the weak imposition of slip conditions along the boundary of the embedded geometry. The latter constraint is formulated by means of an auxiliary parameter whose limit towards zero recovers the no-slip condition which is of interest within this study.

<sup>5</sup>Ratio of the projected area of the structure along the flow direction to the cross-sectional area of the domain



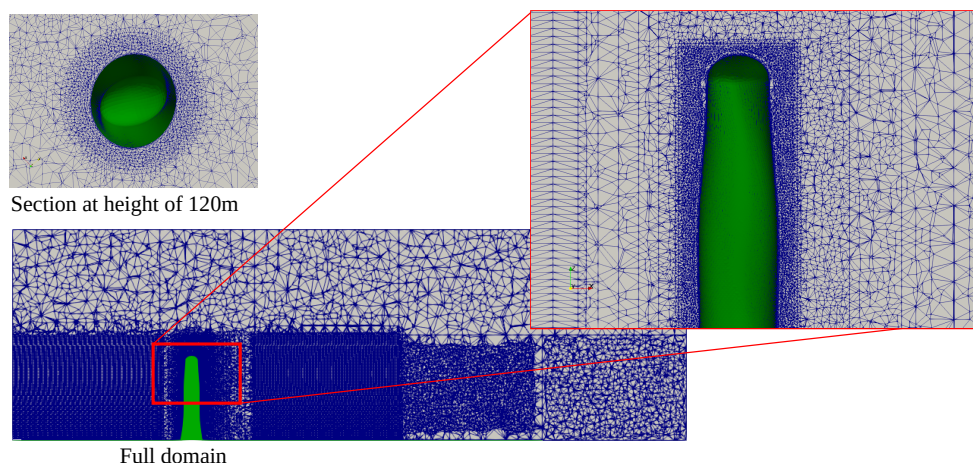


Figure 14: Details of adopted meshes

The corresponding variational formulation of the VMS-stabilised Navier–Stokes equations is extended by Nitsche-type terms accounting for a weakly-zero-velocity field along the boundary of the embedded geometry. The formulation of the momentum conservation includes a penalty term on the embedded boundary to constrain the normal projection of velocity in order to ensure non-penetration (see Zorrilla et al. 2019, for details).

## 5.2 Minimisation of the expected base moment

### 5.2.1 Description of the optimisation problem

**Geometry of the building** A tall building of height 180 m is used for the problem. The two design parameters that can be optimised are the twist of the building and the tapering of the building. These two parameters are shown in figure 15. Twist denotes how much the top cross section is rotated around vertical axis in the clockwise direction with respect to the cross section at the base. The major and minor axis of each of the cross section is changed along the height in such a way that the area of cross section remains the same. This indicates that the floor area of the building (i.e. the usable space) remains the same for the initial and the optimal geometry. Tapering along the height indicates how much the minor axis length is changed along the height. The major axis length is constrained to this quantity through the area.

Consequently, the design space of this problem is  $\mathbb{R}^2 =: Z$ . It parameterises the building surface  $B(z)$ . We denote by  $\mathcal{A} := [0, 2\pi[ \times ]0, +\infty[ \subset Z$  the set of admissible designs.

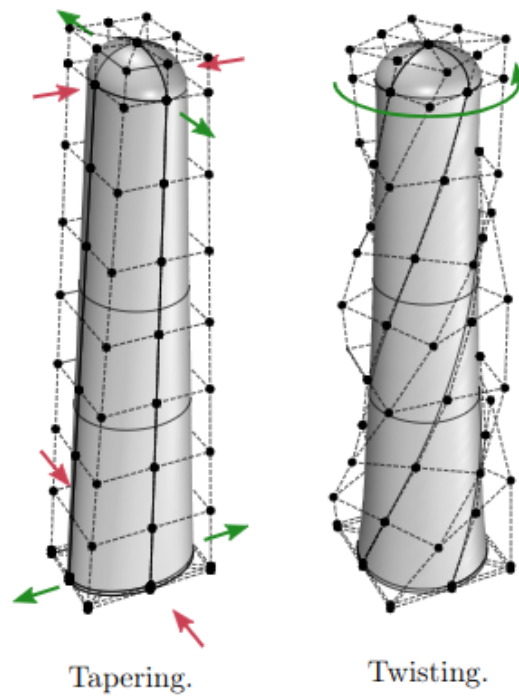


Figure 15: Details of tower geometry and design parameters

**Quantities of interest** A complete and accurate modelling and analysis of the building geometry under uncertain wind is required as the wind load are complex and flow pattern is greatly influenced by the geometrical features of the building. Two design criteria need to be met in the design of the building, namely the strength and serviceability criteria. The strength criterion guarantees that the building will not fail under the extreme design load with a specific probability. The serviceability criterion ensures the comfort of the occupants in the building. These are the displacement and acceleration of the building are below certain threshold with a specific probability. The quantities of interest discussed in the current study relate to the strength criterion of the design; they are of interest to the design of the foundation and structural elements of the building. A reduction in these values indicate a reduction in total cost of the building.

Two important quantities regarding the strength criterion are the force and the base moment. For any given design  $z \in Z$ , the force applied by the fluid at any point over the surface of structure  $B(z)$  is

$$\mathbf{F}(z) = \int_{B(z)} p \mathbf{n} dS,$$

where  $\mathbf{n}$  is the normal to the surface  $dS$  and  $p$  is the pressure solution of problem 6; the values from table 1 still apply. Similarly, the fluid pressure creates at the base  $O$  of the building the mechanical moment

$$\mathbf{M}_O(z) = \int_{B(z)} (\mathbf{x} - O) \times p(\mathbf{x}) \mathbf{n}(\mathbf{x}) dS(\mathbf{x}).$$

NB  $p$  depends directly on the design as well. Since  $p$  depends on random events, so do  $\mathbf{F}(z)$  and  $\mathbf{M}(z)$ .

**Formulation of the optimisation problem** In this application, so far we have considered the following minimisation problem.

**Problem 7** (Minimal expected base moment). Let us recall the previous definitions of  $\mathcal{A}$ ,  $B$  and  $\mathbf{M}_O$ . We consider as quantity of interest the time average of the magnitude of the base moment:

$$Q(z) := \langle \|\mathbf{M}_O(z)\|^2 \rangle_T. \quad (5.4)$$

The optimal design  $z_\star \in Z$  is defined as the one minimising the expected value of this random quantity:

$$z_\star := \operatorname{argmin}\{J(z) := \mathbb{E}(Q(z)) : z \in \mathcal{A}\}.$$

◇

A critical drawback of the choice of risk measure in this problem is that the expectation does not account for extreme loading conditions. This shortcoming will be discussed in § 5.3.

### 5.2.2 Preliminary results on optimisation under uncertainties

Some preliminary results are included in this section. The simulations are run only for a time window  $T = 90$  s. At a given design  $z$ , we choose to approximate  $J(z)$  from  $m \in \mathbb{N}$  i.i.d. samples with a MC estimator  $\mu_m$ ; this is one of the common procedures used in stochastic optimisation (Kouri and Shapiro 2018). To describe the procedure, we begin by using problem 7 as an example.

At each iteration  $k \in \mathbb{N}$ , we choose a number of samples  $m_k$ . Let  $Q^{(i)}(z_k)$  be independent random variables following the same probability distribution as  $Q(z_k)$ . We estimate

$$J(z_k) \approx \mu_{m_k}(Q(z_k)) = \frac{1}{m_k} \sum_{i=1}^{m_k} Q^{(i)}(z_k)$$

The gradient of each sample,  $\nabla Q^{(i)}(z_k)$ , is estimated via finite differences with a first-order, forward scheme:

$$\Delta_{h, e_i} Q^{(i)}(z_k) := \frac{Q^{(i)}(z_k + h e_i) - Q^{(i)}(z_k)}{h} \approx e_i \cdot \nabla Q^{(i)}(z_k).$$

From these we get an approximation  $\Delta_h Q^{(i)}(z_k)$  of  $\nabla Q^{(i)}(z_k)$ , similarly to what was described in § 2.1 p. 10. This involves  $1 + \dim(Z)$  numerical fluid simulations per sample. Due to the chaotic nature of the flow, an accurate gradient for an individual sample requires a long time interval (Lea et al. 2000), the exact length of which is problem-dependent. In addition, special care must be taken to select an appropriate finite difference increment in each component of design space.

Once each of the individual estimations  $\Delta_h Q^{(i)}(z_k)$  have been computed, we estimate the descent direction as

$$\mu_{m_k}(\Delta_h Q(z_k)) \approx \nabla J(z_k).$$

Then we choose a step size  $\gamma_k \in ]0, +\infty[$  and update the design in a steepest-descent fashion:

$$z_{k+1} = z_k - \gamma_k \mu_{m_k}(\Delta_h Q(z_k)).$$

The initial and final design are shown in figure 16. The objective function and the design parameters are shown in figure 17. The CVaR is also computed for each of the optimisation steps along with the expectation.

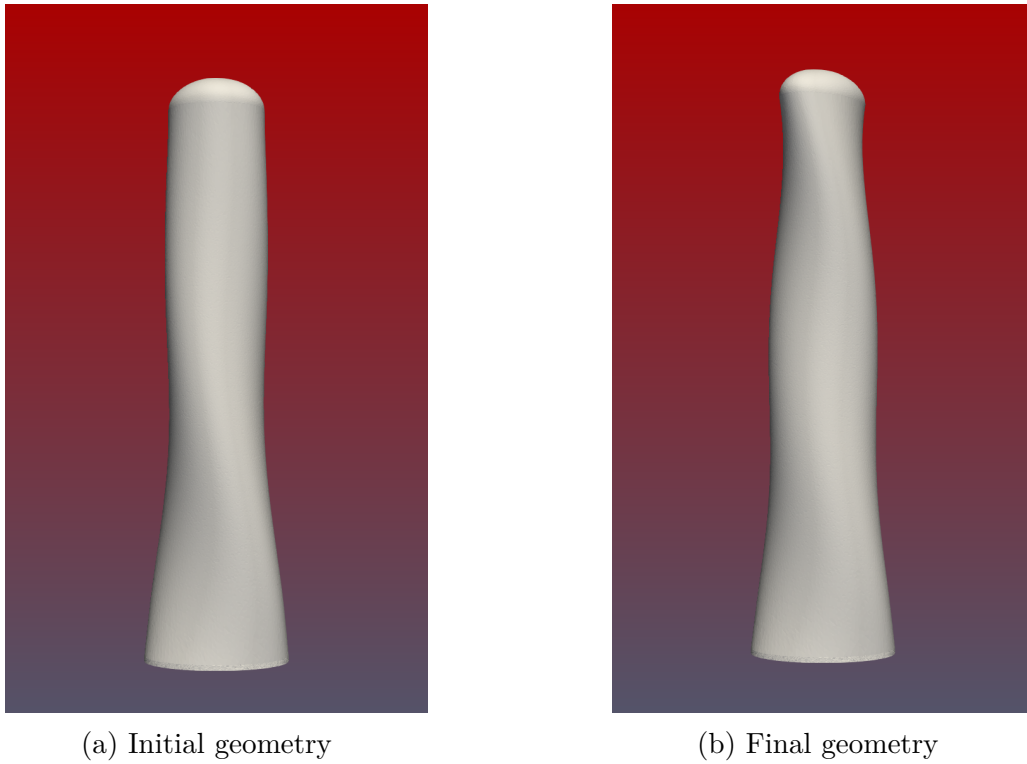


Figure 16: Geometry modifications from optimisation

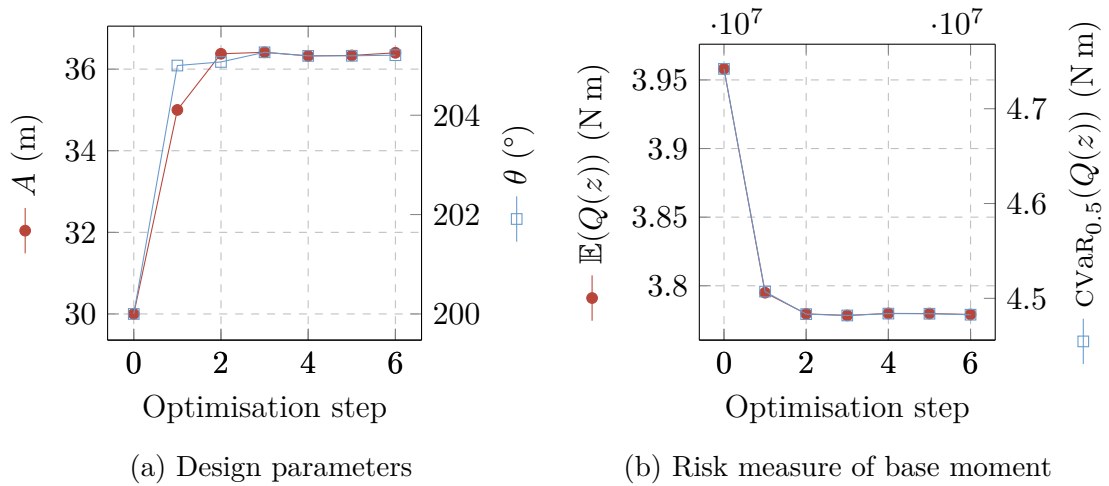


Figure 17: Optimisation results

### 5.3 Prospects on CVaR optimisation

In many applications, the CVaR is a more realistic measure of risk than the VaR because controlling expected failure states is more important than simply controlling the most optimistic failure state. Indeed,  $\text{VaR}_\beta(X)$  represents the most optimistic value that  $X$  can achieve in the worst  $(1 - \beta) \times 100\%$  of possible events. Alternatively,  $\text{CVaR}_\beta(X)$  represents the expected value of  $X$  in the worst  $(1 - \beta) \times 100\%$  of possible events. Therefore, in the near future we wish to consider a problem of OUU similar to problem 7 but using the CVaR instead of the expectation as a risk measure.

Consequently, we reformulate problem 7 into problem

**Problem 8** (CVaR-minimal base moment). We define  $Q$  as in (5.4): the temporal average of the squared magnitude of the base moment. The optimal design  $z_\star$  that we seek is such that

$$(z^\star, s^\star) := \operatorname{argmin}\{J(z, s) := \phi(Q(z), s) : (z, s) \in \mathcal{A} \times \mathbb{R}\}.$$

The set of admissible designs  $\mathcal{A}$  is defined as in problem 7. ◇

The optimisation problem 8 fits into the category discussed in § 2. Therefore, the generic algorithm 1 is applicable, albeit not exactly as in § 3 and 4. There are two major differences:

- (I) the turbulent flow is too chaotic for MLMC methods to perform well, so we consider only single-level MC methods;
- (II) adjoint-based sensitivities are not available to us, which lead us to consider finite-difference methods to estimate the gradient of the objective function.

Item (I) does not prevent us from applying algorithm 2, since a MC estimator is a particular MLMC estimator. However, it prevents us from leveraging simpler models to reduce the cost of statistical estimations. This may yet be achieved with a multi-fidelity Monte Carlo (MFMC) approach, which may perform better than a MLMC one in this case. We refer the reader to deliverable 5.4, § 5.1 for more details on this option.

Item (II) is the only change to algorithm 2 required for problem 8. The major drawback of finite differences compared to the adjoint approach is the complexity with respect to  $\dim Z$ . As discussed previously in § 2.1 p. 10, the partial objective function  $J_1 := z \mapsto J(z, s)$  (for a given  $s \in \mathbb{R}$ ) has to be evaluated at every point of a stencil which grows at least linearly with  $\dim Z$ . Consequently, only shapes with few parameters can realistically be optimised. Here are two considerations to

mitigate this. First, let us consider the simple example of a first-order, forward difference scheme:

$$\begin{aligned} \mathbb{E}(\text{D } J_1(\mathbf{z})(\mathbf{e}_i)) &\approx \mathbb{E}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z})) = \Delta_{h,\mathbf{e}_i} \mathbb{E}(J_1(\mathbf{z})), \\ \text{with } \Delta_{h,\mathbf{e}_i} J_1(\mathbf{z}) &:= \frac{J_1(\mathbf{z} + h\mathbf{e}_i) - J_1(\mathbf{z})}{h}. \end{aligned}$$

Its variance reads

$$\begin{aligned} \text{Var}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z})) &= \frac{1}{h^2} (\text{Var}(J_1(\mathbf{z} + h\mathbf{e}_i)) + \text{Var}(J_1(\mathbf{z})) \\ &\quad - 2\text{Cov}(J_1(\mathbf{z} + h\mathbf{e}_i), J_1(\mathbf{z}))). \end{aligned} \quad (5.5)$$

We see from (5.5) that a high, positive correlation between the values of  $J_1$  across the stencil may yield a finite-difference coefficient with much smaller variance than the individual values of  $J_1$ . It is therefore beneficial to compute samples of  $J_1$  across the stencil jointly rather independently, i.e. for the same random events:

$$\mathbb{E}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z})) \approx \mu_{m'}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z})) \neq \frac{\mu_{m'}(J_1(\mathbf{z} + h\mathbf{e}_i)) - \mu_{m'}(J_1(\mathbf{z}))}{h}. \quad (5.6)$$

This dependent sampling is even necessary, since setting  $\text{Cov}(J_1(\mathbf{z} + h\mathbf{e}_i), J_1(\mathbf{z}))$  to 0 in (5.5) entails  $\lim_{h \rightarrow 0} \text{Var}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z})) = +\infty$ . Even then, a good choice of value for  $h$  is not trivial: reducing it may increase  $\text{Var}(\Delta_{h,\mathbf{e}_i} J_1(\mathbf{z}))$ .

Equation (5.6) leads us to the second consideration. The estimation of this gradient occurs at line 6 of algorithm 2, at which point  $m$  samples of  $J_1(\mathbf{z}_k)$  are known, from lines 4–5. We propose to choose as many as possible of the  $m'$  random events used in (5.6) from these  $m$  events. Thus, only  $m'$  samples of  $J_1(\mathbf{z}_k + h\mathbf{e}_i)$  and  $(m' - m)^+$  of  $J_1(\mathbf{z}_k)$  need be computed along each direction  $\mathbf{e}_i$ .

## 5.4 Variation: parameterised facade

We present here a final application of interest to the project, proposed by our partner `structure`. It is a variation of the previous problem of OUU, with a different building shape and parameterisation. The goal is to optimise a facade design without interference on the structure itself (i.e. sectional geometry fixed). This application is motivated by its industrial interest: such designs are already being investigated and even built, as illustrated by the example on figure 18.

Ideally, the whole volume would be parameterised with a rich design space (e.g. as the airfoil in § 4), to achieve the kind of design represented in figure 19. However, since adjoint-based sensitivities are not currently available for this CFD problem, we use a finite-difference method to estimate the shape sensitivities; this



Figure 18: Aqua tower in Chicago, Illinois. Credits to Steve Hall © Hedrich Blessing

method is notoriously expensive in high dimension. Therefore, we consider a single bulge on the facade, as represented on figure 20. This spherical bulge is defined by its radius and the cylindrical coordinates of its centre, thus the design is controlled by four parameters:

- the radius of the bulge;
- the radial distance of the bulge centre to the axis of the building<sup>6</sup>;
- the azimuth angle of the bulge centre around the axis of the building<sup>6</sup>;
- the height of the bulge centre.

Consequently, the design space is  $Z := \mathbb{R}^4$  and the set of admissible designs is

$$\mathcal{A} := \left[ \sqrt{b_1^2 + b_2^2}, b_3 \right] \times [0, b_3] \times [0, 2\pi] \times \left[ 0, \sqrt{b_1^2 + b_2^2} \right]; \quad (5.7)$$

where  $b_1$ ,  $b_2$ , and  $b_3$  are respectively the length, width, and height of the building. The box constraints represented by  $\mathcal{A} \subsetneq Z$  amount to preventing the bulge from being too large and keeping its centre within the cylinder circumscribing the building.

**Problem 9** (Facade for CVAR-minimal base moment). Let us denote by  $\tilde{B}(z)$  a building consisting of a bulge parameterised by  $z \in Z$  over a fixed block of dimensions  $\mathbf{b} \in \mathbb{R}^3$ , as described above. We wish to find the optimal design  $z_*$  defined as

$$(z_*, s_*) := \operatorname{argmin} \{ \mathbb{E}(\phi(Q(z), s)) : (z, s) \in \mathcal{A} \times \mathbb{R} \}.$$

---

<sup>6</sup>In the horizontal plane.



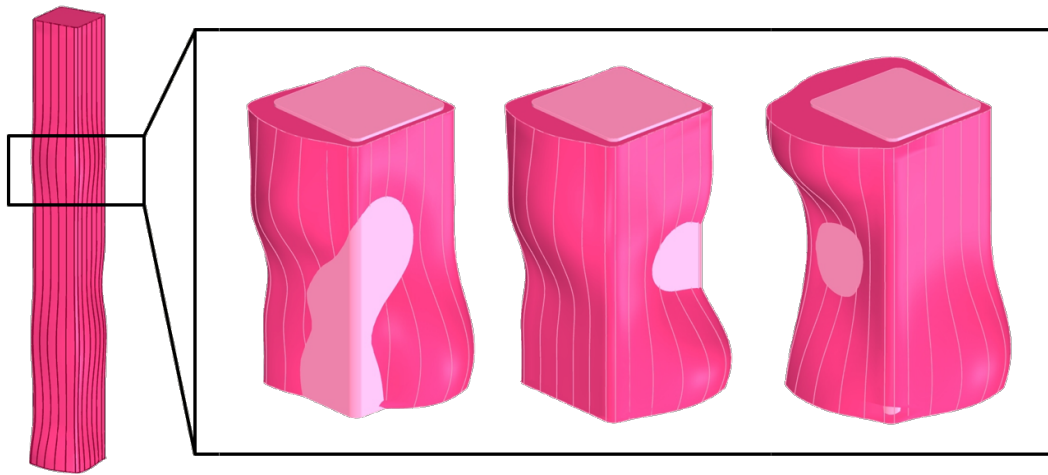


Figure 19: Conceptual goal of facade design

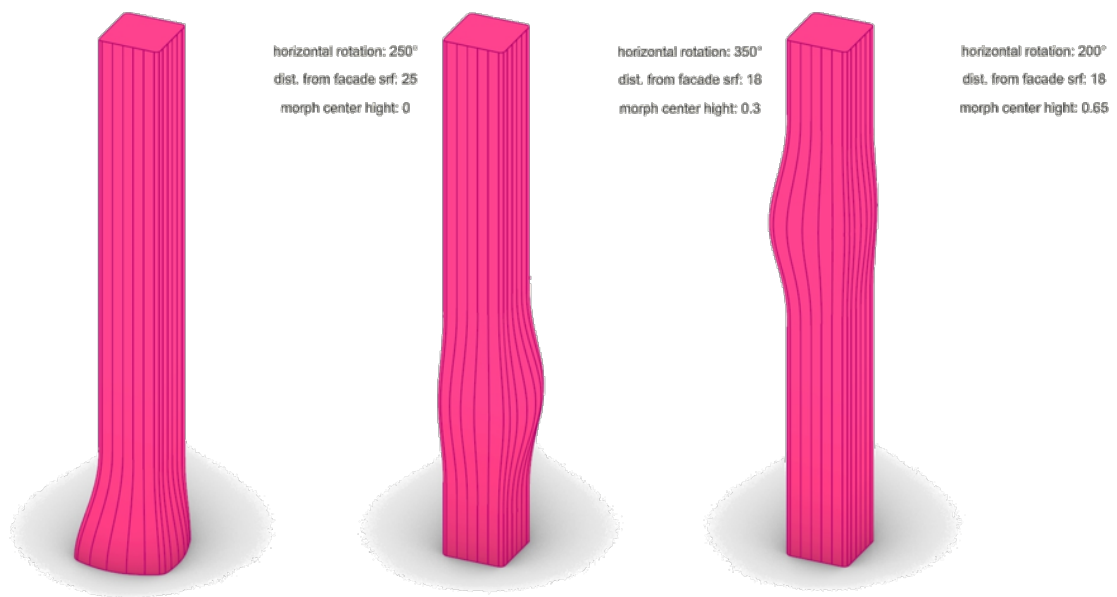


Figure 20: Parametric facade design with few parameters

The set of admissible designs is given in (5.7); the function  $\phi$  is expressed in (2.3); and the quantity of interest is defined in (5.4) from the base moment.  $\diamond$

As expressed above, the industrial interest is the main motivation for this problem. From the point of view of the properties of the optimisation problem, it is not different from problem 8 – the reformulation of problem 7 with the CVAR as measure of risk. Therefore, the methodology discussed in § 5.3 apply here as well. The larger dimension of the design space will only stress the challenge of computing efficiently the descent direction.

## 6 Conclusion

This report brought together work on optimisation under uncertainties (OUU) for the efficient and accurate optimisation of the conditional value at risk (CVaR) on one hand, and developments on computational fluid dynamics (CFD) and deterministic optimisation for the numerical simulation of engineering problems on the other hand. The applicability of the method proposed was explained, along with possible solutions to the issues identified. This led to the explanation of the next steps to take until the end of the projects.

The practical optimisation method proposed in this report is of the gradient-descent type and relies either on adjoint solutions or finite differences for the computation of shape sensitivities. Statistical estimations are performed with a multi-level Monte Carlo (MLMC) method whose accuracy is controlled via reliable error estimators, which allow one to adapt the sample sizes to the required precision. The algorithm proposed is meant to be cost-efficient, and well-suited to parallel computing. The particularisation of this algorithm to a specific unsteady problem was illustrated on the Van der Pol oscillator. This trial problem is comparatively easy to implement and solve, and therefore will be used to assess the implementation and performance of the algorithm.

Two target applications were described in detail, and the application of the proposed optimisation method was presented. First, the shape optimisation of an airfoil is a problem that has been extensively studied, which evinces a suitable mesh convergence and for which the adjoint-based sensitivities can be computed. Results of deterministic optimisation are already available and the application of the proposed method is straightforward, although its implementation is more involved (fluid solver, remesher, etc.); its higher cost will test the efficiency of the optimisation method. It is already used as a benchmark within the project, and will serve as such for the tools of OUU, once implemented. Secondly, the shape optimisation of a building in turbulent wind is the most emblematic and most aspired application of the ExaQUTE project. It is also the most challenging problem discussed: the chaotic behaviour of turbulent fluids precludes the use of a MLMC method, and adjoint solutions are not available to us; work on its simulation is still underway. An adaption of the optimisation method was proposed, with a single-level Monte Carlo (MC) method and finite differences instead of adjoint solutions. Other leads will be explored for the acceleration of statistical estimation – e.g. multi-fidelity approaches. This problem was presented in two variants of shape: the twisted and tapered building, already well studied; the parameterised facade, novel to the project and proposed for its industrial interest.

The next step will be to implement the proposed optimisation method. Most of the tools of uncertainty quantification required are already available in the XMC (Ayoul-Guilmard, Ganesh, Nobile, Rossi et al. 2020) library. However, a few

improvements and new features will be required: ability to resume an interrupted algorithm, better parallelisation, interface extended for optimisation, etc.. Many recent and ongoing improvements on efficient parallelisation are to be leveraged in the optimisation tool; e.g. support for Message Passing Interface (MPI) in the solver and remesher. This tool, once tested locally on the oscillator problem, will be benchmarked on the airfoil problem in a distributed environment – more challenging on the implementation and efficiency. Finally, this will be applied to – and adjusted for – the building problem, benefiting from the future improvements on its simulation. This is the eventual goal for the end of the project.

There are a few noteworthy topics of interest that are beneficial but not necessary to ExaQUTE. These deserve to be mentioned as they may be pursued concurrently or subsequently to the project. First the two ideas mentioned in § 3.3: to use varying time windows for stochastic optimisation of a temporal average, and to explore alternative interpretations of the CVaR for random processes. Then there is the research of a multi-fidelity approach well-suited to turbulent fluid dynamics. Finally, adaptive mesh refinement such as developed in the ExaQUTE project could be leveraged much more efficiently in a MLMC method with a more sophisticated strategy.

## References

- Anderson, W. Kyle and V. Venkatakrisnan (1999). ‘Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation’. In: *Computers and Fluids* 28.4-5, pp. 443–480. ISSN: 00457930. DOI: [10.1016/S0045-7930\(98\)00041-3](https://doi.org/10.1016/S0045-7930(98)00041-3).
- Apostolatos, Andreas and Brendan Keith (14th Apr. 2020). *Deterministic optimization software*. Deliverable 6.1. Version 2.0. ExaQUTE consortium. 17 pp.
- 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 Nuñez, Riccardo Rossi and Riccardo Tosi (13th Nov. 2020). *Report on MLMC for time-dependent problems*. Deliverable 5.4. Version 1.0.0. ExaQUTE consortium.
- Ayoul-Guilmard, Quentin, Sundar Ganesh, Fabio Nobile, Riccardo Rossi, Riccardo Tosi, Rosa M. Badia and Ramon Amela (10th Nov. 2020). *XMC. Python library for hierarchical Monte Carlo algorithms*. Version 2.0.0. DOI: [10.5281/zenodo.4265429](https://doi.org/10.5281/zenodo.4265429).
- Cotela-Dalmau, Jordi, Riccardo Rossi and Eugenio Oñate (Nov. 2016). ‘A FIC-based stabilized finite element formulation for turbulent flows’. In: *Computer Methods in Applied Mechanics and Engineering* 315. DOI: [10.1016/j.cma.2016.11.020](https://doi.org/10.1016/j.cma.2016.11.020).
- Davari, M., R. Rossi, P. Dadvand, I. López and R. Wüchner (15th 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: 1432-0924. DOI: [10.1007/s00466-018-1624-3](https://doi.org/10.1007/s00466-018-1624-3).
- 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\(1392\)](https://doi.org/10.1061/(ASCE)0733-9399(1990)116:6(1392)).
- Eller, David (Sept. 2012). ‘Fast, Unstructured-Mesh Finite-Element Method for Nonlinear Subsonic Flow’. In: *Journal of Aircraft* 49.5, pp. 1471–1479. ISSN: 0021-8669. DOI: [10.2514/1.c031738](https://doi.org/10.2514/1.c031738).
- Ferrándiz, Vicente Mataix, Philipp Bucher, Riccardo Rossi, Rubén Zorrilla, Jordi Cotela, Josep Maria, Miguel Angel Celigueta and Guillermo Casas (Nov. 2020). *Kratos Multiphysics*. Version 8.1. DOI: [10.5281/zenodo.4289897](https://doi.org/10.5281/zenodo.4289897).
- Ferrándiz, Vicente Mataix, Riccardo Tosi et al. (Nov. 2020). *Kratos Examples*. Version 8.1. DOI: [10.5281/zenodo.4293800](https://doi.org/10.5281/zenodo.4293800).
- 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.

- Giles, Michael B. and Niles A. Pierce (2000). ‘An introduction to the adjoint approach to design’. In: *Flow, Turbulence and Combustion*. ISSN: 13866184. DOI: [10.1023/A:1011430410075](https://doi.org/10.1023/A:1011430410075).
- Hojjat, Majid, Electra Stavropoulou and Kai Uwe Bletzinger (Jan. 2014). ‘The Vertex Morphing method for node-based shape optimization’. In: *Computer Methods in Applied Mechanics and Engineering* 268, pp. 494–513. ISSN: 00457825. DOI: [10.1016/j.cma.2013.10.015](https://doi.org/10.1016/j.cma.2013.10.015).
- Joint Committee on Structural Safety (2001). *Probabilistic model code*. Standard.
- Kaimal, Jagadish Chandran and John J. Finnigan (1994). *Atmospheric boundary layer flows: their structure and measurement*. Oxford university press.
- Kareem, Ahsan and A. Tamura (2015). *Advanced structural wind engineering*. Springer.
- Kouri, Drew P. and Alexander Shapiro (2018). ‘Optimization of PDEs with Uncertain Inputs’. In: *Frontiers in PDE-Constrained Optimization*. Springer, pp. 41–81.
- 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](https://doi.org/10.1137/17M1135566).
- Lea, Daniel J., Myles R. Allen and Thomas W. N. Haine (2000). ‘Sensitivity analysis of the climate of a chaotic system’. In: *Tellus A: Dynamic Meteorology and Oceanography* 52.5, pp. 523–532.
- 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](https://doi.org/10.2514/6.1995-1806).
- Rockafellar, Tyrrell R. and Stanislav Uryasev (1st Apr. 2000). ‘Optimization of conditional value-at-risk’. In: *Journal of risk* 2.3, pp. 21–41. DOI: [10.21314/JOR.2000.038](https://doi.org/10.21314/JOR.2000.038).
- Structural Eurocode, Technical Committee CEN/TC250 (2005). *Actions on structures. General actions — Wind actions*. Eurocode 1:1-4. Brussels: European Committee for Standardization, EN 1991-1-4.
- Uryasev, Stanislav, Sergey Sarykalin, Gaia Serraino and Konstantin Kalinchenko (2010). *VaR vs CVaR in Risk Management and Optimization*. CARISMA conference.
- Van der Pol, Balthasar (1920). ‘A theory of the amplitude of free and forced triode oscillations’. In: *Radio Review* 1, pp. 701–710.
- Zorrilla, R., A. Laresse and R. Rossi (2019). ‘A modified Finite Element formulation for the imposition of the slip boundary condition over embedded volumeless geometries’. In: *Computer Methods in Applied Mechanics and Engineering* 353, pp. 123–157. ISSN: 0045-7825. DOI: [10.1016/j.cma.2019.05.007](https://doi.org/10.1016/j.cma.2019.05.007).

## A Adjoint of first-order ODE with additive noise

We present here the derivation of the adjoints for a first-order ordinary differential equation (ODE) with white noise forcing for an objective function containing the CVaR of a time-averaged quantity of the trajectory. Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a complete probability space. Let  $u(t) \in \mathbb{R}^{N_u}$  be the state vector at time  $t \in [0, T]$  governed by the following ODE with additive noise.

$$\begin{aligned} \dot{u} &= g(u, z) + \tau \dot{W} \quad \text{over } ]0, T], \\ u(0) &= u^0 \end{aligned}$$

where  $z \in Z \subset \mathbb{R}^{N_z}$  are the input parameters,  $g : \mathbb{R}^{N_u} \times Z \rightarrow \mathbb{R}^{N_u}$ , and  $W : [0, T] \times \Omega \rightarrow \mathbb{R}^{N_u}$  is a  $N_u$ -dimensional standard Wiener process.

We discretise the problem on a uniform temporal grid  $\mathbb{T}$  where the interval  $[0, T]$  is divided into  $N \in \mathbb{N}$  segments of step size  $\Delta t = T/N$ .  $\mathbb{T} := \{t_n := n\Delta t : n \in \llbracket 0, N_t \rrbracket\}$ . The ODE is discretised using the Euler–Maruyama scheme, which reads as follows.

$$\begin{aligned} u^{n+1} &= u^n + \Delta t g(u^n, z) + \tau \sqrt{\Delta t} \xi^n \\ u^0 &= u_0 \end{aligned}$$

where  $\xi^n \in \mathbb{R}^{N_u}$  are  $N_u$ -dimensional random vectors whose components are independent and identically-distributed (i.i.d.) standard normal variables. We are interested in computing the statistics of time-averages of functions of the trajectory.

$$Q = \langle f(u) \rangle_T$$

We approximate the time integral using the trapezoid rule on the aforementioned temporal grid, leading to

$$Q \approx \langle f(u) \rangle_{\mathbb{T}} := \sum_{n=0}^{N-1} \left( \frac{f(u^n) + f(u^{n+1})}{2} \right) \frac{\Delta t}{T}$$

We are interested in minimising the CVaR of this quantity over the parameters  $z$  but use the combined formulation as follows.

$$(z_\star, s_\star) := \operatorname{argmin} \left\{ s + \frac{\mathbb{E}((Q - s)^+)}{1 - \beta} : s \in \mathbb{R}, z \in Z \right\}.$$

The Lagrangian for the problem reads

$$\begin{aligned} \mathcal{L} &= s + \frac{\mathbb{E}((Q - s)^+)}{1 - \beta} \\ &\quad + \mathbb{E} \left( \sum_{n=0}^{N-1} \lambda^{n+1} (u^n + \Delta t g^n + \tau \sqrt{\Delta t} \xi^n - u^{n+1}) - \lambda^0 (u^0 - u_0) \right), \end{aligned}$$

where we use  $g^n := g(u^n, z)$ , and  $\lambda^n \in \mathbb{R}^{N_u}$ ,  $n \in \llbracket 0, N \rrbracket$  denote the Lagrange multipliers for the initial condition and the steps of the discretised equations.

Differentiating with respect to  $z$  gives

$$\begin{aligned} \frac{d\mathcal{L}}{dz} &= \mathbb{E} \left( \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \sum_{n=0}^N \left( \frac{f_u^n u_z^n + f_u^{n+1} u_z^{n+1}}{2} \right) \Delta t \right) \\ &\quad + \mathbb{E} \left( \sum_{n=0}^{N-1} \lambda^{n+1} (u_z^n + \Delta t (g_u^n u_z^n + g_z^n) - u_z^{n+1}) \right) =: \mathbb{E}(\hat{\mathcal{L}}) \end{aligned}$$

Re-arranging the terms leads to

$$\begin{aligned} \hat{\mathcal{L}} &= u_z^N \left( \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \frac{f_u^N \Delta t}{2} - \lambda^N \right) + \Delta t \lambda^{N+1} g_z^N \\ &\quad + \sum_{n=1}^{N-1} u_z^n \left( \lambda^{n+1} (1 + \Delta t g_u^n) - \lambda^n + \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \Delta t f_u^n \right) \\ &\quad + u_z^0 \left( \lambda^1 (1 + \Delta t g_u^0) + \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \frac{f_u^0 \Delta t}{2} \right) + \Delta t \lambda^1 g_z^0, \end{aligned}$$

where we have used the subscript notation for partial derivatives.

We have in our case that  $u_z^0 = 0$ . To remove terms dependent on  $u_z^n$ , we set

$$\begin{aligned} \forall n \in \llbracket 1, N-1 \rrbracket, \quad \lambda^n &= \lambda^{n+1} (1 + \Delta t g_u^n) + \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \Delta t f_u^n, \\ \text{and } \lambda^N &= \frac{\mathbf{1}(Q \geq s)}{(1-\beta)T} \frac{f_u^N \Delta t}{2}. \end{aligned}$$

This gives us the adjoint equations which are solved backwards in time. Once the adjoints are solved, the sensitivity can be computed as

$$\frac{d\mathcal{L}}{dz} = \mathbb{E} \left( \sum_{n=0}^{N-1} \Delta t \lambda^{n+1} g_z^n \right).$$

That is, setting

$$J(z, s) = s + \frac{\mathbb{E}((Q(z) - s)^+)}{1-\beta},$$

we have that  $\bar{\nabla}_z^0 J = \mathbb{E} \left( \sum_{n=0}^{N-1} \Delta t \lambda^{n+1} g_z^n \right)$ .



## List of Algorithms

1	Ideal gradient-descent with exact VaR . . . . .	11
2	Practical gradient-descent for CVaR optimisation with MLMC . . . . .	14

## List of Figures

1	Trajectory of an unforced Van der Pol oscillator . . . . .	16
2	Scheme of the domain in the potential flow formulation . . . . .	19
3	Scheme of an element split by the wake . . . . .	23
4	Embedded wake representation in body fitted meshes . . . . .	25
5	Contour fills of the primal solution for a NACA0012 airfoil . . . . .	26
6	Example mesh and geometry parameters for a NACA0012 airfoil . . . . .	27
7	Scheme of the main airfoil geometrical characteristics . . . . .	30
8	Example of the adjoint field solution . . . . .	32
9	Example of the shape update in two steps of the optimisation problem . . . . .	32
10	Evolution of objective and relative tolerance . . . . .	34
11	Comparison of the initial and final results in the optimisation problem . . . . .	35
12	Absolute World towers . . . . .	37
13	Details of the simulation domain and boundary conditions . . . . .	40
14	Details of adopted meshes . . . . .	41
15	Details of tower geometry and design parameters . . . . .	42
16	Geometry modifications from optimisation . . . . .	45
17	Optimisation results . . . . .	45
18	Aqua tower . . . . .	48
19	Conceptual goal of facade design . . . . .	49
20	Parametric facade design with few parameters . . . . .	49

## Acronyms

<b>ABL</b>	atmospheric boundary layer
<b>CFD</b>	computational fluid dynamics
<b>CDF</b>	cumulative distribution function
<b>CVaR</b>	conditional value at risk
<b>FEM</b>	finite-element method
<b>HPC</b>	high-performance computing
<b>MC</b>	Monte Carlo
<b>MFMC</b>	multi-fidelity Monte Carlo
<b>MLMC</b>	multi-level Monte Carlo
<b>MPI</b>	Message Passing Interface
<b>MSE</b>	mean squared error
<b>OUU</b>	optimisation under uncertainties
<b>PDE</b>	partial differential equation
<b>ODE</b>	ordinary differential equation
<b>PDF</b>	probability density function
<b>QOI</b>	quantity of interest
<b>VaR</b>	value at risk
<b>VMS</b>	variational multi-scale

## 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

**NB** nota bene