Exascale Quantification of Uncertainties for Technology and Science Simulation

D6.3 Report on stochastic optimisation for simple 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 6.3
Status:	Final

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

Authoring

Prepared by EPFL				
Authors	Partner	Modified	Version	Comments
Quentin Ayoul-Guilmard	EPFL	All	1.0	Redaction
Sundar Ganesh				Review
Fabio Nobile				Expertise and review

Change log

Versions	Modified Page/Sections	Comments
1.0	All	Submitted version

Approval

Approved by EPFL				
	Name	Partner	Date	OK
Task leader	Fabio Nobile	EPFL	2020-07-10	OK
Coordinator	Riccardo Rossi	CIMNE	2020-07-27	OK

Executive summary

This report addresses the general matter of optimisation under uncertainties, following a previous report on stochastic sensitivities (deliverable 6.2). It describes several theoretical methods, as well their application into implementable algorithms. The specific case of the conditional value at risk chosen as risk measure, with its challenges, is prominently discussed. In particular, the issue of smoothness – or lack thereof – is addressed through several possible approaches. The whole report is written in the context of high-performance computing, with concern for parallelisation and cost-efficiency.

Contents

1 Introduction

The ExaQUte project researches and develops methods and tools to optimise the shape of tall buildings to withstand wind. This encompasses many scientific fields, among which is stochastic optimisation constrained by partial differential equations (PDE), since the wind is modelled as an uncertain loading in a problem of fluid dynamics. Also immanent to every aspect of the methodology is the need for high-performance computing (HPC), due to the complexity of the simulations involved.

The formulation and resolution of this problem of optimisation under uncertainties (OUU) of the shape of a building requires several modelling and methodology choices. Foremost among them is that of a deterministic quantification of the risk entailed by a given shape. The conditional value at risk (CVaR) has been selected for its mathematical properties and pertinence to risk-averse engineering design. In order to optimise complex shapes, a rich, high-dimensional design space has to be considered, which bears heavily on the computation of the sensitivity of the solution of the PDE with respect to the design. The chosen methodology is to use adjoint-based sensitivities, which leads us to consider gradient-descent algorithms to solve the optimisation problem. Finally, multi-level Monte Carlo (MLMC) estimators are meant to leverage parallel computations and sophisticated remeshing tools in order to accelerate statistical estimations; the ExaQUte project develops a dedicated library (XMC by Ayoul-Guilmard et al. 2019) for this purpose.

This report follows and continues deliverable 6.2 by Ganesh et al., which studied a general OUU problem. It derived optimality conditions and expressed the stochastic sensitivities involved, with examples for some classical risk measures. Nevertheless, it did not discuss the actual computation of an optimal solution. The current report proposes methods to perform this optimisation, and is organised as follows.

Section 2 presents the general case of smooth optimisation. We begin the definition of the OUU problem, then describe a generic gradient-descent algorithm for any smooth risk measure. Then we discuss possible ways to discretise the probability space, in order to implement the algorithm in practice; in particular, we introduce MLMC estimators. Finally, practical, implementable adaptations of the generic gradient-descent algorithm are proposed. Section 3 deals with the case where the risk measure is CVaR, and the specific considerations thereof – chiefly the non-smoothness of this risk measure. First, the definition and relevant properties of the CVaR are given and the OUU problem is reformulated accordingly. Then, two alternative approaches are proposed to solve it. For each approach, we propose a corresponding algorithm and discuss its practical implementation, including the issue of non-smoothness. In section 4, we discuss the possible regularisation of the optimisation problem in order to address the lack of smoothness of the CVaR.

Two different regularisation approaches are described, along with the consequent alterations of the previous algorithms. Finally, we conclude with a summary of the essential points, and an outlook on future developments, either required or advisable.

2 Optimisation of a smooth risk measure

2.1 Optimisation problem

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Let Z denote the Hilbert space of design variables and $\mathcal{A} \subset Z$ the subset of feasible designs. For any Banach space B, we will denote by B^* its dual space and by $\langle \cdot, \cdot \rangle_{B,B^*}$ the duality pairing.

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. We will write this constraint in abstract form as

$$F(u(z)(\omega), z, \omega) = 0, \text{ for } \mathbb{P}\text{-a.e. } \omega \in \Omega, \tag{2.1}$$

where $u(z)(\omega) \in U$ is the state of the system and $F : U \times Z \times \Omega \to Y^*$ represents the residual of the PDE, with U and Y suitable Banach spaces. We assume that, for any $z \in \mathcal{A}$, there exists a unique $u(z) \in \mathrm{L}^p(\Omega, \mathcal{F}, \mathbb{P}; U)$ satisfying (2.1), with $p \in [1, +\infty]$. Thus, the map $u : Z \to \mathrm{L}^p(\Omega, \mathcal{F}, \mathbb{P}; U)$ is well-defined over $\mathcal{A}$. Here, $\mathrm{L}^p(\Omega, \mathcal{F}, \mathbb{P}; U)$ denotes the Bochner space of random variables from $(\Omega, \mathcal{F}, \mathbb{P})$ to U with at least p finite moments. From now on, we will simplify this notation to $\mathrm{L}^p(\Omega, U)$. We also assume that $\exists n' \in [1, +\infty[$, s.t. $\forall z \in Z$, $\forall y \in \mathrm{L}^p(\Omega, U)$, $F(y(\cdot), z, \cdot) \in \mathrm{L}^{n'}(\Omega, Y^*)$; i.e. the residual of the PDE has n' finite moments, whenever evaluated on a candidate solution that has p finite moments.

The quantity of interest (QoI) with respect to which an optimal design is sought is a function $Q : \mathrm{L}^p(\Omega, U) \to \mathrm{L}^r(\Omega, \mathbb{R})$ of the state, for some $r \in [1, +\infty]$. In particular, to simplify the exposition, we consider only a QoI Q that does *not* depend explicitly on the random effects, nor on the design. We assume that there exists $\bar{Q} : U \to \mathbb{R}$ such that

$$\forall v \in \mathrm{L}^p(\Omega, U), \quad Q(v)(\omega) = \bar{Q}(v(\omega)), \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega.$$

Let us consider problem 1.

Problem 1 (Generic OUU)**.** Find the optimal design $z^\star \in \mathcal{A}$ solution of

$$\begin{cases} \min\{J(z) := \mathcal{R}(Q(u(z))) + P(z) : z \in \mathcal{A}\} \\ \text{s.t. } F(u(z)(\omega), z, \omega) = 0, \text{ for } \mathbb{P}\text{-a.e. } \omega \in \Omega. \end{cases} \tag{2.2}$$

The objective function J in (2.2) features a risk measure $\mathcal{R} : \mathrm{L}^r(\Omega, \mathbb{R}) \to \overline{\mathbb{R}}$, where $\overline{\mathbb{R}} := [-\infty, \infty]$, and a penalisation term $P : Z \to \mathbb{R}$ to oppose expensive or otherwise undesirable designs.

Definition 1 (Proper risk measure)**.** A risk measure $\mathcal{R} : \mathrm{L}^1(\Omega, \mathbb{R}) \to \overline{\mathbb{R}}$ is said to be 'proper' if and only if (iff.) the two following properties hold

$$\forall X \in \mathrm{L}^1(\Omega, \mathbb{R}),\ \mathcal{R}(X) > -\infty;$$
$$\operatorname{dom}(\mathcal{R}) := \{X \in \mathrm{L}^1(\Omega, \mathbb{R}) : \mathcal{R}(X) < +\infty\} \neq \emptyset.$$

Definition 2 (Coherent risk measure)**.** A risk measure $\mathcal{R} : \mathrm{L}^1(\Omega, \mathbb{R}) \to \overline{\mathbb{R}}$ is said to be 'coherent' if it satisfies the following properties for any $X, Y \in \mathrm{L}^1(\Omega, \mathbb{R})$.

Monotonicity:

$X \leqslant Y \text{a.s.} \implies \mathcal{R}(X) \leqslant \mathcal{R}(Y)$.

Equivariance by translation:

$\forall a \in \mathbb{R},\ \mathcal{R}(X + a) = \mathcal{R}(X) + a$.

Convexity[1]*:*

$\forall b \in \,]0, 1[,\ \mathcal{R}(bX + (1 - b)Y) \leqslant b\mathcal{R}(X) + (1 - b)\mathcal{R}(Y)$.

Positive homogeneity:

$\forall c \in [0, +\infty[,\ \mathcal{R}(cX) = c\mathcal{R}(X)$.

As an example of the above definition, it is immediate to see that the expectation is a coherent risk measure, whereas the variance is not (it is not monotonous). In this report, we only consider the case of a *proper* and *coherent* risk measure $\mathcal{R}$.

Let us now introduce a few notations on differentiation. We denote by $\mathcal{L}(V, W)$ the space of linear operators from a vector space V to a vector space W. Let $\boldsymbol{V} := V_1 \times V_2 \times \ldots \times V_n$ be a product of $n \in \mathbb{N}$ vector spaces, $f : \boldsymbol{V} \to W$ a differentiable multivariate function and $v \in \boldsymbol{V}$. The partial differential of f in v with respect to the k-th variable is denoted by $\mathrm{D}_k f(v)$, and $\mathrm{D}_k f(v) \in \mathcal{L}(V_k, W)$. If V_k is a Hilbert space and $W = \mathbb{R}$, then the Riesz representation theorem states that

$$\exists!\, \nabla_k f(v) \in V_k,\ \forall x \in V_k, \quad \mathrm{D}_k f(v)(x) = \langle \nabla_k f(v), x \rangle_{V_k},$$

where $\langle \cdot, \cdot \rangle_{V_k}$ denotes the inner product of V_k. This dual representation of $\mathrm{D}_k f(v) \in V_k^*$ is called the 'gradient' of f at v in V_k, or in the k-th direction. The subscript k will be omitted for univariate functions.

In this section, we assume that $\mathcal{R}$ and P are continuously differentiable. Since $\mathcal{R} : \mathrm{L}^r(\Omega, \mathbb{R}) \to \overline{\mathbb{R}}$, we have that $\forall X \in \mathrm{L}^r(\Omega, \mathbb{R})$, $\mathrm{D}\,\mathcal{R}(X) \in \mathcal{L}(\mathrm{L}^r(\Omega, \mathbb{R}), \mathbb{R})$. Then,

[1] Sometimes replaced by the weaker property of sub-additivity. Both are equivalent under assumption of positive homogeneity.

a consequence of theorem 6.10, p. 267 of Shapiro et al. 2009 is that there is a unique element $\nabla \mathscr{R}(X)$ of the dual space[2] $\mathrm{L}^{r'}(\Omega, \mathbb{R})$ such that

$$\forall Y \in \mathrm{L}^r(\Omega, \mathbb{R}), \quad \mathrm{D}\, \mathscr{R}(X)(Y) = \mathbb{E}(Y \nabla \mathscr{R}(X)). \tag{2.3}$$

Optimality conditions for an OUU problem such as problem 1 were inferred in deliverable 6.2. A design $z^\star \in \mathcal{A}$ optimal in the sense of (2.2) satisfies

$$\langle \mathrm{D}\, P(z^\star) - \mathbb{E}(\mathrm{D}_2\, F^*(u(z^\star)(\cdot), z^\star, \cdot)(\lambda(z^\star)(\cdot))), z^\star - z \rangle_{Z^*, Z} \geqslant 0, \quad \forall z \in \mathcal{A}. \tag{2.4}$$

where, for $\mathbb{P}$-a.e. $\omega \in \Omega$, $u(z^\star) \in \mathrm{L}^p(\Omega, U)$ satisfies

$$F(u(z^\star)(\omega), z^\star, \omega) = 0 \tag{2.5}$$

and $\lambda(z^\star) \in \mathrm{L}^n(\Omega, Y)$ verifies

$$\mathrm{D}_1\, F^*(u(z^\star)(\omega), z^\star, \omega)(\lambda(z^\star)(\omega)) = \nabla \mathscr{R}(Q(u(z^\star)))(\omega)\, \mathrm{D}\, \bar{Q}(u(z^\star)(\omega)). \tag{2.6}$$

In (2.6), we denote $\mathrm{D}_1\, F^*(u(z)(\omega), z, \omega) \in \mathcal{L}(Y, U^*)$ the adjoint of $\mathrm{D}_1\, F(u(z)(\omega), z, \omega) \in \mathcal{L}(U, Y^*)$, and $n := (1 - n'^{-1})^{-1}$.

Equations (2.4)–(2.6) are the first-order optimality conditions for problem 1. We refer the reader to deliverable 6.2 for a more detailed explanation.

2.2 Gradient-descent algorithm

Algorithm 1 outlines a generic gradient-descent algorithm based on optimality conditions (2.4)–(2.6). By assumption, $J : Z \to \mathbb{R}$ is differentiable and Z is a Hilbert space, so $\forall z \in Z$, $\mathrm{D}\, J(z) \in Z^*$ and there exist a unique $\nabla J(z) \in Z$ defined as in (2.3). In line 5, we denote $\prod_{\mathcal{A}} : Z \to \mathcal{A}$ the projector onto $\mathcal{A}$.

The stopping criterion on line 1 has been set on the stagnation:

$$\frac{\|z_{k+1} - z_k\|}{\|z_k\|} \leqslant \eta,$$

for a chosen relative tolerance $\eta \in]0, +\infty[$. The step size γ_k may be chosen either a priori or adaptively; in the latter case, the adaptation is to ensure reduction of the objective value, e.g. as in a backtracking line-search method (see Armijo 1966). This choice is unspecified in algorithm 1.

The algorithm is called ‘ideal’ because it requires exact evaluation of the risk measure, as well as exact resolution of the primal and adjoint equations (2.5) and (2.6), a.e. in Ω.

[2] $\mathrm{L}^r(\Omega, \mathbb{R})^* = \mathrm{L}^{r'}(\Omega, \mathbb{R})$ with $r' := (1 - r^{-1})^{-1}$.

Algorithm 1: Generic ideal gradient-descent algorithm

```
1 Initialise: $z_0$, $\gamma$, $\eta$
2 While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
3     Find $u$ s.t. $F(u(\omega), z_k, \omega) = 0$ for $\mathbb{P}$-a.e. $\omega \in \Omega$
4     Find $\lambda$ s.t., for $\mathbb{P}$-a.e. $\omega \in \Omega$,
        $\mathrm{D}_1 F^*(u(\omega), z_k, \omega)(\lambda(\omega)) = \nabla \mathcal{R}(Q(u))(\omega)\, \mathrm{D}\, \bar{Q}(u(\omega))$
5     Compute descent direction $\nabla J(z_k) := \nabla P(z_k) - \mathbb{E}(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda))$
6     Set new design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J(z_k))$
7 Result: $z_{k+1}$
```

In this report, we discuss only the case where Q is linear; however, this study can easily be extended to a class of functions with suitable regularity. With this assumption, $\exists q \in U^*$, s.t. $\forall v \in \mathrm{L}^p(\Omega, U)$, $Q(v)(\omega) = \langle q, v(\omega) \rangle_{U,U^*}$ for a.e. $\omega \in \Omega$. This allows the following simplification in line 4 of algorithm 1:

$$\mathrm{D}\, \bar{Q}(u(\omega)) = q. \tag{2.7}$$

Furthermore, we will assume that, $\forall z \in \mathcal{A}$, the cumulative distribution function (CDF) of $\psi_z := Q(u(z))$ is absolutely continuous, in order to be able to define a probability density function (PDF) $f_{\psi_z} \in \mathrm{L}^1(\mathbb{R})$ such that

$$\mathbb{E}(Q(u(z))) = \mathbb{E}(\psi_z) = \int_{\mathbb{R}} x f_{\psi_z}(x)\, \mathrm{d}x.$$

2.3 Discretisation of the OUU problem

In the previous section we have discussed an idealised algorithm in the continuous setting, with an exact resolution of problem 1 a.e. in Ω and exact evaluation of the risk measure. However, practical, implementable algorithms will require discretising the PDE as well as the probability space. We focus first on the latter, which also entails an approximation of the expectation of any random variable on Ω.

Hereafter, we discuss two strategies for approximating the probability space and expectation, namely the Monte Carlo (MC) and multi-level Monte Carlo (MLMC) methods. Then, we will propose practical adaptations of idealised algorithm 1, using either MC or MLMC approximations, combined with two common approaches of stochastic programming: sample-average approximation (SAA) and stochastic approximation (SA).

2.3.1 Monte Carlo and multi-level Monte Carlo quadratures

Estimation of the expectation Let us choose $m \in \mathbb{N}$ and draw randomly and independently $\boldsymbol{\omega} = (\omega_i)_{i=1}^m \in \Omega^m$ from the measure $\mathbb{P}$. From this we define an empirical measure as follows:

$$\forall A \in \mathcal{F}, \quad \mu_m(A) := \frac{1}{m}\sum_{i=1}^m \mathbb{1}(\omega_i \in A),$$

with

$$\mathbb{1}(B) = \begin{cases} 1 \text{ if } B \text{ is true,} \\ 0 \text{ if } B \text{ is false.} \end{cases}$$

Consequently, the expectation of a random variable $X : \Omega \to \mathbb{R}$ is approximated as

$$\mathbb{E}(X) \approx \frac{1}{m}\sum_{i=1}^m X(\omega_i) =: \mu_m(X). \tag{2.8}$$

The estimator μ_m is called a MC estimator, and its mean squared error (MSE) is

$$\text{MSE}(\mu_m(X)) := \mathbb{E}((\mu_m(X) - \mathbb{E}(X))^2) = \frac{\mathbb{V}\text{ar}(X)}{m},$$

where the expectation is taken here with respect to the random sample $\boldsymbol{\omega}$. The rate of convergence $\mathcal{O}(m^{-1})$ of the MSE is notoriously slow, and the cost of a sample is typically high in our applications of interest. Numerous methods have been proposed to accelerate the convergence or improve the complexity of Monte Carlo estimators; we discuss here the MLMC approach.

Let us assume that the random variable X is inaccessible; however, we can draw samples from suitable approximations $(X_l)_{l=0}^L$ of it, where the index l quantifies the accuracy of the approximation: the higher the index, the more accurate the approximation. With $\boldsymbol{m} \in \mathbb{N}^{L+1}$, we make the following approximation:

$$\begin{aligned} \mathbb{E}(X) \approx \mathbb{E}(X_L) &= \mathbb{E}(X_0) + \sum_{l=1}^L \mathbb{E}(X_l - X_{l-1}) \\ &\approx \mu_{m_0}(X_0) + \sum_{l=1}^L \mu_{m_l}(X_l - X_{l-1}) =: \mu_{\boldsymbol{m}}(X_L) \end{aligned}$$

The estimator $\mu_{\boldsymbol{m}}$ is called a MLMC estimator, with $L+1$ levels and the associated sets of events in $\Omega^{\boldsymbol{m}} := \prod_{i=1}^{L+1} \Omega^{m_i}$. In particular, for any two distinct levels $k, l \in \{0 \dots L\}$, the MC estimators μ_{m_k} and μ_{m_l} use distinct and independent sets of events $\boldsymbol{\omega}_k \in \Omega^{m_k}$ and $\boldsymbol{\omega}_l \in \Omega^{m_l}$.

The associated MSE is

$$\operatorname{MSE}(\mu_{\boldsymbol{m}}(X_L)) := \overbrace{\mathbb{E}((\mu_{\boldsymbol{m}}(X_L) - \mathbb{E}(X_L))^2)}^{\text{statistical error}} + \overbrace{(\mathbb{E}(X_L) - \mathbb{E}(X))^2}^{\text{bias error}} \tag{2.9}$$

$$= \frac{\mathbb{V}\mathrm{ar}(X_0)}{m_0} + \sum_{l=1}^{L} \frac{\mathbb{V}\mathrm{ar}(X_l - X_{l-1})}{m_l} + \mathbb{E}(X_L - X)^2 \tag{2.10}$$

Since X per se is not accessible, we had implicitly introduced the additional approximation $\mathbb{E}(X) \approx \mathbb{E}(X_L)$, hence the two terms in the right-hand side of (2.9). One could likewise consider the MC estimator $\mu_m(X_L)$, which would then have the same bias contribution to the MSE as in (2.9). In the context of MLMC methods, one typically assumes the existence of a converging sequence of approximations $(X_l)_{l\in\mathbb{N}}$ such that $\lim_{l\to+\infty} \mathbb{E}(X_l - X) = 0$. The highest approximation level L should then be chosen large enough to keep the bias error below a prescribed tolerance. Likewise, the sample sizes $\boldsymbol{m}$ should be chosen large enough to control the statistical error, as per (2.10).

For differential equations, X is generally a function of the solution of the differential problem posed over an infinite-dimensional function space, and the sequence $(X_l)_{l\in\mathbb{N}}$ corresponds to increasingly-fine discretisations of that function space. Therefore, the bias error is sometimes referred to as 'discretisation error'. Henceforth, for any $l \in \mathbb{N}$ we will denote by $F_l : U_l \times Z \times \Omega \to Y_l^*$ the discretised version of F for suitable finite-dimensional spaces U_l and Y_l.

Although a MC estimator is simpler to implement and preserves convexity, a MLMC estimator achieves a better complexity with a good choice of L and $\boldsymbol{m}$. The celebrated complexity result by Giles 2008 assumes suitable decays of $\mathbb{V}\mathrm{ar}(X_l - X_{l-1})$ and $\mathbb{E}(X_l - X)$ and reasonable increase of the sampling cost, with respect to l. Verifying these assumptions is not always trivial, but will not be discussed in this report. Adaptive methods to tune a MLMC estimator from suitable error estimations have been studied, in particular by Giles 2015 and Collier et al. 2015.

Multi-level estimation of a general risk measure Hereafter we will use exclusively MLMC estimators, of which the standard (single-level) MC estimator is a particular case (setting $L := 0$). Let us assume that, for $m \in \mathbb{N}$, we have an *unbiased* estimator $\mathcal{R}_m$ of $\mathcal{R}$ associated with the empirical probability defined by a set of events $\boldsymbol{\omega} \in \Omega^m$. The precise definition of the unbiased estimator $\mathcal{R}_m$ depends on $\mathcal{R}$; e.g. if $\mathcal{R} = \mathbb{E}$ then μ_m as defined in (2.8) is such an estimator. Let us choose $\boldsymbol{m} \in \mathbb{N}^{L+1}$ then draw randomly and independently $\boldsymbol{\omega} := (\boldsymbol{\omega}_l)_{l=0}^{L} \in \Omega^{\boldsymbol{m}}$ from the measure $\mathbb{P}$. We can write a multi-level estimator of $\mathcal{R}(X)$ as

$$\begin{aligned}\mathcal{R}(X) \approx \mathcal{R}(X_L) &:= \mathcal{R}(X_0) + \sum_{l=1}^{L} \mathcal{R}(X_l) - \mathcal{R}(X_{l-1}) \\ &\approx \mathcal{R}_{m_0}(X_0) + \sum_{l=1}^{L} \mathcal{R}_{m_l}(X_l) - \mathcal{R}_{m_{l-1}}(X_{l-1}) =: \mathcal{R}_{\boldsymbol{m}}(X_L)\end{aligned}$$

Since the estimators are all unbiased, $\mathbb{E}(\mathcal{R}_{\boldsymbol{m}}(X_L)) = \mathcal{R}(X_L)$: $\mathcal{R}_{\boldsymbol{m}}(X_L)$ is an unbiased estimator of $\mathcal{R}(X_L)$ – albeit not of $\mathcal{R}(X)$.

In this section, we will assume that $\mathcal{R}$ remains continuously differentiable after discretisation; i.e. $\mathcal{R}_{m_l}$ (for any $l \in \{0 \dots L\}$), and hence $\mathcal{R}_{\boldsymbol{m}}$, are continuously differentiable. In particular, for any $X \in \mathrm{L}^r(\Omega, \mathbb{R})$, we can define $\nabla \mathcal{R}_{m_l}(X)$ and

$$\nabla \mathcal{R}_{\boldsymbol{m}}(X) := \nabla \mathcal{R}_{m_0}(X_0) + \sum_{l=1}^{L} \nabla \mathcal{R}_{m_l}(X_l) - \nabla \mathcal{R}_{m_{l-1}}(X_{l-1}).$$

Neither this smoothness assumption nor the unbiasedness assumption hold if the risk measure $\mathcal{R}$ is the CVaR. We will come back to this issue in § 3.

2.3.2 Sample-average approximation

In this approach, a discretisation of the probability measure $\mathbb{P}$ is performed a priori, and then the OUU problem is solved for the finite set of MLMC realisations. The sample-average approximation of problem 1 is then problem 2.

Problem 2 (SAA of problem 1)**.** Let $L \in \mathbb{N}$, $\boldsymbol{m} \in \mathbb{N}^{L+1}$ and $\boldsymbol{\omega} \in \Omega^{m}$. Find the optimal design $z^\star \in \mathcal{A}$ solution of

$$\begin{cases} \min\{J_{\boldsymbol{m}}(z) := \mathcal{R}_{\boldsymbol{m}}(Q(u(z))) + P(z) : z \in \mathcal{A}\} \\ \text{s.t. } \forall l \in \{0 \dots L\},\ \forall i \in \{1 \dots m_l\},\ \begin{cases} F_l(u_l(z)(\omega_{l,i}), z, \omega_{l,i}) = 0, \\ F_{l-1}(u_{l-1,i}(z)(\omega_{l,i}), z, \omega_{l,i}) = 0. \end{cases} \end{cases}$$

This approach is straightforward to implement. To illustrate it, algorithm 2 presents the SAA version of algorithm 1. By convention, lines 7 and 12 are skipped if $l = 0$. Note that the whole algorithm, including the MLMC estimator on line 13, uses the same set of events $\boldsymbol{\omega} \in \Omega^{\boldsymbol{m}}$ drawn on line 2. Without further assumption on $\mathcal{R}$, the two inner loops must be separated by line 8: the computation of $\nabla \mathcal{R}_{\boldsymbol{m}}$ may require all solutions to the primal problems; e.g. the case of the variance in deliverable 6.2, § 3.4.1.

A major drawback of the SAA is the lack of flexibility to adapt the sample size. Consequently the probability-discretisation error can only be controlled a priori, and reliable a priori error estimators are not often available. For this reason, and because a SAA is often more straightforward to develop, we will focus henceforth on the stochastic approximation.

Algorithm 2: SAA-MLMC version of algorithm 1

```
Input: $z_0$, $\gamma$, $\eta$, $\boldsymbol{m}$
Initialise: Draw $\boldsymbol{\omega} \in \Omega^{\boldsymbol{m}}$
While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
    For $l \in \{0 \dots L\}$ do
        For $i \in \{1 \dots m_l\}$ do
            Find $u_l(\omega_{l,i})$ s.t. $F_l(u_l(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
            Find $u_{l-1}(\omega_{l,i})$ s.t. $F_{l-1}(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
    Compute $\nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega)$, $\forall \omega \in \boldsymbol{\omega}$
    For $l \in \{0 \dots L\}$ do
        For $i \in \{1 \dots m_l\}$ do
            Find $\lambda_l(\omega_{l,i})$ s.t.
              $\mathrm{D}_1 F_l^*(u_l(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_l(\omega_{l,i})) = q \nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega_{l,i})$
            Find $\lambda_{l-1}(\omega_{l,i})$ s.t.
              $\mathrm{D}_1 F_{l-1}^*(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_{l-1}(\omega_{l,i})) = q \nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega_{l,i})$
    Compute descent direction
      $\nabla J_{\boldsymbol{m}}(z_k) := \nabla P(z_k) - \mu_{\boldsymbol{m}}(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda))$
    Set new design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J_{\boldsymbol{m}}(z_k))$
Result: $z_{k+1}$
```

2.3.3 Stochastic approximation

This approach picks a new discretisation of the measure $\mathbb{P}$ at every iteration of the algorithm, unlike SAA which fixes it a priori. The main difference with SAA is that the samples are drawn independently at every iteration. We can distinguish two strategies. In the first case, an accurate evaluation of the gradient ∇J is computed at every iteration $k \in \mathbb{N}$ and the step size γ_k is either kept constant or estimated by some backtracking line-search method, as in algorithms 1–2. In the second case, a very crude approximation of the gradient of the risk measure is evaluated at each iteration. This could comprise even only a single realisation (one primal and one adjoint solutions), provided that the estimator is unbiased[3]. In this case, the step size should be progressively reduced over the iterations (e.g. $\gamma_k := \gamma_0 k^{-1}$) to achieve convergence. Since the first case is a straightforward modification of algorithm 2, hereafter we present only the second case.

Stochastic gradient This well-known approach was introduced by Robbins and Monro 1951 for gradient-descent algorithms. If we consider a single-level MC approximation, this approach can be summarised as follows.

$$z_{k+1} = \prod_{\mathcal{A}}\Big(z_k + \frac{\gamma_0}{k}\nabla J_1(z_k)\Big) = \prod_{\mathcal{A}}\Big(z_k + \frac{\gamma_0}{k}(\nabla P(z_k) + \mu_1(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda)))\Big).$$

In particular, only a single event is used by the MC estimator. The gist is that the algorithm will converge regardless of the variance of the estimator of ∇J, provided that it is unbiased, that $\sum_{k=1}^{+\infty} \gamma_k = +\infty$ and $\sum_{k=1}^{+\infty} \gamma_k^2 < +\infty$. It was shown by Martin, Krumscheid et al. 2018, (see table 1 p. 21) that it could achieve better complexity than SAA, for a strongly-convex, Lipschitz-continuous optimal control problem.

In our context, an obvious drawback is the lack of parallelisation. However, the single sample can be replaced with a small sample size, still delivering inaccurate gradient estimations, yet leveraging parallelisation. E.g., we may choose the sample size at every iteration so as to use all computing resources currently available. This strategy is sometimes called ‘stochastic gradient with mini-batches’.

Combinations of the stochastic gradient (SG) with MLMC estimators have been proposed in Martin, Nobile et al. 2019. We review hereafter the two strategies they developed: multi-level stochastic gradient (MLSG) and randomised multi-level stochastic gradient (RMLSG).

[3] The estimator is not always unbiased; we will come back to this issue when discussing SA methods for the CVaR.

Multi-level stochastic gradient This is the intuitive generalisation of the SG method from MC estimators to MLMC ones: at every step k, $\boldsymbol{m}_k \in \mathbb{N}^{L_k+1}$ is chosen and $\omega \in \Omega^{\boldsymbol{m}_k}$ is drawn. The rest is identical to algorithm 2, using the MLMC estimator $\mu_{\boldsymbol{m}_k}$.

Since the SG approach relies on an unbiased estimation to converge, the bias error $\mathbb{E}(X - X_L)$ defined in § 2.3.1 is an issue here. This can be solved by choosing the sequence $(L_k)_{k\in\mathbb{N}}$ such that $\lim_{k\to+\infty} L_k = +\infty$, so that $\lim_{k\to+\infty} \mathbb{E}\big(X - X_{L_k}\big) = 0$. For the choice of the cost-optimal rate of increase of L_k, we refer the reader to Martin, Nobile et al. 2019, theorem 3.7. An alternative, yet less sophisticated, view is the one often taken with single-level MC estimator: to consider X_L to be the reference instead of X, and so disregard this possible source of bias.

Regarding the choice of $\boldsymbol{m}_k$, this should be chosen as small as possible while using all available computing resources. Adaptive algorithms such as described by Giles 2015 and Collier et al. 2015 can be used to distribute the resources over the levels (through the samples sizes $\boldsymbol{m}_k$) so as to minimise the complexity of the MLMC estimator. We refer again to Martin, Nobile et al. 2019 for cost-optimal choices of $\boldsymbol{m}_k$.

Randomised multi-level stochastic gradient Let us define a discrete probability measure p on $\{0 \dots L\}$; for any $l \in \{0 \dots L\}$, we denote the probability mass by $p_l := p(l)$. The RMLSG estimator originally proposed by Rhee and Glynn 2015 draws a single sample from the level l randomly chosen. With the convention $X_{-1} := 0$, the estimator reads

$$\mu_{1,p}(X) := \frac{1}{p_l}(X_l - X_{l-1}), \quad l \sim p.$$

In the interest of parallelisation, this estimator can be replaced with a mini-batch of size $m \in \mathbb{N}$ as

$$\mu_{m,p}(X) := \frac{1}{mp_l} \sum_{i=1}^{m} X_l(\omega_i) - X_{l-1}(\omega_i); \qquad l \sim p;\ \omega_i \sim \mathbb{P} \text{ i.i.d.}$$

The issue of the bias error applies here as well as for the MLSG case. The original proposition by ibid. was to define p over $\mathbb{N}$, which leads to an unbiased estimator. However, the cost of this unbiased estimator has a large variance and entails a risk of untractable computations. We prefer the solution discussed for the MLSG estimator: to choose a sequence of highest levels L_k such that $\lim_{k\to+\infty} L_k = +\infty$, leading to a RMLSG estimator that is asymptotically unbiased.

Algorithm 3 illustrates this approach, whose consistency relies on $\nabla J_{m,p}(z_k)$ being an unbiased estimator of ∇J independently of m, at least as the highest-level L diverges to infinity along the iterations. Therefore, on line 3, L is chosen so as to satisfy a tolerance ϵ_k on an estimator E_{bias} of the bias error, where the sequence $\epsilon \subset]0, +\infty[$ is chosen a priori and $\lim_{k\to+\infty} \epsilon_k = 0$. The sample size chosen on line 6 would be 1 for the canonical RMLSG estimator; for parallel computations, one may want to choose it according to computational resources.

Algorithm 3: Sa-RMLSG version of algorithm 1

```
1  Input: $z_0$, $\gamma$, $\eta$, $\epsilon$
2  While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
3      Set highest level $L := \min\{l \in \mathbb{N} : E_{\text{bias}}(l) \leqslant \epsilon_k\}$
4      Choose probability masses $p_l = p(l)$, $\forall l \in \{0 \dots L\}$, s.t. $\sum_{l=0}^{L} p_l = 1$
5      Draw level $l \sim p$
6      Pick sample size $m \in \mathbb{N}$ and draw $\boldsymbol{\omega} \in \Omega^m$
7      For $i \in \{1 \dots m\}$ do
8          Find $u_l(\omega_i)$ s.t. $F_l(u_l(\omega_i), z_k, \omega_i) = 0$
9          Find $u_{l-1}(\omega_i)$ s.t. $F_{l-1}(u_{l-1}(\omega_i), z_k, \omega_i) = 0$
10     Compute $\nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega_i)$, $\forall i \in \{1 \dots m\}$
11     For $i \in \{1 \dots m\}$ do
12         Find $\lambda_l(\omega_i)$ s.t. $\mathrm{D}_1 F_l^*(u_l(\omega_i), z_k, \omega_i)(\lambda_l(\omega_i)) = q \nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega_i)$
13         Find $\lambda_{l-1}(\omega_i)$ s.t.
             $\mathrm{D}_1 F_{l-1}^*(u_{l-1}(\omega_i), z_k, \omega_i)(\lambda_{l-1}(\omega_i)) = q \nabla \mathcal{R}_{\boldsymbol{m}}(Q(u))(\omega_i)$
14     Compute descent direction
         $\nabla J_{m,p}(z_k) := \nabla P(z_k) - \mu_{m,p}(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda))$
15     Set new design $z_{k+1} := \prod_{\mathcal{A}} \big(z_k - \gamma_k \nabla J_{m,p}(z_k)\big)$
16 Result: $z_{k+1}$
```

3 Approaches to cvaR optimisation

Section 2 discussed general OUU for a smooth risk measure. In this section, we will introduce a non-smooth risk measure and discuss approaches to optimise it, as well as their practical implementation. The choice of this risk measure is motivated by its suitable mathematical properties as well as its usefulness and popularity in risk-averse engineering design. We refer the reader to Rockafellar and Royset 2015 for a detailed review of risk measures pertinent to engineering risk-averse decisions.

3.1 Definition and properties of the cvaR

To introduce the cvaR, we first define the value at risk (vaR).

Definition 3 (Value at risk)**.** Let $\beta \in]0,1[$. The value at risk (vaR) of significance[4] β of any $X \in \mathrm{L}^1(\Omega,\mathbb{R})$ is defined as

$$\mathrm{vaR}_\beta(X) := \inf\{t \in \mathbb{R} : \mathbb{P}(X \leqslant t) \geqslant \beta\}$$

Although the vaR provides useful information on the tails of the distribution, it is *not a coherent risk measure*: it is not sub-additive[5]; a fortiori not convex. Nevertheless, it is closely related to the cvaR, and useful to define it.

Definition 4 (Conditional value at risk)**.** Let $\beta \in]0,1[$ and $X \in \mathrm{L}^1(\Omega,\mathbb{R})$. We define the cvaR of significance β of X as the following expectation, conditional on the vaR:

$$\mathrm{cvaR}_\beta(X) := \mathbb{E}\big(X \mid X \geqslant \mathrm{vaR}_\beta(X)\big).$$

The cvaR is well-defined over $\mathrm{L}^1(\Omega,\mathbb{R})$, and has been proven to be a coherent risk measure by Pflug 2000. The cvaR has been presented with several different definitions in the literature (cf. Rockafellar and Uryasev 2000), generally equivalent for random variables whose CDF is sufficiently regular. For simplicity, we will assume that the random variables whose cvaR are considered have an absolutely continuous CDF, as we did in section 2. Henceforth we will consider $\mathcal{R} := \mathrm{cvaR}_\beta$, and discuss two alternative approaches to solve the OUU problem 1 for this definition of the risk measure.

3.2 Pure gradient-descent

Here we propose an adaptation of the gradient-descent method introduced in § 2.2 for the OUU problem 1, with the cvaR as the risk measure.

[4] Sometimes called 'value at risk $1-\beta$'.

[5] $\exists (X,Y) \in \mathrm{L}^1(\Omega,\mathbb{R})^2 : \mathrm{vaR}_\beta(X+Y) > \mathrm{vaR}_\beta(X) + \mathrm{vaR}_\beta(Y)$.

3.2.1 Idealised setting

The computation of a $\textsc{cVaR}$ is generally not trivial, and definition 4 is not the most convenient for our practical use. Assuming that $X \in \mathrm{L}^1(\Omega, \mathbb{R})$ has an absolutely continuous CDF, Rockafellar and Uryasev 2000 proved that

$$\textsc{cVaR}_\beta(X) = \inf\left\{ \overbrace{t + \frac{1}{1-\beta}\mathbb{E}((X-t)^+)}^{R(X,t)} : t \in \mathbb{R} \right\} \tag{3.1}$$

with $(\cdot)^+ := \max(0,\cdot)$. This infimum is reached on a closed, bounded interval whose minimum is $\textsc{VaR}_\beta(X)$.

Since definition (3.1) of the $\textsc{cVaR}$ features an infimum, the OUU problem 1 appears as a nested minimisation problem: $\inf_{\mathcal{A}} \inf_{\mathbb{R}}(\cdot)$. It is convenient to consider therefore problem 3, i.e. an equivalent minimisation problem over $\mathcal{A} \times \mathbb{R}$. Particularly, the objective function does not feature $\textsc{VaR}_\beta$ any more, which instead is now part of the solution, as $t^\star$.

Problem 3 (CVaR optimisation)**.** Find the solution $(z^\star, t^\star) \in \mathcal{A} \times \mathbb{R}$ of

$$\begin{cases} \min\{J(z,t) := R(Q(u(z)),t) + P(z) : z \in \mathcal{A};\ t \in \mathbb{R}\} \\ \qquad \text{s.t. } F(u(z)(\omega), z, \omega) = 0, \text{ for } \mathbb{P}\text{-a.e. } \omega \in \Omega \end{cases}$$

Algorithm 1 can easily be adapted to this new formulation as a gradient-descent over $\mathcal{A} \times \mathbb{R}$. For the derivatives of R to be well-defined and continuous, we assume that the CDF $t \mapsto \mathbb{P}(Q(u(z)) \leqslant t)$ is absolutely continuous and depends smoothly on $z \in \mathcal{A}$. In addition to (2.4)–(2.6), a solution $(z,t) \in \mathcal{A} \times \mathbb{R}$ of problem 3 has to fulfil the following optimality condition:

$$\mathrm{D}_2\, R(Q(u(z)),t) = 1 - \frac{1}{1-\beta}\mathbb{E}(\mathbb{1}(Q(u(z)) \geqslant t)) = 0. \tag{3.2}$$

With the new definition of the objective function, the gradient of the risk measure, as featured in line 4 of algorithm 1, is now

$$\nabla_1\, R(X,t) = \frac{\mathbb{1}(X \geqslant t)}{1-\beta}. \tag{3.3}$$

We can then particularise algorithm 1 for $\mathcal{R} := \textsc{cVaR}_\beta$ as algorithm 4 below. The gradient descent over $\mathcal{A} \times \mathbb{R}$ is done consecutively over $\mathbb{R}$ in line 4, then over $\mathcal{A}$ in line 7. The linearity assumption on Q and formula (2.7) have also been used on line 5. Although a normal gradient-descent algorithm would use t_k at line 5, we chose to use the more accurate t_{k+1} since it incurs no extra cost in this case. The step size γ'_k, as for γ_k, may be chosen either a priori or adaptively. This is left unspecified in algorithm 4.

Algorithm 4: Ideal gradient-descent for the cVaR

```
Input: $(z_0, t_0)$, $\eta$,
While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
    Find $u$ s.t. $F(u(\omega), z_k, \omega) = 0$ for $\mathbb{P}$-a.e. $\omega \in \Omega$
    Update quantile estimation $t_{k+1} := t_k - \gamma'_k \big(1 - \frac{1}{1-\beta} \mathbb{E}(\mathbb{1}(Q(u) \geqslant t_k))\big)$
    Find $\lambda$ s.t., for $\mathbb{P}$-a.e. $\omega \in \Omega$,
      $\mathrm{D}_1 F^*(u(\omega), z_k, \omega)(\lambda(\omega)) = \frac{q}{1-\beta} \mathbb{1}(Q(u) \geqslant t_{k+1})(\omega)$
    Compute descent direction $\nabla J(z_k) := \nabla P(z_k) - \mathbb{E}(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda))$
    Update design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J(z_k))$
Result: $(z_{k+1}, t_{k+1})$
```

3.2.2 Stochastic approximation using subgradients

As for the smooth case in § 2, the practical implementation of algorithm 4 will require a discretisation of the probability space as well as the underlying PDE. The major difference here lies in the weaker regularity of the objective function. For example, the objective function of problem 3 features a term of the form $\mathbb{E}((X - t)^+)$, which is continuously differentiable as a function of either $t \in \mathbb{R}$ or $X \in \mathrm{L}^1(\Omega, \mathbb{R})$ (with absolutely continuous CDF). However, if we approximate $\mathbb{P}$ as in § 2.3, with a discrete measure from a set of events $\boldsymbol{\omega} \in \Omega^m$ ($m \in \mathbb{N}$), this term is approximated as

$$\mathbb{E}((X - t)^+) \approx \mu_m((X - t)^+) = \frac{1}{m} \sum_{i=1}^{m} (X(\omega_i) - t)^+,$$

which is not continuously differentiable any more, with respect to neither X nor t. Obviously, the same consideration applies to MLMC estimators. Consequently, the discretised risk measure – and thus the objective function – is not continuously differentiable. Nevertheless, for convex functions one can define subderivatives.

Definition 5 (Subderivative)**.** Let B be a Banach space, $C \subset B$ a convex set and $f : C \to \mathbb{R}$ a convex function. For any $c \in C$, we define

$$\partial f(c) := \{v^* \in B^* : \forall v \in C,\ f(v) - f(c) \geqslant v^*(v - c)\},$$

the 'subdifferential' of f at c, i.e. the set of subgradients of f at c. This set is not empty, even if f is not differentiable at c; iff. it is, $\partial f(c) = \{\nabla f(c)\}$. For a multivariate function f, we will denote by $\partial_k f$ the subdifferential of the k-th partial application of f.

It is therefore possible to extend the definition of the gradient of a non-smooth convex risk measure such as the discretised CVaR by choosing a subgradient at points where the gradient is not defined. For any $(X,t) \in \mathrm{L}^r(\Omega,\mathbb{R}) \times \mathbb{R}$,

$$\partial_1 R(X,t) = \{\mathbb{1}(X>t) + \alpha\mathbb{1}(X=t) : \alpha \in [0,1]\} \subset \mathrm{L}^{r'}(\Omega,\mathbb{R}), \tag{3.4}$$

$$\partial_2 R(X,t) = \{1 - \mathbb{E}(\mathbb{1}(X>t) + \alpha\mathbb{1}(X=t)) : \alpha \in [0,1]\} \subset \mathbb{R}. \tag{3.5}$$

It is clear from the expressions (3.4)–(3.5) that R is differentiable at (X,t) iff. $\mathbb{P}(X=t)=0$. If the CDF of X is discontinuous at t, R has only subderivatives at (X,t) – no derivative. If X has an absolutely continuous CDF, R is differentiable; however its approximation for a *discrete* probability measure μ_m is not if $\mu_m(X = t) > 0$, hence the need for subgradients. The calculations leading to (3.4)–(3.5) can be found in appendix A.

We propose to adapt algorithm 4 to an empirical measure by following a SA approach, due to its greater flexibility regarding sample sizes. The convergence analysis of a SG method usually rely on the assumption that the gradient of the objective function is at least Lipschitz-continuous, which is not the case here. However, Shamir and Zhang 2012, theorem 2 proved convergence of $\mathbb{E}\Big(\tilde{J}(z_k) - \min\tilde{J}\Big)$ for a non-smooth convex objective function $\tilde{J}$, albeit at a suboptimal[6] rate of $\mathcal{O}(k^{-1/2}\log(k))$ (for $k \in \mathbb{N}$ iterations).

Algorithm 5 proposes an adaptation of idealised algorithm 4 to the MLSG method introduced in § 2.3.3. A sequence of decreasing tolerances $(\epsilon_k)_{k\in\mathbb{N}} \subset \,]0,+\infty[$ converging to zero is set a priori. Then, on line 3, the MLMC estimator is tuned so that the current tolerance ϵ_k is satisfied by the MSE estimated based on data from the previous iteration. In practice, this adaptivity criterion is actually split between a condition on the bias error and another one on the statistical error, to choose respectively the highest level L and sample sizes $\boldsymbol{m}$. Martin, Nobile et al. 2019, theorem 3.7 proposed an analysis of rates at which L and $\boldsymbol{m}$ should increase to achieve optimal asymptotic complexity, albeit for an optimal-control problem with a different risk measure.

For simplicity and brevity, algorithm 5 uses the same MLMC estimator $\mu_{\boldsymbol{m}}$ on lines 9 and 14, and that estimator is adapted for the latter, i.e. the descent direction in the design space. However, the MLSG strategy introduced in § 2.3.3 would not adapt these estimator based on error estimations but instead use a crude, cost-efficient estimation with a small number of samples, e.g. chosen according to available computing resources. Alternatively, one may choose to estimate accurately the descent direction by tuning its estimation separately. This can be achieved by defining another sequence of tolerances $(\epsilon'_k)_{k\in\mathbb{N}}$ converging to zero, then using a different estimator $\mu_{\boldsymbol{m}'}$ on line 9 with $\boldsymbol{m}' \in \mathbb{N}^{L'+1}$ and $L' \in \mathbb{N}$ chosen such that

[6] Compared to the SG convergence rate of $\mathcal{O}(k^{-1})$, for a smooth convex objective function.

MSE$(\mu_{\boldsymbol{m}'}(\mathbb{1}(Q(u) \geqslant t_k))) \leqslant \epsilon'_k$. The total cost of these estimations can be reduced by sharing samples of u between $\mu_{\boldsymbol{m}'}$ and $\mu_{\boldsymbol{m}}$. With respect to the samples of λ: the higher β is, the more likely the right-hand side of the adjoint equations on lines 12–13 is to be zero, which reduces the number of resolutions of the adjoint problem to compute.

Algorithm 5: Sa-mlsg version of algorithm 4

```
1  Input: $(z_0, t_0)$, $(\gamma, \gamma')$, $\epsilon$, $\eta$
2  While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
3      Choose $L \in \mathbb{N}$ and $\boldsymbol{m} \in \mathbb{N}^{L+1}$ s.t. MSE$(\mu_{\boldsymbol{m}}(\mathrm{D}_2\, F^*(u, z_{k-1}, \cdot))) \leqslant \epsilon_k$
4      Draw $\boldsymbol{\omega} \in \Omega^{\boldsymbol{m}}$
5      For $l \in \{0 \dots L\}$ do
6          For $i \in \{1 \dots m_l\}$ do
7              Find $u_l(\omega_{l,i})$ s.t. $F_l(u_l(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
8              Find $u_{l-1}(\omega_{l,i})$ s.t. $F_{l-1}(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
9      Update quantile estimation $t_{k+1} := t_k - \gamma'_k \big(1 - \frac{1}{1-\beta} \mu_{\boldsymbol{m}}(\mathbb{1}(Q(u) \geqslant t_k))\big)$

10     For $l \in \{0 \dots L\}$ do
11         For $i \in \{1 \dots m_l\}$ do
12             Find $\lambda_l(\omega_{l,i})$ s.t.
                 $\mathrm{D}_1\, F_l^*(u_l(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_l(\omega_{l,i})) = \frac{q}{1-\beta} \mathbb{1}(Q(u_l(\omega_{l,i})) \geqslant t_{k+1})$
13             Find $\lambda_{l-1}(\omega_{l,i})$ s.t. $\mathrm{D}_1\, F_{l-1}^*(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_{l-1}(\omega_{l,i})) =$
                 $\frac{q}{1-\beta} \mathbb{1}(Q(u_{l-1}(\omega_{l,i})) \geqslant t_{k+1})$
14     Compute descent direction
         $\nabla J_{\boldsymbol{m}}(z_k) := \nabla P(z_k) - \mu_{\boldsymbol{m}}(\mathrm{D}_2\, F^*(u, z_k, \cdot)(\lambda))$
15     Set new design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J_{\boldsymbol{m}}(z_k))$
16 Result: $z_{k+1}$
```

Besides the generic MSE defined in § 2.3.1, noteworthy alternative error estimators have been proposed specifically for gradient-descent algorithms, such as the 'norm test' by Byrd et al. 2012 or the 'inner product test' by Bollapragada et al. 2018. Both have been studied recently by Urbainczyk 2020, leading to adaptive MC sampling strategies for constrained shape optimisation under uncertainties.

3.3 Gradient-descent with accurate VaR estimation

In this section we propose an approach alternative to that of § 3.2, motivated by the method proposed by Krumscheid and Nobile 2018 for the estimation of parametric expectations.

3.3.1 Idealised setting

Section 3.2 proposed to solve problem 3 by a gradient-descent algorithm over $\mathcal{A} \times \mathbb{R}$. A different tactic would be to minimise alternatively over $\mathcal{A}$ and $\mathbb{R}$, using a gradient-descent for the former and a direct minimisation for the latter. At every iteration $k \in \mathbb{N}$, we first find $u(z_k) \in \mathrm{L}^p(\Omega, U)$ verifying $F(u(z_k)(\cdot), z_k, \cdot)$ a.e. in Ω. Then we compute $\mathrm{VaR}_\beta(Q(u_{z_k}))$; this also yields $\mathrm{CVaR}_\beta(Q(u(z_k))) = R(Q(u(z_k)), \mathrm{VaR}_\beta(Q(u_{z_k})))$. Finally, we proceed to update the design as in § 3.2: by solving the adjoint problem to compute the descent direction, using the exact VaR. This leads to algorithm 6, to be compared with algorithm 4.

Algorithm 6: Ideal gradient-descent with exact VaR

```
Input: $z_0$, $\gamma$, $\eta$
While $\|z_{k+1} - z_k\| \geqslant \eta\|z_k\|$ do
    Find $u$ s.t. $F(u(\omega), z_k, \omega) = 0$ for $\mathbb{P}$-a.e. $\omega \in \Omega$
    Find VaR $t_{k+1} := \operatorname{argmin}\{R(Q(u), r) : r \in \mathbb{R}\}$
    Find $\lambda$ s.t., for $\mathbb{P}$-a.e. $\omega \in \Omega$,
      $\mathrm{D}_1 F^*(u(\omega), z_k, \omega)(\lambda(\omega)) = \frac{q}{1-\beta}\mathbb{1}(Q(u) \geqslant t_{k+1})(\omega)$
    Compute descent direction $\nabla J(z_k) := \nabla P(z_k) - \mathbb{E}(\mathrm{D}_2 F^*(u, z_k, \cdot)(\lambda))$
    Update design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J(z_k))$
Result: $z_{k+1}$
```

3.3.2 Multi-level stochastic gradient algorithm

As in section 3.2, we propose to adapt the idealised algorithm to an empirical measure in a MLSG method. In particular, we extend the definition of gradients of $\mathcal{R}$ using subgradients, in exactly the same way as described in § 3.2.2.

The major difference is the estimation of VaR_β, for which we use the method proposed by Krumscheid and Nobile 2018. Let $X \in \mathrm{L}^1(\Omega, \mathbb{R})$; $R(X, t)$ is an expectation parameterised by $t \in \mathbb{R}$:

$$R(X, t) = \mathbb{E}(\phi(t, X)) \quad \text{with} \quad \phi(t, X) := t + \frac{(X - t)^+}{1 - \beta}.$$

Let I be a suitable interpolator (e.g. spline), $(L, n) \in \mathbb{N}^2$ and $(\boldsymbol{m}, \boldsymbol{r}) \in \mathbb{N}^{L+1} \times \mathbb{R}^n$. The parametric expectation is estimated as

$$\forall t \in \mathbb{R}, \quad R(X, t) \approx I(\{\mu_{\boldsymbol{m}}(\phi(r_i, X)) : i \in \{1 \dots n\}\})(t) =: \Phi_{\boldsymbol{m},\boldsymbol{r}}(X)(t), \tag{3.6}$$

from which one can estimate

$$\textsc{VaR}_\beta(X) \approx \operatorname{argmin} \Phi_{\boldsymbol{m},\boldsymbol{r}}(X) \quad \text{and} \quad \textsc{CVaR}_\beta(X) \approx \min \Phi_{\boldsymbol{m},\boldsymbol{r}}(X).$$

For any $\epsilon \in]0, +\infty[$, a posteriori error estimators enable to choose n, L and $\boldsymbol{m}$ such that

$$\textsc{MSE}\big(\operatorname{argmin} \Phi_{\boldsymbol{m},\boldsymbol{r}}(X)\big) := \mathbb{E}\Big(\big|\textsc{VaR}_\beta(X) - \operatorname{argmin} \Phi_{\boldsymbol{m},\boldsymbol{r}}(X)\big|^2\Big) \leqslant \epsilon. \tag{3.7}$$

In the same way that (2.9) was split two-way, in practice this MSE-adaptivity criterion is split into criteria on the interpolation, bias and statistical errors to choose respectively n, L and $\boldsymbol{m}$. We refer the reader to Krumscheid and Nobile 2018 for further details.

Algorithm 7 uses this method in a MLSG version of algorithm 6. Unlike algorithm 5, here we tune the MLMC estimator used for the VaR distinctly from the estimator used for the descent direction in the design space. Consequently, we have two sequences of tolerances $\epsilon, \epsilon' \subset]0, +\infty[$ set a priori. A typical stochastic-gradient strategy would dispense with ϵ' and use a small sample size $\boldsymbol{m}'$ without regard for the accuracy (e.g. kept constant or chosen according to computing resources) provided $\mu_{\boldsymbol{m}'}$ is unbiased – at least asymptotically, as the algorithm iterates. However, this requires a quantile estimation with controlled and increasing accuracy, hence the need for $\mu_{\boldsymbol{m}'}$ to be tuned with respect to a diminishing sequence of tolerances ϵ in any case. Let us note that the error estimations in lines 3 and 10 are made a posteriori, based on data from the previous iteration. To reduce the total cost of these estimations, samples of u are shared between the two estimators. Obviously, the comment made on algorithm 5 about the right-hand side of the adjoint equations applies here as well. Appendix B presents a variation of this algorithm on a more specific OUU problem.

Algorithm 7: Sa-mlsg version of algorithm 6

```
1  Input: $z_0$, $\gamma$, $(\epsilon, \epsilon')$, $\eta$
2  While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
3      Choose $(L', n) \in \mathbb{N}^2$ and $(\boldsymbol{m'}, \boldsymbol{r}) \in \mathbb{N}^{L'+1} \times \mathbb{R}^n$ s.t.
         $\mathrm{MSE}\big(\mathrm{argmin}\, \Phi_{\boldsymbol{m'},\boldsymbol{r}}(Q(u))\big) \leqslant \epsilon'_k$
4      Draw $\boldsymbol{\omega} \in \Omega^{\boldsymbol{m'}}$
5      For $l \in \{0 \dots L'\}$ do
6          For $i \in \{1 \dots m'_l\}$ do
7              Find $u_l(\omega_{l,i})$ s.t. $F_l(u_l(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
8              Find $u_{l-1}(\omega_{l,i})$ s.t. $F_{l-1}(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i}) = 0$
9      Estimate VaR as $t_{k+1} := \mathrm{argmin}\, \Phi_{\boldsymbol{m'},\boldsymbol{r}}(Q(u))$
10     Choose $L \in \mathbb{N}$ and $\boldsymbol{m} \in \mathbb{N}^{L+1}$ s.t. $\mathrm{MSE}(\mu_{\boldsymbol{m}}(\mathrm{D}_2\, F^*(u, z_{k-1}, \cdot))) \leqslant \epsilon_k$
11     For $l \in \{0 \dots L\}$ do
12         For $i \in \{1 \dots m_l\}$ do
13             If $i > m'_l$ then
14                 Draw $\omega_{l,i} \in \Omega$
15                 Compute $u_l(\omega_{l,i})$ and $u_{l-1}(\omega_{l,i})$ as in lines 7–8.
16             Find $\lambda_l(\omega_{l,i})$ s.t.
                 $\mathrm{D}_1\, F^*_l(u_l(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_l(\omega_{l,i})) = \frac{q}{1-\beta}\mathbb{1}(Q(u_l(\omega_{l,i})) \geqslant t_{k+1})$
17             Find $\lambda_{l-1}(\omega_{l,i})$ s.t. $\mathrm{D}_1\, F^*_{l-1}(u_{l-1}(\omega_{l,i}), z_k, \omega_{l,i})(\lambda_{l-1}(\omega_{l,i})) =$
                 $\frac{q}{1-\beta}\mathbb{1}(Q(u_{l-1}(\omega_{l,i})) \geqslant t_{k+1})$
18     Compute descent direction
         $\nabla J_{\boldsymbol{m}}(z_k) := \nabla P(z_k) - \mu_{\boldsymbol{m}}(\mathrm{D}_2\, F^*(u, z_k, \cdot)(\lambda))$
19     Set new design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J_{\boldsymbol{m}}(z_k))$
20 Result: $z_{k+1}$
```

4 Regularisation for CVaR optimisation

The CVaR is not differentiable because of the positive part $(\cdot)^+$ featured in (3.1), as was mentioned in § 3.2.2 where we considered subgradients. Alternative to the use of subgradients, we discuss here possibilities to regularise problem 3.

4.1 Smooth optimisation

We consider first 'smoothing' the risk measure, i.e. approximate it with a more regular function. An immediate benefit would be to place ourselves in the smooth case described in § 2. Additionally, a higher regularity of the objective function would allow for Newton-like optimisation methods (e.g. Broyden–Fletcher–Goldfarb–Shanno method), which could significantly accelerate the optimisation process. Another benefit is to enable SAA and deterministic quadrature methods (e.g. quasi Monte Carlo, sparse grids), to accelerate quadrature by exploiting the regularity of the random variable. This section is based on the works of Kouri and Surowiec 2016; Chen and Mangasarian 1995.

4.1.1 Smooth approximation of the CVaR

We propose to define a family of functions $\{(\cdot)^+_\varsigma : \varsigma \in \,]0,+\infty[\}$ to approximate the non-differentiable positive part in R. For any $x \in \mathbb{R}$, we let

$$(x)^+_\varsigma := \int_{-\infty}^{x} \int_{-\infty}^{y} \frac{1}{\varsigma} g\left(\frac{z}{\varsigma}\right) \mathrm{d}z \, \mathrm{d}y$$

where g is a function satisfying the following assumptions.

1. $\exists a \in \,]0,+\infty[\, : g \in \mathrm{C}^0(\mathbb{R},[0,a])$.
2. $\int_{\mathbb{R}} g = 1$.
3. $\int_{\mathbb{R}} g(x)|x| \,\mathrm{d}x \in \mathbb{R}$.
4. Either $g_1 := \int_{\mathbb{R}} g(x)x \,\mathrm{d}x \leqslant 0$ or $g_2 := \int_{-\infty}^{0} g(x)|x| \,\mathrm{d}x = 0$.
5. $\operatorname{supp} g$ is connected.

From this definition, $(\cdot)^+_\varsigma \in \mathrm{C}^2(\mathbb{R})$, is increasing and convex. Besides,

$$\exists b \in \,]0, \max\{g_1, g_2\}],\ \forall x \in \mathbb{R}, \quad |(x)^+_\varsigma - (x)^+| \leqslant b\varsigma. \tag{4.1}$$

From this, Kouri and Surowiec 2016, lemma 4.3 proved

$$\forall X \in \mathrm{L}^1(\Omega), \quad |\textsc{cVaR}_{\beta,\varsigma}(X) - \textsc{cVaR}_\beta(X)| \leqslant \frac{b\varsigma}{1-\beta}, \tag{4.2}$$

with the 'smoothed cVaR' defined as

$$\textsc{cVaR}_{\beta,\varsigma}(X) := \inf\Bigg\{ \underbrace{t + \frac{1}{1-\beta}\mathbb{E}\big((X-t)^+_\varsigma\big)}_{R(X,t,\varsigma)} : t \in \mathbb{R} \Bigg\}. \tag{4.3}$$

Like the cVaR, this smoothed cVaR is a coherent risk measure.

Below are examples of suitable choices[7] of $(\cdot)^+_\varsigma$ for any $x \in \mathbb{R}$:

$$(x)^+_{\varsigma,1} := x + \varsigma \ln\left(1 + \exp\left(\frac{-x}{\varsigma}\right)\right), \qquad b = \ln 2;$$

$$(x)^+_{\varsigma,2} := \begin{cases} 0 \text{ if } x \leqslant 0, \\ \dfrac{x^3}{\varsigma^2} - \dfrac{x^4}{2\varsigma^3} \text{ if } x \in]0,\varsigma[, \\ x - 0.5\varsigma \text{ if } x \geqslant \varsigma, \end{cases} \qquad b = \frac{1}{2};$$

$$(x)^+_{\varsigma,3} := \left(x + \frac{\varsigma}{2}\right)^+_{\varsigma,2}, \qquad b = \frac{3}{32}.$$

These examples are illustrated on figure 1. In particular, $\forall x \in \mathbb{R}$, they are ordered as

$$(x)^+_{\varsigma,2} \leqslant (x)^+ \leqslant (x)^+_{\varsigma,3} \leqslant (x)^+_{\varsigma,1}.$$

4.1.2 Adaptation of previous algorithms

Let $\varsigma \in]0,+\infty[$ and let us note s_ς the derivative of $(\cdot)^+_\varsigma$: $s_\varsigma \in \mathrm{C}^1(\mathbb{R})$. By replacing $R(\cdot,\cdot)$ with $R(\cdot,\cdot,\varsigma)$, optimality condition (3.2) becomes

$$\mathrm{D}_2\, R(X,t,\varsigma) = 1 - \frac{1}{1-\beta}\mathbb{E}\big(s_\varsigma(X-t)\big) = 0, \tag{4.4}$$

and (3.3) is now

$$\nabla_1\, R(X,t,\varsigma) = \frac{s_\varsigma(X-t)}{1-\beta}. \tag{4.5}$$

Therefore, algorithms 4 and 6 can be adapted to this smoothed cVaR by using (4.4)–(4.5) instead of (3.2)–(3.3); i.e. replacing expressions of the form $\mathbb{1}(Q(u) \geqslant t)$ with $s_\varsigma(Q(u)-t)$ in lines 4 and 5 of algorithm 4, and line 5 of algorithm 6.

[7] Proposed by Kouri and Surowiec 2016

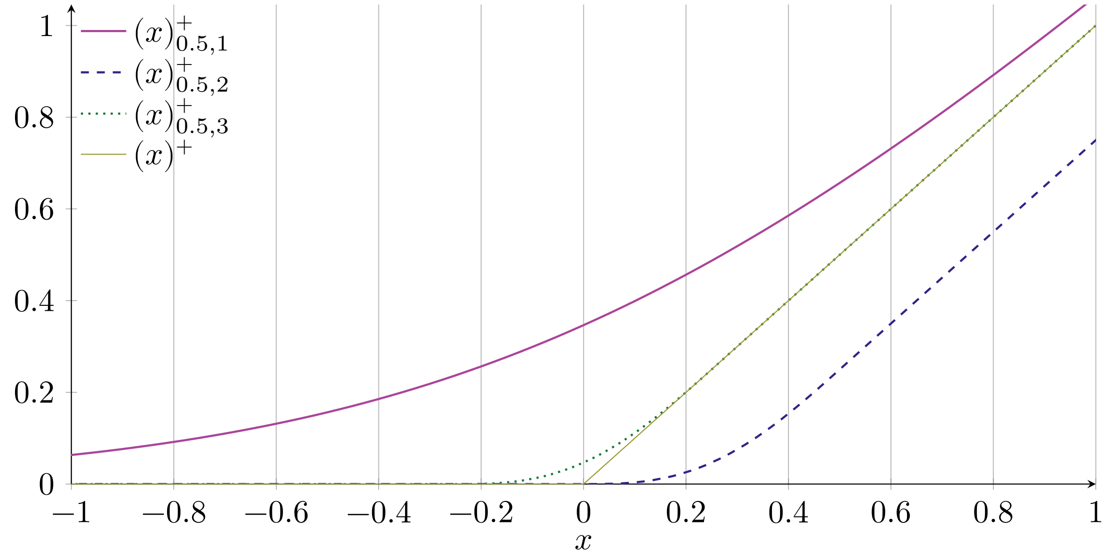

Figure 1: Example of smooth approximations of the positive part

Moreover, there is no more need to extend the gradient definition with subgradients, as was proposed in § 3.2, and it is possible to recover the optimal convergence rate of the SG method.

A priori error estimators (4.1)–(4.2) enable us to choose the smoothing parameter ς so as to control the 'smoothing error' introduced by our approximation of $(\cdot)^+$. As for the bias and statistical errors in the previous MLSG algorithms, we wish this smoothing error to converge to zero along the optimisation process.

In particular, algorithm 7 can be adapted to this smoothing approximation by first defining $\Phi_{\boldsymbol{m},\boldsymbol{r},\varsigma}$ as in (3.6):

$$\forall t \in \mathbb{R}, \quad \Phi_{\boldsymbol{m},\boldsymbol{r},\varsigma}(X)(t) := I(\{\mu_{\boldsymbol{m}}(\phi(r_i, X, \varsigma)) : i \in \{1 \dots n\}\})(t) \approx R(X, t, \varsigma),$$

with

$$\phi(t, X, \varsigma) := t + \frac{(X - t)^+_\varsigma}{1 - \beta}.$$

Recalling the MSE-adaptivity criterion (3.7) and the a priori error estimation (4.1), for any $X \in \mathrm{L}^1(\Omega, \mathbb{R})$ and $\epsilon \in]0, +\infty[$ we can choose $\varsigma \in]0, +\infty[$, $(n, L) \in \mathbb{N}^2$ and $\boldsymbol{m} \in \mathbb{N}^{L+1}$ such that

$$\mathbb{E}\big((\mathrm{VaR}_\beta(X) - \operatorname{argmin} \Phi_{\boldsymbol{m},\boldsymbol{r},\varsigma}(X))^2\big) \leqslant \epsilon. \tag{4.6}$$

As for (3.7), the MSE in (4.6) can be split four-way into smoothing, interpolation, bias and statistical errors to choose respectively ς, n, L and $\boldsymbol{m}$.

An example of such regularisation is provided in appendix B, for a MLSG algorithm on a more specific OUU problem.

4.2 Dual approach to cVaR optimisation

We consider here a different characterisation of the cVaR, leading to a reformulation of OUU problem 3 and different computations of sensitivities of the risk measure. A practical gradient-descent algorithm is then proposed for this approach. This section is based on material from Shapiro et al. 2009, ch. 6; Kouri and Surowiec 2016; a similar approach was also taken by Curi et al. 2019. This is a briefer study than the previous sections, to introduce this dual approach to regularisation. It warrants further research towards better understanding and efficiency.

4.2.1 Dual characterisation of the cVaR and its derivative

First, let us define the following set of probability densities bounded by $(1-\beta)^{-1}$:

$$\mathcal{U} := \left\{\theta \in \mathrm{L}^\infty(\Omega, \mathbb{R}) : \theta(\omega) \in \left[0, \frac{1}{1-\beta}\right] \text{ for a.e. } \omega \in \Omega;\ \mathbb{E}(\theta) = 1\right\}.$$

Then,

$$\forall X \in \mathrm{L}^1(\Omega, \mathbb{R}), \quad \text{cVaR}_\beta(X) = \sup\{\mathbb{E}(\theta X) : \theta \in \mathcal{U}\},$$

and

$$\begin{aligned} \partial\,\text{cVaR}_\beta(X) &= \operatorname{argmax}\{\mathbb{E}(\theta X) : \theta \in \mathcal{U}\} \\ &= \left\{ \theta \in \mathcal{U} : \begin{aligned} \theta(\omega) = 0 &\text{ if } X(\omega) < \text{VaR}_\beta(X), \\ \theta(\omega) \in [0, (1-\beta)^{-1}] &\text{ if } X(\omega) = \text{VaR}_\beta(X), \\ \theta(\omega) = (1-\beta)^{-1} &\text{ if } X(\omega) > \text{VaR}_\beta(X). \end{aligned} \right\} \end{aligned} \tag{4.7}$$

Since $\text{cVaR}_\beta(X)$ is formulated as a supremum over a subset of the dual space $\mathrm{L}^\infty(\Omega, \mathbb{R})$ of $\mathrm{L}^1(\Omega, \mathbb{R}) \ni X$, this is generally called a 'dual characterisation' of the cVaR; hence the section title. If this supremum is reached at $\theta^\star \in \mathcal{U}$, $\theta^\star\mathbb{P}$ is sometimes called the 'risk-adjusted' measure, since cVaR_β is the expectation with respect to this measure.

We define the risk measure by adding a strongly-concave regularisation term parameterised by $\varsigma \in]0, +\infty[$:

$$\mathcal{R}_\varsigma(X) := \sup\left\{\mathbb{E}(\theta X) - \frac{1}{2}\varsigma\mathbb{E}(\theta^2) : \theta \in \mathcal{U}.\right\} \tag{4.8}$$

Consequently we consider problem 4, equivalent to problem 3.

Problem 4 (Dual version of problem 3)**.** Let $\varsigma \in]0, +\infty[$; the risk measure $\mathcal{R}_\varsigma$ is defined as per (4.8). Find the solution $z^\star \in \mathcal{A}$ of

$$\begin{cases} \min\{J_\varsigma(z) := \mathcal{R}_\varsigma(Q(u(z))) + P(z) : z \in \mathcal{A}\} \\ \text{s.t. } F(u(z)(\omega), z, \omega) = 0, \text{ for } \mathbb{P}\text{-a.e. } \omega \in \Omega \end{cases}$$

Assuming that P is differentiable and $\hat{Q}(\cdot;\omega) \mapsto z \mapsto Q(u(z)(\omega))$ is differentiable for a.e. $\omega \in \Omega$, Kouri and Surowiec 2016, theorem 5.1, p. 382 implies[8] that the objective function J_ς is Gâteaux-differentiable with

$$\forall (z, \delta z) \in Z^2, \quad \mathrm{D}\, J_\varsigma(z)(\delta z) := \mathbb{E}\Big(\hat{\theta}\, \mathrm{D}_1\, \hat{Q}(z;\omega)(\delta z)\Big) + \mathrm{D}\, P(z)(\delta z).$$

The probability measure $\hat{\theta}$ above is uniquely defined as

$$\begin{aligned} \hat{\theta} &:= \operatorname{argmax}\Big\{\mathbb{E}(\theta Q) - \frac{1}{2}\varsigma\,\mathbb{E}(\theta^2) : \theta \in \mathcal{U}\Big\} \\ &= \frac{(\hat{Q}(z;\cdot) - \nu)^+}{\varsigma} - \left(\frac{\hat{Q}(z;\cdot) - \nu}{\varsigma} - \frac{1}{1-\beta}\right)^+, \end{aligned} \tag{4.9}$$

where ν satisfies

$$\mathbb{E}\left(\frac{(\hat{Q}(z;\cdot) - \nu)^+}{\varsigma} - \left(\frac{\hat{Q}(z;\cdot) - \nu}{\varsigma} - \frac{1}{1-\beta}\right)^+\right) = 1.$$

Let us remark that expression (4.9) means:

$$\forall \omega \in \Omega, \quad \hat{\theta}(\omega) = \begin{cases} 0 & \text{if } \hat{Q}(z;\omega) < \nu, \\ \dfrac{\hat{Q}(z;\omega) - \nu}{\varsigma} & \text{if } \hat{Q}(z;\omega) \in \left[\nu, \nu + \dfrac{\varsigma}{1-\beta}\right], \\ \dfrac{1}{1-\beta} & \text{if } \hat{Q}(z;\omega) > \nu + \dfrac{\varsigma}{1-\beta}. \end{cases} \tag{4.10}$$

Comparing (4.10) to (4.7) gives some insights into the interpretation of the regularisation parameterised by ς; in particular, $\lim_{\varsigma\to 0} \nu = \mathrm{VaR}_\beta(\hat{Q}(z;\cdot))$. Figure 2 gives an example of an optimal measure $\theta^\star$ as defined by (4.7) compared with a possible regularisation $\hat{\theta}$ as defined by (4.10).

4.2.2 Gradient-descent for regularised optimisation problem

Algorithm 8 proposes a straightforward adaptation of algorithm 1 to problem 4 with a stochastic approximation. A sequence of regularisation parameters $\varsigma := (\varsigma_k)_{k\in\mathbb{N}} \subset]0, +\infty[$ converging to zero is set a priori.

Unlike the rest of the report, we only use (single-level) MC estimators here – a consistent MLMC estimation of $\hat{\theta}$ as defined by (4.9) has yet to be established. In particular, line 7 uses the method introduced in § 3.3.2 to build an approximation

[8] These assumptions can be weakened; see ibid.

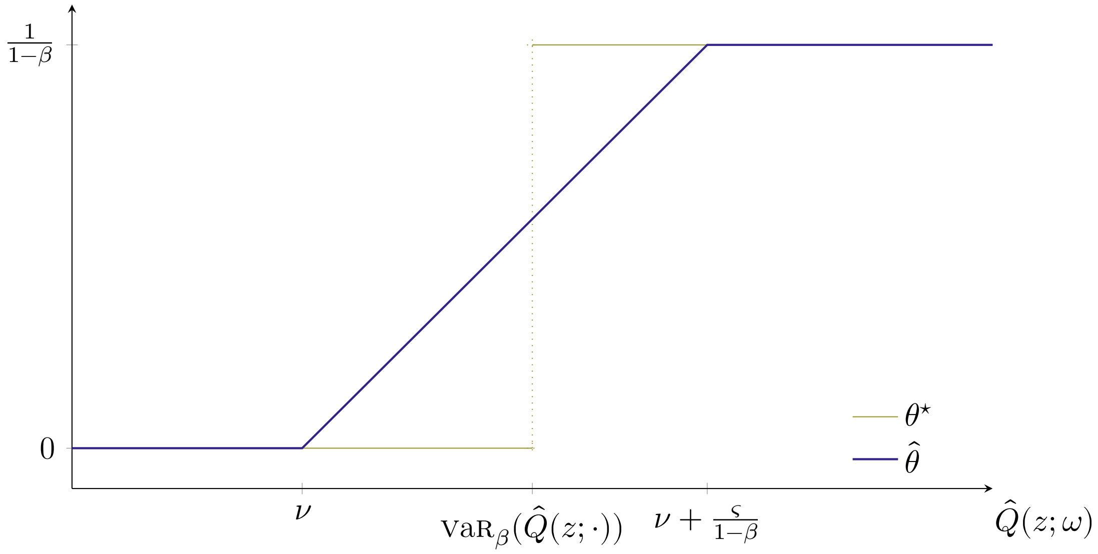

Figure 2: Example of optimal measure $\theta^\star \in \partial\,\text{CVaR}_\beta(\hat{Q}(z;\cdot))$ and regularisation $\hat{\theta}$

of the parametric expectation and estimate the value of the parameter ν. For simplicity, the same number of samples is used on lines 7 and 9. However, it would be more accurate to tune each of these estimators separately, as illustrated in previous algorithms. With regard to cost-efficiency, one may note that the values of $\hat{\theta}$ computed on line 8 are likely to be zero – the larger β is, the more likely it is. For any $\omega \in \Omega$ s.t. $\hat{\theta}(\omega) = 0$, one can dispense with computing $\nabla_1 \hat{Q}(z_k;\omega)$ on line 9. This echoes a comment made on algorithms 5 and 7.

More generally, this method shows resemblance to the primal approach with accurate estimation of the VaR that was presented in 3.3, with a smoothed CVaR. The comparative merits of each remain to be investigated.

Algorithm 8: Gradient-descent with SA-MC for regularised CVaR

```
Input: $z_0$, $\gamma$, $\varsigma$, $\eta$
While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
    Choose $m \in \mathbb{N}$ and draw $\boldsymbol{\omega} \in \Omega^m$
    For $i \in \{1 \dots m\}$ do
        Find $u_i$ s.t. $F(u_i, z_k, \omega_i) = 0$
        Compute $\tilde{q}_i = Q(u(z_k)(\omega_i))$
    Find $\nu \in \mathbb{R}$ s.t. $\mu_m\left(\frac{(\tilde{q}-\nu)^+}{\varsigma_k} - \left(\frac{\tilde{q}-\nu}{\varsigma_k} - \frac{1}{1-\beta}\right)^+\right) = 1$
    Compute $\hat{\theta}_i = \frac{(\tilde{q}_i-\nu)^+}{\varsigma_k} - \left(\frac{\tilde{q}_i-\nu}{\varsigma_k} - \frac{1}{1-\beta}\right)^+$, $\forall i \in \{1 \dots m\}$
    Compute descent direction $\nabla J(z_k) := \mu_m(\hat{\theta}\, \nabla_1 \hat{Q}(z_k; \cdot)) + \nabla P(z_k)$
    Update design $z_{k+1} := \prod_{\mathcal{A}}(z_k - \gamma_k \nabla J(z_k))$
Result: $z_{k+1}$
```

5 Conclusions and future leads

5.1 Summary of key points

This report first presented a general introduction to gradient-based optimisation under uncertainties for a generic coherent, smooth risk measure. An idealised gradient-descent algorithm was proposed, based on optimality conditions from deliverable 6.2. We then described two ways to discretise the probability space in order to make this algorithm practically tractable: sample-average approximation (SAA) and stochastic approximation (SA). Each of these was combined with multi-level Monte Carlo (MLMC) estimators; for SA, we distinguished two possibles combinations: multi-level stochastic gradient (MLSG) and randomised multi-level stochastic gradient (RMLSG). These methods were illustrated with practical, implementable versions of the idealised algorithm. Since the fine tuning of these algorithms is an extensive topic, we merely outlined these aspects and referred to the pertinent literature.

Secondly, we considered a coherent, non-smooth risk measure: the conditional value at risk (CVaR). Since neither the value nor the derivatives of the CVaR are easily accessible, and their estimators are typically biased, the gradient-descent algorithm from the previous section had to be adapted; we proposed two different approaches to this end. The first one is a joint gradient descent over both the design space and a parameter space related to the CVaR; the CVaR itself is never precisely evaluated. The second, on the other hand, uses a MLMC method to estimate accurately the CVaR[9] at every step of the gradient-descent algorithm, which explores only the design space. Stochastic approximation of each approach are detailed, with subgradients due to the lack of regularity of the risk measure.

Finally, we discussed possibilities to regularise the optimisation problem when the CVaR is the risk measure. The first, most detailed possibility is to approximate the CVaR with a smooth function, so as to have a smooth optimisation problem as described in the first part of the report. Suitable smooth approximations were proposed, with a priori error estimators of the smoothing error. The adaptation of previous algorithms to this 'smoothed CVaR' were detailed, including control of the smoothing error. The second possibility is to use a dual characterisation of the CVaR in order to compute its sensitivities. This approach was merely outlined and will be further detailed in a future document; particularly its possible combination with MLMC methods.

[9] As well as the associated value at risk (VaR).

5.2 Future plans and challenges

The methods described in this report can readily be used for 'simple' OUU problems, with a coherent risk measure that is either smooth or the CVaR. Since several alternative methods were discussed on various aspects of the optimisation process, one would have to be chosen. A comparison on a suitable, small problem may provide the necessary insights. Although the optimisation algorithms themselves have yet to be implemented, the MLMC methods used throughout the report are available in the XMC library (Ayoul-Guilmard et al. 2019). This include in particular the adaptive estimation of parametric expectations. Beyond these immediate considerations, several other challenges are expected for the ExaQUte project.

The first of these challenges is to consider time-dependent partial differential equations (PDE). Although the optimal design itself is to be independent of time, this may change the way we tune the algorithm and MLMC estimators. This point will be discussed in the next report (deliverable 6.4). A noteworthy corollary is that solutions of the adjoint problem may not be available. Although this is not an optimisation challenge, it means we may need to find other means to compute the sensitivities of the solution of the PDE with respect to the design parameters. Finally, the quantity of interest (QoI) itself may depend on the design parameters directly (not simply through the solution of the PDE) and non-linearly; e.g. consider the optimisation of a shape, with a QoI featuring an integral over that shape. The gradient of the objective function will therefore feature additional sensitivities, which will have to be estimated.

Beyond these known challenges, planned to be addressed in future reports, there are leads both promising and pertinent to the ExaQUte project that ought to be mentioned here. The first one is the dual approach outlined in § 4.2, as a mean to compute the CVaR and its sensitivity. Headway has already been made on that approach, yet more research is needed to combine it with MLMC estimators, and compare it with the alternative approaches. A second one, also mentioned in this report, is to exploit higher-order derivatives for optimisation, e.g. with Newton-like methods. These methods could significantly accelerate the optimisation. It is one of the motivations for smoothing the CVaR. Finally, a project of high-performance computing such as ExaQUte can benefit greatly from asynchronous algorithms to make the most of parallel computing. The theory exists – see Kushner 2003, chapter 12 – and has been applied to stochastic programming before (e.g. by Woodruff et al. 2018). However, we are not aware of any such work for a problem of optimisation under uncertainties such as the one considered in this report, and deem this to be worth investigating.

References

Armijo, Larry (1st Jan. 1966). 'Minimization of functions having Lipschitz continuous first partial derivatives'. In: *Pacific Journal of Mathematics* 16, pp. 1–3. DOI: 10.2140/pjm.1966.16.1.

Ayoul-Guilmard, Quentin, Sundar Ganesh, Fabio Nobile, Riccardo Tosi, Riccardo Rossi, Ramon Amela and Rosa M Badia and (May 2019). *XMC library*. Comp. software. ExaQUte consortium. DOI: 10.5281/zenodo.3235833.

Bollapragada, Raghu, Richard Byrd and Jorge Nocedal (2018). 'Adaptive Sampling Strategies for Stochastic Optimization'. In: *SIAM Journal on Optimization* 28.4, pp. 3312–3343. DOI: 10.1137/17M1154679.

Byrd, Richard H., Gillian M. Chin, Jorge Nocedal and Yuchen Wu (24th June 2012). 'Sample size selection in optimization methods for machine learning'. In: *Mathematical Programming* 134, pp. 127–155.

Chen, Chunhui and O. L. Mangasarian (1st Nov. 1995). 'Smoothing methods for convex inequalities and linear complementarity problems'. In: *Mathematical Programming* 71 (1), pp. 51–69. ISSN: 1436-4646. DOI: 10.1007/BF01592244.

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

Curi, Sebastian, Kfir. Y. Levy, Stefanie Jegelka and Andreas Krause (28th Oct. 2019). 'Adaptive Sampling for Stochastic Risk-Averse Learning'. In: *arXiv e-prints*. arXiv: 1910.12511 [cs.LG].

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

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

— (2015). 'Multilevel Monte Carlo methods'. In: *Acta Numerica* 24, pp. 259–328. DOI: 10.1017/S096249291500001X.

Kouri, D. and T. Surowiec (Feb. 2016). 'Risk-Averse PDE-Constrained Optimization Using the Conditional Value-At-Risk'. In: *SIAM Journal on Optimization* 26.1, pp. 365–396. DOI: 10.1137/140954556.

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.

Kushner Harold; Yin, George (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. 2nd ed. Vol. 35. Springer-Verlag. ISBN: 978-0-387-21769-7. DOI: 10.1007/b97441.

Martin, Matthieu Claude, Sebastian Krumscheid and Fabio Nobile (Apr. 2018). *Analysis of stochastic gradient methods for PDE-constrained optimal Control*

Problems with uncertain parameters. Tech. rep. 04.2018. MATHICSE. DOI: 10.5075/epfl-MATHICSE-263568.

Martin, Matthieu Claude, Fabio Nobile and Panagiotis Tsilifis (2019). 'A Multilevel Stochastic Gradient method for PDE-constrained Optimal Control Problems with uncertain parameters'. In: *arXiv eprints*. arXiv: 1912.11900 [math.OC].

Pflug, Georg Ch. (2000). 'Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk'. In: *Probabilistic Constrained Optimization: Methodology and Applications*. Ed. by Stanislav P. Uryasev. Boston, MA: Springer US, pp. 272–281. ISBN: 978-1-4757-3150-7. DOI: 10.1007/978-1-4757-3150-7_15.

Rhee, Chang-Han and Peter W. Glynn (2015). 'Unbiased Estimation with Square Root Convergence for SDE Models'. In: *Operations Research* 63.5, pp. 1026–1043. DOI: 10.1287/opre.2015.1404.

Robbins, Herbert and Sutton Monro (Sept. 1951). 'A Stochastic Approximation Method'. In: *Annals of Mathematical Statistics* 22.3, pp. 400–407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Rockafellar, R. Tyrrell and Johannes O. Royset (6th Mar. 2015). 'Engineering Decisions under Risk Averseness'. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 1.2, p. 04015003. DOI: 10.1061/AJRUA6.0000816.

Rockafellar, R. Tyrrell 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.

Shamir, Ohad and Tong Zhang (28th Dec. 2012). 'Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes'. In: *arXiv eprints*. arXiv: 1212.1824 [cs.LG].

Shapiro, Alexander, D. Dentcheva and A. Ruszczyński (2009). *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898718751.

Urbainczyk, Simon (2nd June 2020). 'Adaptive sampling for stochastic optimization with applications in risk-averse engineering design and machine learning'. Technische Universität München.

Woodruff, David L., Jonathan Eckstein and Jean-Paul Watson (30th Oct. 2018). 'Asynchronous Projective Hedging for Stochastic Programming'. URL: www.optimization-online.org/DB_HTML/2018/10/6895.html.

A Calculation of subdifferentials

Let $(X, t) \in \mathrm{L}^r(\Omega, \mathbb{R}) \times \mathbb{R}$. Let us express the subdifferentials of $R(X, t) := t + \frac{1}{1-\beta}\mathbb{E}((X-t)^+)$ with respect to either variable, as per definition 5.

First subdifferential

Let $Y \in \mathrm{L}^{r'}(\Omega, \mathbb{R})$, with $r' := (1 - r^{-1})^{-1}$.

$$\begin{aligned}
&\lim_{\epsilon\to 0} \mathbb{E}\left(\frac{(X+\epsilon Y - t)^+ - (X-t)^+}{\epsilon}\right) \\
&= \lim_{\epsilon\to 0} \mathbb{E}(Y\mathbb{1}(X \geqslant t, X+\epsilon Y \geqslant t)) + \mathbb{E}\left(\frac{X+\epsilon Y - t}{\epsilon}\mathbb{1}(X < t, X + \epsilon Y \geqslant t)\right) \\
&\qquad - \mathbb{E}\left(\frac{X-t}{\epsilon}\mathbb{1}(X \geqslant t, X+\epsilon Y > t)\right).
\end{aligned}$$

We see that the last two terms vanish by dominated convergence. Let us pass the limit inside the remaining expectation.

For the left side:

$$\begin{aligned}
\lim_{\epsilon\to 0^-} \mathbb{E}\left(\frac{(X+\epsilon Y - t)^+ - (X-t)^+}{\epsilon}\right) &= \mathbb{E}\Big(\lim_{\epsilon\to 0^-} Y\mathbb{1}(X \geqslant t, X+\epsilon Y \geqslant t)\Big) \\
&= \mathbb{E}(Y\mathbb{1}(X > t)\mathbb{1}(Y \geqslant 0)) + \mathbb{E}(Y\mathbb{1}(X \geqslant t)\mathbb{1}(Y < 0)) \\
&= \mathbb{E}(Y\mathbb{1}(X > t)) + \mathbb{E}(Y\mathbb{1}(X = t)\mathbb{1}(Y < 0)) \\
&= \mathbb{E}(Y\mathbb{1}(X \geqslant t)) - \mathbb{E}(Y\mathbb{1}(X = t)\mathbb{1}(Y \geqslant 0)).
\end{aligned} \tag{A.1}$$

For the right side:

$$\begin{aligned}
\lim_{\epsilon\to 0^+} \mathbb{E}\left(\frac{(X+\epsilon Y - t)^+ - (X-t)^+}{\epsilon}\right) &= \mathbb{E}\Big(\lim_{\epsilon\to 0^+} Y\mathbb{1}(X \geqslant t, X+\epsilon Y \geqslant t)\Big) \\
&= \mathbb{E}(Y\mathbb{1}(X \geqslant t)\mathbb{1}(Y \geqslant 0)) + \mathbb{E}(Y\mathbb{1}(X > t)\mathbb{1}(Y < 0)) \\
&= \mathbb{E}(Y\mathbb{1}(X > t)) + \mathbb{E}(Y\mathbb{1}(X = t)\mathbb{1}(Y \geqslant 0)) \\
&= \mathbb{E}(Y\mathbb{1}(X \geqslant t)) - \mathbb{E}(Y\mathbb{1}(X = t)\mathbb{1}(Y < 0))
\end{aligned} \tag{A.2}$$

From (A.1) and (A.2) we conclude that, for any $\alpha \in [0,1]$, the quantity

$$\frac{1}{1-\beta}\mathbb{E}(Y(\mathbb{1}(X>t)+\alpha\mathbb{1}(X=t))) = \left\langle Y, \frac{1}{1-\beta}\mathbb{1}(X>t)+\alpha\mathbb{1}(X=t)\right\rangle_{\mathrm{L}^{r'},\mathrm{L}^{r}} \tag{A.3}$$

is a subderivative of R at (X,t) with respect to the first variable in the direction Y. Reciprocally, every such subderivative can be expressed in the form (A.3). Therefore, the first subdifferential is

$$\partial_1 R(X,t) = \left\{\frac{1}{1-\beta}\mathbb{1}(X>t)+\alpha\mathbb{1}(X=t) : \alpha\in[0,1]\right\}.$$

Second subdifferential

Let us consider the right-side derivative first:

$$\begin{aligned}\lim_{\epsilon\to0^+}&\mathbb{E}\left(\frac{(X-t-\epsilon)^+-(X-t)^+}{\epsilon}\right)\\ &= \lim_{\epsilon\to0^+}\underbrace{-\mathbb{E}(\mathbb{1}(X\leqslant t+\epsilon))}_{T_1}\underbrace{-\mathbb{E}\left(\frac{X-t}{\epsilon}\mathbb{1}(t\leqslant X<t+\epsilon)\right)}_{T_2}.\end{aligned}$$

Regarding T_1, the sequence $\mathbb{1}(X\geqslant t+\epsilon)$ is dominated by 1. By passing to the limit we have

$$\lim_{\epsilon\to0^+}-\mathbb{E}(\mathbb{1}(X\geqslant t+\epsilon)) = -\mathbb{E}\left(\lim_{\epsilon\to0^+}\mathbb{1}(X\geqslant t+\epsilon)\right) = -\mathbb{E}(\mathbb{1}(X>t)). \tag{A.4}$$

Concerning T_2, the sequence $(X-t)\mathbb{1}(t\leqslant X<t+\epsilon)/\epsilon$ is dominated by 1 as well. Let $\omega\in\Omega$: either

(i) $X(\omega)\neq t$, and therefore $\mathbb{1}(t\leqslant X<t+\epsilon)=0$ for ϵ small enough; or
(ii) $X(\omega)=t$ and so $(X(\omega)-t)=0$.

Therefore

$$\lim_{\epsilon\to0^+}\mathbb{E}\left(-\frac{X-t}{\epsilon}\mathbb{1}(t\leqslant X<t+\epsilon)\right) = \mathbb{E}\left(\lim_{\epsilon\to0^+}-\frac{X-t}{\epsilon}\mathbb{1}(t\leqslant X<t+\epsilon)\right) = 0 \tag{A.5}$$

Putting (A.4) and (A.5) together, we conclude

$$\lim_{\epsilon\to0^+}\mathbb{E}\left(\frac{(X-t-\epsilon)^+-(X-t)^+}{\epsilon}\right) = -\mathbb{E}(\mathbb{1}(X>t)) \tag{A.6}$$

Let us now consider the left-side derivative.

$$\begin{aligned}\lim_{\epsilon\to0^-}&\mathbb{E}\left(-\frac{X-t}{\epsilon}\mathbb{1}(t\leqslant X<t+\epsilon)\right)\\ &= \lim_{\epsilon\to0^-}-\mathbb{E}(\mathbb{1}(X\geqslant t))+\underbrace{\mathbb{E}\left(\frac{X-t-\epsilon}{\epsilon}\mathbb{1}(t-\epsilon\leqslant X<t)\right)}_{T_3}\end{aligned}$$

Regarding T_3, the sequence $f_\epsilon = \frac{X-t-\epsilon}{\epsilon}\mathbb{1}(t-\epsilon \leqslant X < t)$ is dominated by 1 and pointwise, $\forall\omega \in \Omega$, $\lim_{\epsilon\to 0^-} f_\epsilon(\omega) = 0$. Therefore

$$\lim_{\epsilon\to 0^-} \mathbb{E}\left(-\frac{X-t}{\epsilon}\mathbb{1}(t \leqslant X < t+\epsilon)\right) = -\,\mathbb{E}(\mathbb{1}(X \geqslant t)). \tag{A.7}$$

From (A.6) and (A.7), we conclude that the second subdifferential is

$$\partial_2 R(X,t) = \{1 - \mathbb{E}(\mathbb{1}(X > t)) - \alpha\,\mathbb{E}(\mathbb{1}(X = t)) : \alpha \in [0,1]\}.$$

B Example of a MLSG algorithm for a smoothed CVaR

Let us illustrate a few of the methods described in this report, on a specific problem of OUU. We apply a SA-MLSG algorithm with accurate VaR estimation (i.e. algorithm 7) for a smoothed CVaR, on problem 5.

Problem 5 (Applied example)**.** Let $D \subset \mathbb{R}^3$, $a \in \mathrm{L}^2(\Omega, \mathrm{H}^1(D, \mathbb{R}^3))$, $\bar{z} \in \mathrm{H}^1(D, \mathbb{R})$, $b, q \in \mathrm{L}^2(D, \mathbb{R})$ and $\beta \in]0,1[$. Find the optimal design

$$z^\star := \operatorname{argmin}\left\{ \mathrm{CVaR}_\beta\left(\int_D qu\right) + \frac{1}{2}\int_D (z-\bar{z})^2 : z \in \mathrm{H}^1(D, \mathbb{R});\ u \in \mathrm{H}^1_0(D, \mathbb{R}) \text{ s.t. (B.1)} \right\}.$$

constrained by the PDE

$$\nabla \cdot (a(\omega)u - \mathrm{e}^z \nabla u) = b, \text{ for } \mathbb{P}\text{-a.e. } \omega \in \Omega. \tag{B.1}$$

We remark that problem 5 is a particular case of problem 1 with $Z := \mathrm{H}^1(D, \mathbb{R})$, $U := \mathrm{H}^1_0(D, \mathbb{R})$,

$$\mathcal{R} := \mathrm{CVaR}_\beta, \qquad P(z) := \frac{1}{2}\int_D (z-\bar{z})^2, \quad \text{and} \quad F(u,z,\omega) := \nabla \cdot (a(\omega)u - \mathrm{e}^z \nabla u) - b.$$

Since the risk measure is a CVaR, this problem can be reformulated as a minimisation problem over $Z \times \mathbb{R}$ in exactly the same way that problem 3 was; we proceed under the same assumptions. Additionally, we follow the regularisation approach described in § 4.1, i.e. we substitute to CVaR_β a smoothed approximation $\mathrm{CVaR}_{\beta,\varsigma}$ defined as in (4.3) for any $\varsigma \in]0, +\infty[$. The choice of the smoothing function $(\cdot)^+_\varsigma$ is left undetermined, provided that the assumptions formulated in § 4.1.1 are satisfied. Let us recall that its derivative is denoted s_ς. The resulting of OUU problem can be solved using the gradient-descent method with accurate VaR estimation proposed in § 3.3.

For the practical implementation, the probability measure is discretised following a SA strategy to which the gradient-descent algorithm is adapted using the MLSG approach described in § 3.3.2. We assume that we can define for U a sequence $(U_l)_{l\in\mathbb{N}}$ of nested, finite-dimensional approximation spaces based on increasingly-fine discretisations $(D_l)_{l\in\mathbb{N}}$ of D, suitable for MLMC estimations (see § 2.3.1); likewise for $L^2(D, \mathbb{R})$. We also choose a finite dimensional approximation $\mathcal{A}$ of the design space Z based on a discretisation $D_{-1} \subset D_0$ of the spatial domain – i.e. the chain of approximation reads $\mathcal{A} \subset \mathrm{H}^1(D_{-1}) \subset \mathrm{H}^1(D_0) \subset Z$.

An illustration of this method is proposed in algorithm 9. This is largely a particularisation of 7 to problem 5, with the modifications detailed in § 4.1 for the regularisation. On lines 16–17 appears the adjoint equation: for a given event $\omega \in \Omega$ and design $z \in Z$, the adjoint $\lambda \in \mathrm{H}^1_0(D, \mathbb{R})$ of a solution $u \in U$ of (B.1) satisfies the adjoint equation

$$-a(\omega) \nabla \lambda - \nabla \cdot (\mathrm{e}^z \nabla \lambda) = \frac{q}{1-\beta} s_\varsigma\left(\int_D qu - \mathrm{VaR}_\beta\left(\int_D qu \right) \right).$$

On line 18 is an estimation of the gradient of the objective functional

$$\nabla J(z) = (z - \bar{z}) + \mathbb{E}(\mathrm{e}^z \nabla u \nabla \lambda).$$

As for algorithm 7, the tolerances η, $\epsilon := (\epsilon_k)_{k\in\mathbb{N}}$ and $\epsilon' := (\epsilon'_k)_{k\in\mathbb{N}}$ are set a priori, whereas the choice of step sizes $\gamma := (\gamma_k)_{k\in\mathbb{N}}$ is left undetermined. Let us recall that tuning the estimator $\mu_{\boldsymbol{m}'}$ for accuracy is not necessary, and that a typical stochastic-gradient strategy would keep the estimator inexpensive albeit inaccurate (see similar remark for algorithm 7).

Algorithm 9: Applied example of algorithm 7 with smoothed cVaR

```
1  Input: $z_0$, $\gamma$, $\epsilon$, $\epsilon'$, $\eta$
2  While $\|z_{k+1} - z_k\| \geqslant \eta \|z_k\|$ do
3      Choose $\varsigma \in ]0, +\infty[$, $(L', n) \in \mathbb{N}^2$ and $(\boldsymbol{m}', \boldsymbol{r}) \in \mathbb{N}^{L'+1} \times \mathbb{R}^n$ s.t.
         $\mathrm{mse}\big(\operatorname{argmin} \Phi_{\boldsymbol{m}',\boldsymbol{r},\varsigma}(Q(u))\big) \leqslant \epsilon'_k$
4      Draw $\boldsymbol{\omega} \in \Omega^{\boldsymbol{m}'}$
5      For $l \in \{0 \dots L'\}$ do
6          For $i \in \{1 \dots m'_l\}$ do
7              Find $u_l(\omega_{l,i})$ s.t. $\nabla \cdot (a_l(\omega_{l,i}) u_l(\omega_{l,i}) - \mathrm{e}^{z_k} \nabla u_l(\omega_{l,i})) = b_l$
8              Find $u_{l-1}(\omega_{l,i})$ s.t.
                 $\nabla \cdot (a_{l-1}(\omega_{l,i}) u_{l-1}(\omega_{l,i}) - \mathrm{e}^{z_k} \nabla u_{l-1}(\omega_{l,i})) = b_{l-1}$
9      Estimate VaR as $t := \operatorname{argmin} \Phi_{\boldsymbol{m}',\boldsymbol{r},\varsigma}(Q(u))$
10     Choose $L \in \mathbb{N}$ and $\boldsymbol{m} \in \mathbb{N}^{L+1}$ s.t. $\mathrm{mse}(\mu_{\boldsymbol{m}}(\mathrm{e}^{z} \nabla u \nabla \lambda)) \leqslant \epsilon_k$
11     For $l \in \{0 \dots L\}$ do
12         For $i \in \{1 \dots m_l\}$ do
13             If $i > m'_l$ then
14                 Draw $\omega_{l,i} \in \Omega$
15                 Compute $u_l(\omega_{l,i})$ and $u_{l-1}(\omega_{l,i})$ as in lines 7–8.
16             Find $\lambda_l(\omega_{l,i})$ s.t. $-a_l(\omega_{l,i}) \nabla \lambda_l(\omega_{l,i}) - \nabla \cdot (\mathrm{e}^{z_k} \nabla \lambda_l(\omega_{l,i})) =$
                 $\frac{q_l}{1-\beta} s_\varsigma\big(\int_D q_l u_l(\omega_{l,i}) - t\big)$
17             Find $\lambda_{l-1}(\omega_{l,i})$ s.t.
                 $-a_{l-1}(\omega_{l,i}) \nabla \lambda_{l-1}(\omega_{l,i}) - \nabla \cdot \big(\mathrm{e}^{z_k} \nabla \lambda_{l-1}(\omega_{l,i})\big) =$
                 $\frac{q_{l-1}}{1-\beta} s_\varsigma\big(\int_D q_{l-1} u_l(\omega_{l,i}) - t\big)$
18     Compute descent direction $\nabla J_{\boldsymbol{m}}(z_k) := (z_k - \bar{z}) + \mu_{\boldsymbol{m}}(\mathrm{e}^{z_k} \nabla u \nabla \lambda)$
19     Set new design $z_{k+1} := z_k - \gamma_k \nabla J_{\boldsymbol{m}}(z_k)$
20 Result: $z_{k+1}$
```

List of Algorithms

Acronyms

BFGS	Broyden–Fletcher–Goldfarb–Shanno
CDF	cumulative distribution function
CVaR	conditional value at risk
HPC	high-performance computing
MC	Monte Carlo
MLMC	multi-level Monte Carlo
MLSG	multi-level stochastic gradient
MSE	mean squared error
OUU	optimisation under uncertainties
PDE	partial differential equations
PDF	probability density function
QoI	quantity of interest
RMLSG	randomised multi-level stochastic gradient
SA	stochastic approximation
SAA	sample-average approximation
SG	stochastic gradient
VaR	value at risk

Abbreviations

a.e.	almost every, almost everywhere
a.s.	almost surely
cf.	confer
e.g.	exempli gratia
et al.	et alii
ibid.	ibidem
i.e.	id est
iff.	if and only if
i.i.d.	independent and identically distributed
s.t.	such that