Efficient evaluation of reliability-oriented sensitivity indices

Gilles Defaux, Guillaume Perrin

Efficient evaluation of reliability-oriented sensitivity indices

G. Perrin[a], G. Defaux[a]

[a]*CEA/DAM/DIF, F-91297, Arpajon, France*

Abstract

The role of simulation keeps increasing for the reliability analysis of complex systems. Most of the time, these analyses can be reduced to estimating the probability of occurrence of an undesirable event, also called failure probability, using a stochastic model of the system. If the considered event is rare, sophisticated sample-based procedures are generally introduced to get a relevant estimate of the failure probability. Based on the samples constructed for the evaluation of this estimate, this work defines two types of reliability-oriented sensitivity indices. The first ones are introduced to identify the model inputs whose variability has to be reduced in priority to decrease this probability. The second ones are used to find the model inputs whose distribution has to be particularly well-characterized for the available estimate to be realistic. It is also shown how these sensitivity indices can be derived when the true model is approximated by a surrogate model. In particular, an innovative procedure is proposed to take into account the surrogate model uncertainty in the estimation of these sensitivity indices. The proposed approach is then applied to the reliability analysis of a series of numerical and industrial examples.

Keywords:
Sobol indices, Gaussian process, sensitivity analysis, risk analysis.

1. Introduction

The reliability analysis of complex systems is more and more quantified using numerical simulations. Hence, computer codes in the form $y(\boldsymbol{x}) =$

$g(\boldsymbol{x}; \boldsymbol{d})$ are generally involved. Here, $\boldsymbol{x} = (x_1, \dots, x_D) \in \mathbb{X}$ is the vector of stochastic inputs and $\boldsymbol{d}$ is the vector of deterministic inputs. Formally, given an adapted threshold q, the problem of estimating the probability of occurrence of an undesirable event, seen as $y(\boldsymbol{x})$ exceeding q, can be reduced to computing the probability

$$p := \mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q) = \mathbb{E}_{\boldsymbol{x}} \left[1_{y(\boldsymbol{x}) > q} \right], \tag{1}$$

$$1_{y(\boldsymbol{x}) > q} = \begin{cases} 1 \text{ if } y(\boldsymbol{x}) > q, \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

While D increases and low values of p are considered, the evaluation of p cannot be handled with usual quadratures. Sampling techniques are preferred, such as the Monte Carlo Simulation (MCS) [34]. In MCS, the code is computed in a large number of inputs, and probability p is estimated by counting the number of responses that are above the threshold q. However, the square value of the coefficient of variation of the estimator provided by MCS is proportional to $1/p$. Hence, the number of code evaluations required for MCS to estimate small values of p (say $p < 10^{-3}$ for the considered applications) quickly becomes burdensome. To circumvent this problem, various approaches have been proposed. On the one hand, several nonstatistical approaches, such as the first-order or second-order reliability methods (FORM/SORM) [14, 31, 18, 5], propose to approximate the limit state function as a parametric function. Then, these approximations are used to evaluate p at a low computational cost, but at the expense of a reduced precision. On the other hand, the splitting methods [16] propose to rewrite p using a finite sequence of increasing thresholds. Depending on the choice of the thresholds, the variance of the aggregated estimator can be much smaller than the one given by MCS, as it will be explained in the next section.

In addition to this estimation of p, it is useful to quantify the importance of each model input on the failure probability. This is the purpose of reliability-oriented global sensitivity analysis (ro-GSA). Here, the word *global* refers to the definition given by [36], as the whole variation domain of the inputs is considered. Indeed, if one model input happens to be strongly influential, it could be worth trying to decrease its variability. On the contrary, if one model input seems to have no influence on p, its variability can be neglected, resulting in a simpler model. Several methods have been proposed

to carry out such sensitivity analyses. A first approach aims at quantifying how the probability of exceeding q is affected by fixing one input model to a given value. It can be shown that this is equivalent to computing the Sobol indices associated with the indicator function $1_{y(\boldsymbol{x})>q}$ (see Section 3). However, dedicated evaluations of the code are generally required to assess such indices [40], which can be much more numerous that the ones required to get a good estimate of p. Hence, the first objective of this work is to propose a method using nonparametric statistics to evaluate such Sobol indices without additional code evaluations.

The sensitivity of probability p to each model input can also be evaluated by comparing the partial derivatives of p with respect to the statistical moments of each model input. Using adapted strategies, such derivatives can be computed as a simple post-processing of the code evaluations that were carried out for the estimation of p [33]. Whereas Sobol indices are associated with a fixed distribution for the inputs, such indicators assess the sensitivity of p to small changes of the input distribution. Hence, the meanings of these two sets of indicators are different but complementary. On the one hand, the Sobol indices indicate the model inputs whose variability has to be reduced in priority if we want to decrease p. On the other hand, the derivative-based indices show the model inputs whose statistical moments have to be particularly well-controlled to get a relevant estimation of p. In that prospect, generalizing the works achieved in [19], the second objective of this work is to propose reliability-oriented indices that can consider more general modifications of the model inputs distributions, and which can be used to evaluate cross-effects between model inputs. Such indices will also be computed as a simple post-processing of the simulations used to estimate p.

When the numerical cost associated with one evaluation of the code is high (between several minutes to several days CPU), surrogate models are commonly introduced to emulate the time-demanding computer code for the estimation of p. Among these methods, the Gaussian process regression (GPR) method, or kriging, plays a major role. This is mostly due to its ability to provide an uncertainty on the evaluation of p that is due to the substitution of the true code by its emulator [35, 37]. Finally, this paper shows how to derive each reliability-oriented sensitivity index in the case when the code is replaced by a Gaussian emulator. In particular, the impact of the emulator uncertainty on the estimation of these indices is quantified.

The outline of this work is as follows. First, Section 2 briefly reviews existing sample-based methods for estimating p. Section 3 introduces reliability-

3

oriented sensitivity indices to compare the influence of the inputs variability on p. Another type of sensitivity indices is defined in Section 4, in order to quantify the robustness of the evaluation of p to small perturbations of the input distribution. Then, Section 5 introduces the estimation of these indices when the computer code is replaced by a Gaussian emulator. At last, a series of examples are shown in Section 6 to illustrate the interest of the proposed methods.

Notations

The following notations are adopted:

- x, y correspond to scalars.

- X, Y correspond to integers.

- $\boldsymbol{x}, \boldsymbol{y}$ correspond to vectors.

- Let x_i be the components of a vector $\boldsymbol{x}$.

- For all D-dimensional vector $\boldsymbol{x} = (x_1, \ldots, x_D)$, we denote by $\boldsymbol{x}_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_D)$ the vector that gathers all the components of $\boldsymbol{x}$ but the ith.

- For all random vector $\boldsymbol{x}$, $\mathbb{E}_{\boldsymbol{x}}[\cdot]$ and $\mathbb{V}_{\boldsymbol{x}}[\cdot]$ denote the mathematical expectation and the variance operator associated with the distribution of $\boldsymbol{x}$.

2. Background : sample-based methods to estimate probabilities of exceeding thresholds

Let $\mathcal{S}$ be the system we are interested in, whose properties (dimensions, boundary conditions, material properties...) can be characterized by a vector of $D \geq 1$ parameters $\boldsymbol{x} \in \mathbb{X}$, where $\mathbb{X}$ is a subset of $\mathbb{R}^D$. Vector $\boldsymbol{x}$ is modelled by a random vector to take into account the fact that these parameters are not perfectly known. The components of $\boldsymbol{x}$ are assumed to be statistically independent. For all $1 \leq i \leq D$, let $\mathbb{X}_i$, f_{x_i} and F_{x_i} be the definition domain, the probability density function (PDF) and the cumulative density function (CDF) of component x_i respectively. It follows that

$$\mathbb{X} = \bigtimes_{i=1}^{D} \mathbb{X}_i, \quad f_{\boldsymbol{x}}(\boldsymbol{x}) = \prod_{i=1}^{D} f_{x_i}(x_i), \quad F_{\boldsymbol{x}}(\boldsymbol{x}) = \prod_{i=1}^{D} F_{x_i}(x_i), \tag{3}$$

where $f_{\boldsymbol{x}}$ and $F_{\boldsymbol{x}}$ are the PDF and the CDF of $\boldsymbol{x}$ respectively. In addition, let

$$y : \begin{cases} \mathbb{X} & \to \ \mathbb{R} \\ \boldsymbol{x} & \mapsto y(\boldsymbol{x}) \end{cases} \tag{4}$$

be the real-valued deterministic mapping describing the behaviour of $\mathcal{S}$. In this work, we are interested in the evaluation of the probability p for $y(\boldsymbol{x})$ to exceed a given threshold $q \in \mathbb{R}$,

$$p := \mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q) = \int_{\mathbb{X}} \mathbb{1}_{y(\boldsymbol{x}) > q} f_{\boldsymbol{x}}(\boldsymbol{x}) d\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}} \left[\mathbb{1}_{y(\boldsymbol{x}) > q} \right], \tag{5}$$

but also in the identification of the components of $\boldsymbol{x}$ that play the most important roles on this probability. We moreover assume that the computational cost associated with one evaluation of y is high (between several minutes to several hours CPU), so that the number of code evaluations is supposed to be bounded (less than 10^3 for instance). In that context, we are particularly interested by methods that could allow all the computational budget to be used at the same time for the estimation of p and for the sensitivity analysis.

As the model inputs are assumed independent, they all can be considered as normally distributed, centred and of variance equal to 1 without loss of generality. Indeed, an isoprobabilist transform can been applied to each model input [32, 24, 17], impacting neither the definition of p nor the results of the sensitivity analysis. Therefore, in the following,

$$f_i(x_i) = \varphi(x_i; 0, 1), \quad x_i \in \mathbb{X}_i = \mathbb{R}, \tag{6}$$

where for all (μ, σ) in $\mathbb{R} \times \mathbb{R}^{+*}$,

$$\varphi(x_i; \mu, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right). \tag{7}$$

The most famous sample-based method to estimate p is the MCS. If $\boldsymbol{x}_n$, $1 \leq n \leq N$, denote N independent copies of $\boldsymbol{x}$, it is well known [34] that

$$p_{\mathrm{MC}} := \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{y(\boldsymbol{x}_n) > q} \tag{8}$$

defines an unbiased estimator of p. The associated coefficient of variation verifies

$$\delta_{\mathrm{MC}}^2 = \frac{1-p}{Np}. \tag{9}$$

This approach is particularly easy to implement, but requires a lot of code evaluations to get acceptable values for δ_{MC}. Alternatively, the splitting methods rewrite p using a finite sequence of increasing thresholds $(q_k)_{k=0}^{K}$,

$$p = \mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q_K | y(\boldsymbol{x}) > q_{K-1}) \times \cdots \times \mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q_1 | y(\boldsymbol{x}) > q_0) \times \mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q_0), \tag{10}$$

with $q_0 = -\infty$ and $q_K = q$. Then, classical Monte Carlo estimators can be proposed for each conditional probability. All these estimators being unbiased and independent, the mean of their product is still equal to p. However, the variance of the aggregated estimator strongly depends on the choice of the thresholds. In practice, the sequence of thresholds is defined *on the fly*, which is generally referred as Adaptive splitting [7]. In particular, the Markov chain

$$y_k := (y(\boldsymbol{x}) \mid y(\boldsymbol{x}) > y_{k-1}), \ y_0 = -\infty, \ k \geq 1, \tag{11}$$

is called an increasing random walk. And it can be shown that the counting random variable of the number of events before q, which is denoted by $M :=$ Card $\{k \geq 1 \mid y_k \leq q\}$, follows a Poisson law with parameter $-\log(p)$ [41]. Hence, given $Q \geq 1$ independent random counting variables $(M_q)_{1 \leq q \leq Q}$,

$$p_{\mathrm{MP}} := \left(1 - \frac{1}{Q}\right)^{\sum_{q=1}^{Q} M_q} \tag{12}$$

also defines an unbiased estimator of p, whose coefficient of variation is equal to $\sqrt{-\log(p)/Q}$. This approach is referred as Moving Particle (MP) method in the following. Another strategy to choose the different thresholds is given by the Subset Simulation (SS) method. The interested reader may refer to [1, 8] for further details about this approach.

The main difficulty in the splitting methods is the conditional sampling. Indeed, if we focus on the Markov chain defined by Eq. (11), y_k has to be randomly generated conditionally greater than y_{k-1}. This can be done using the Metropolis-Hastings algorithm [22, 15]. If T is the number of steps that is used to control the convergence of the Markov chain to its stationary distribution, it follows that, on average, $1 - T\log(p)$ samples have to be generated to get one realization of the counting variable M_q. Thus, the mean total number of code evaluations to get the Q samples $M_1, \ldots, M_Q$ is equal to $N = Q(1 - T\log(p))$. It follows that the coefficient of variation of δ_{MP} can be approximated as

$$\delta_{\mathrm{MP}}^2 \approx \frac{\log(p)(T\log(p) - 1)}{N} \approx \frac{T\log(p)^2}{N}. \tag{13}$$

This has to be compared to $\delta_{\mathrm{MC}}^2 \approx 1/(pN)$ for the crude Monte Carlo.

In MCS, SS and MP methods, let $\widehat{p}$ denote the best estimate of p we get once the maximal computational budget is attained. For each of these methods, it can be noticed that the points $\boldsymbol{x}^{(k)}$, where it was observed that $y(\boldsymbol{x}^{(k)})$ is greater than q, are independent realizations of the conditioned random vector $(\boldsymbol{x}|y(\boldsymbol{x}) > q)$. Let us gather all these realisations of $(\boldsymbol{x}|y(\boldsymbol{x}) > q)$ in the set $\mathcal{D}_f := \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N^*)}\right\}$. Hence, in the following, N^* denotes the number of points that have been sampled in the failure domain.

3. Compared influence of the inputs variability on p

Based on the estimated value of p and the elements of $\mathcal{D}_f$ only, the purpose of this section is to identify the components of $\boldsymbol{x}$, whose variability has to be reduced in priority if we want to decrease the value of p. To this end, it can be interesting to quantify the effect on p due to the fact that x_i is fixed to the particular value $x_i^\star$. Indeed, the higher $\left(\mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q) - \mathbb{P}_{\boldsymbol{x}_{-i}}(y(\boldsymbol{x}) > q \mid x_i = x_i^\star)\right)^2$, the more influential x_i. Thus, averaging over x_i, the quantity

$$\mathbb{E}_{x_i}\left[\left(\mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x}) > q) - \mathbb{P}_{\boldsymbol{x}_{-i}}(y(\boldsymbol{x}) > q \mid x_i)\right)^2\right] = \mathbb{V}_{x_i}\left[\mathbb{E}_{\boldsymbol{x}_{-i}}\left[1_{y(\boldsymbol{x})>q} \mid x_i\right]\right] \tag{14}$$

can be used to analyse the sensitivity of p to model input x_i. Normalizing these quantities by $\mathbb{V}_{\boldsymbol{x}}\left[1_{y(\boldsymbol{x})>s}\right]$, we find back the well-known first order Sobol indices [38] associated with function $1_{y(\boldsymbol{x})>q}$:

$$s_i := \frac{\mathbb{V}_{x_i}\left[\mathbb{E}_{\boldsymbol{x}_{-i}}\left[1_{y(\boldsymbol{x})>q} \mid x_i\right]\right]}{\mathbb{V}_{\boldsymbol{x}}\left[1_{y(\boldsymbol{x})>q}\right]}. \tag{15}$$

By construction, index s_i indicates the variance of $1_{y(\boldsymbol{x})>q}$ caused by x_i individually. The variance of $1_{y(\boldsymbol{x})>q}$ caused by x_i including interactions with the components of $\boldsymbol{x}_{-i}$ is given by the ith total Sobol index, denoted by t_i, which verifies:

$$t_i := 1 - \frac{\mathbb{V}_{\boldsymbol{x}_{-i}}\left[\mathbb{E}_{x_i}\left[1_{y(\boldsymbol{x})>q} \mid \boldsymbol{x}_{-i}\right]\right]}{\mathbb{V}_{\boldsymbol{x}}\left[1_{y(\boldsymbol{x})>q}\right]}. \tag{16}$$

Based on Eqs. (15) and (16), the computation of s_i and t_i is nontrivial, since $\mathbb{E}_{\boldsymbol{x}_{-i}}\left[\cdot\right]$ and $\mathbb{V}_{\boldsymbol{x}_{-i}}\left[\cdot\right]$ refer to multidimensional integrals. This motivated the introduction of various algorithms to reduce the computational cost of the Sobol' indices. In particular, efficient sample-based methods can be found in [40, 39, 25, 20] to replace the naive and very expensive double-loop MCS. However, in spite of these developments, the number of dedicated code evaluations that are needed for these methods is still very high. To circumvent this problem, another approach is proposed in this paper, which is based on the Proposition 1, whose proof has been moved to Appendix.

Proposition 1. *For all $1 \leq i \leq D$, we have:*

$$s_i = \frac{p}{1-p}\mathbb{V}_{x_i}\left[\frac{f_{x_i|y(\boldsymbol{x})>q}(x_i)}{f_{x_i}(x_i)}\right], \tag{17}$$

$$t_i = 1 - \frac{p}{1-p}\mathbb{V}_{\boldsymbol{x}_{-i}}\left[\frac{f_{\boldsymbol{x}_{-i}|y(\boldsymbol{x})>q}(\boldsymbol{x}_{-i})}{f_{\boldsymbol{x}_{-i}}(\boldsymbol{x}_{-i})}\right], \tag{18}$$

where, for all $\boldsymbol{x}$ in $\mathbb{X}$,

$$\begin{cases} f_{\boldsymbol{x}|y(\boldsymbol{x})>q}(\boldsymbol{x}) := \dfrac{1}{p}1_{y(\boldsymbol{x})>q}f_{\boldsymbol{x}}(\boldsymbol{x}), \\[2ex] f_{x_i|y(\boldsymbol{x})>q}(x_i) := \displaystyle\int_{\times_{1\leq j\leq D,\ j\neq i}\mathbb{X}_j} f_{\boldsymbol{x}|y(\boldsymbol{x})>q}(\boldsymbol{x}) \prod_{1\leq j\leq D,\ j\neq i} dx_j, \\[2ex] f_{\boldsymbol{x}_{-i}|y(\boldsymbol{x})>q}(\boldsymbol{x}_{-i}) := \displaystyle\int_{\mathbb{X}_i} f_{\boldsymbol{x}|y(\boldsymbol{x})>q}(\boldsymbol{x})dx_i. \end{cases} \tag{19}$$

By construction, the first order and total Sobol indices associated with the indicator function are proportional to the variances of the ratios between the *a priori* PDFs of x_i and $\boldsymbol{x}_{-i}$ and their PDFs conditioned by the fact that $y(\boldsymbol{x})$ is greater than q. In this work, we propose to approximate these PDFs using one of the nonparametric approaches described in [30, 29]. These methods are particularly suited for this kind of approximations, as the construction they propose only requires the presence of independent realizations of the random vector to be modelled. For each $1 \leq i \leq D$, let $\widehat{f}_{x_i|y(\boldsymbol{x})>q}$ and $\widehat{f}_{\boldsymbol{x}_{-i}|y(\boldsymbol{x})>q}$ be these approximations of functions $f_{x_i|y(\boldsymbol{x})>q}$ and $f_{\boldsymbol{x}_{-i}|y(\boldsymbol{x})>q}$ based on the elements of the set $\mathcal{D}_f$ only. Sobol indices s_i and t_i can then be approximated as:

$$ s_i \;\approx\; \widehat{s}_i := \frac{\widehat{p}}{1-\widehat{p}}\mathbb{V}_{x_i}\left[\frac{\widehat{f}_{x_i|y(\boldsymbol{x})>q}(x_i)}{f_{x_i}(x_i)}\right], \tag{20}$$

$$ t_i \;\approx\; \widehat{t}_i := 1 - \frac{\widehat{p}}{1-\widehat{p}}\mathbb{V}_{\boldsymbol{x}_{-i}}\left[\frac{\widehat{f}_{\boldsymbol{x}_{-i}|y(\boldsymbol{x})>q}(\boldsymbol{x}_{-i})}{f_{\boldsymbol{x}_{-i}}(\boldsymbol{x}_{-i})}\right], \tag{21}$$

where it is reminded that $\widehat{p}$ is the estimated value of p based on one of the sample-based methods presented in Section 2. Finally, generating independent realizations under $\widehat{f}_{x_i}$ and $\widehat{f}_{\boldsymbol{x}_{-i}}$ being quick and easy, Monte-Carlo estimations of $\widehat{s}_i$ and $\widehat{t}_i$ can be calculated numerically with a controlled precision.

4. Robustness analysis of the estimation of p

Indices s_i and t_i, which are defined by Eqs. (17) and (18), are associated with a fixed distribution of the model inputs. In this section, another type of sensitivity indices is defined, which can be used to quantify the robustness of the evaluation of p to small changes of the input distribution. As explained in Introduction, the information provided by these new indices is complementary to the information provided by s_i and t_i. When s_i and t_i allow the identification of the components of $\boldsymbol{x}$ whose variability has to be reduced in priority to decrease the value of p, these new indices can be used to identify the components of $\boldsymbol{x}$ whose distributions have to be particularly well-characterized for a relevant estimation of p.

To quantify the robustness of the estimation of p to small changes of the input distribution, we generally compute the gradient of the failure probability with respect to the parameters that characterize the PDF of the inputs.

9

Indeed, as it was originally proposed in [33], these gradients can efficiently be computed from a simple post-processing of a series of independent and identically distributed failure points provided by any sample-based reliability analysis (such as MCS, SS or MP), without additional computational cost. If an isoprobabilist transform is applied to the model inputs, the link between the parameters describing the input PDF and p is no more explicit. Nevertheless, the PDF of $\boldsymbol{x}$ after the isoprobabilist transform being Gaussian, it is possible to analyse the effect due to small perturbations of the means and of the standard deviations of each model input x_i in this standard space. Hence, at the first order, the sensitivity of p to a small perturbation of the PDF of x_i can be assessed by computing $\widehat{m}_i$ and $\widehat{v}_i$, so that:

$$\widehat{m}_i := \frac{\widehat{p}}{N^*} \sum_{n=1}^{N^*} x_i^{(n)} \ \approx \ \mathbb{E}_{\boldsymbol{x}} \left[1_{y(\boldsymbol{x})>q} x_i \right] = \left. \frac{\partial p}{\partial \mathbb{E}[x_i]} \right|_{\mathbb{E}[x_i]=0}, \qquad (22)$$

$$\widehat{v}_i := \frac{\widehat{p}}{N^*} \sum_{n=1}^{N^*} ((x_i^{(n)})^2 - 1) \ \approx \ \mathbb{E}_{\boldsymbol{x}} \left[1_{y(\boldsymbol{x})>q} (x_i^2 - 1) \right] = \left. \frac{\partial p}{\partial \mathrm{sd}[x_i]} \right|_{\mathrm{sd}[x_i]=1}. \quad (23)$$

There, $\mathrm{sd}[x_i]$ denotes the standard deviation of x_i, and we remind that $x_i^{(1)}, \ldots, x_i^{(N^*)}$ are the ith components of the elements of $\mathcal{D}_f$. As the marginal densities of $\boldsymbol{x}$ are the same, the quantities $\widehat{m}_i$ and $\widehat{v}_i$ can directly be compared to identify the components of $\boldsymbol{x}$, whose PDF has to be controlled in priority for a correct estimation of p. The higher the absolute values of $\widehat{m}_i$ and $\widehat{v}_i$ are, the more attention we have to pay to the way the PDF of x_i was constructed.

Higher degrees derivatives of p can be computed to assess the influence of small changes of the mean and the standard deviation at the same time, or interactions between several components of $\boldsymbol{x}$. However, all these quantities are local and their aggregation to construct one single index that characterizes the total effect of x_i (including or not interactions with other components of $\boldsymbol{x}$) is not trivial. To circumvent this problem, a more global approach is proposed, which is based on the introduction of the following set of univariate Gaussian PDFs:

$$\mathcal{F}(\delta) := \left\{ f \in \mathbb{F}^+(\mathbb{R}) \mid f(x_i) = \varphi(x_i; \mu_i, 1 + \varepsilon_i), \ \max(|\varepsilon_i|, |\mu_i|) \le \delta \right\}, \ \ 0 < \delta < 1. \tag{24}$$

Here, $\mathbb{F}^+(\mathbb{R})$ is the set of functions defined on $\mathbb{R}$ that are positive and function φ is defined by Eq. (7). By construction, for all functions $f_1, \ldots, f_D$ in $\mathcal{F}(\delta)$, the quantity

$$
\begin{aligned}
h(f_1, \ldots, f_D) &:= \int_{\mathbb{X}} 1_{y(\boldsymbol{x}) > q} \prod_{j=1}^{D} f_j(x_j) d\boldsymbol{x} \\
&= \mathbb{E}_{\boldsymbol{x}} \left[1_{y(\boldsymbol{x}) > q} \prod_{j=1}^{D} \frac{f_j(x_j)}{f_{x_j}(x_j)} \right]
\end{aligned}
\tag{25}
$$

defines the probability that $y(\boldsymbol{x})$ is strictly greater than q, under the assumption that the PDF of $\boldsymbol{x}$ is equal to $\prod_{j=1}^{D} f_j$. Then, let p^δ, p_i^δ and p_{-i}^δ be the solutions of the optimization problems defined by Eqs. (26), (27) and (28).

$$
p^\delta := \max_{f_j \in \mathcal{F}(\delta),\ 1 \le j \le D} h(f_1, \ldots, f_D),
\tag{26}
$$

$$
p_i^\delta := \max_{f_i \in \mathcal{F}(\delta)} h(f_{x_1}, \ldots, f_{x_{i-1}}, f_i, f_{x_{i+1}}, \ldots, f_{x_D}),
\tag{27}
$$

$$
p_{-i}^\delta := \max_{f_j \in \mathcal{F}(\delta),\ 1 \le j \le D,\ j \ne i} h(f_1, \ldots, f_{i-1}, f_{x_i}, f_{i+1}, \ldots, f_D).
\tag{28}
$$

By construction, p^δ corresponds to the worst case probability associated with an uncertainty on the PDFs of the inputs controlled by δ. In the same manner, p_i^δ is the worst case probability when there are uncertainties on the PDF of x_i only, whereas p_{-i}^δ is the worst case probability when there are uncertainties on the PDFs of all the components of $\boldsymbol{x}$ but the ith. Thus, $p_i^\delta - p$ corresponds to the part of $p^\delta - p$ that can be explained by uncertainties on the PDF of x_i only, whereas $p_{-i}^\delta - p$ corresponds to the part of $p^\delta - p$ that can be explained by uncertainties on the PDFs of all the components of $\boldsymbol{x}$ but the ith.

Hence, the following indices:

$$
0 \le \varsigma_i^\delta := \frac{p_i^\delta - p}{p^\delta - p} \le 1,
\tag{29}
$$

$$
0 \le \tau_i^\delta := \frac{p^\delta - p_{-i}^\delta}{p^\delta - p} \le 1,
\tag{30}
$$

can be used to quantify the sensitivity of p to perturbations on the PDF of $\boldsymbol{x}$. Approximations of these indices can be obtained by noticing that, for all $f_1, \ldots, f_D$ in $\mathcal{F}(\delta)$,

$$h(f_1, \ldots, f_D) \approx \widehat{h}(f_1, \ldots, f_D) := \frac{\widehat{p}}{N^*} \sum_{n=1}^{N^*} \prod_{j=1}^{D} \frac{f_j(x_j^{(n)})}{f_{x_j}(x_j^{(n)})}. \tag{31}$$

Indeed, if $\widehat{p}^\delta$, $\widehat{p}_i^\delta$ and $\widehat{p}_{-i}^\delta$ are the respective approximations of p^δ, p_i^δ and p_{-i}^δ replacing function h by $\widehat{h}$ in Eqs. (26), (27) and (28), it comes:

$$\varsigma_i^\delta \approx \widehat{\varsigma}_i^\delta := \frac{\widehat{p}_i^\delta - \widehat{p}}{\widehat{p}^\delta - \widehat{p}}, \tag{32}$$

$$\tau_i^\delta \approx \widehat{\tau}_i^\delta := \frac{\widehat{p}^\delta - \widehat{p}_{-i}^\delta}{\widehat{p}^\delta - \widehat{p}}. \tag{33}$$

Index $\widehat{\varsigma}_i^\delta$ can be seen as the individual effect of x_i on p, whereas $\widehat{\tau}_i^\delta$ can be interpreted as the effect of x_i on p including interactions with the other components of $\boldsymbol{x}$. Contrary to the Sobol indices, for which $s_i \leq t_i$, depending on the combined roles of the different elements of $\boldsymbol{x}$, it is possible to construct cases where $\varsigma_i^\delta > \tau_i^\delta$. However, when testing these indices on classical test functions, we always found values of τ_i^δ that were greater than ς_i^δ.

Remarks.

- In this section, an isoprobabilist transform has been applied to each model input before computing the sensitivity indices. It is indeed a very interesting tool to normalize each model input, so that the different indices can be compared quantitatively. However, the class of perturbations could be defined in the "physical" space instead. In that case, it would be important to adapt the perturbations of each model input for a quantitative comparison of the results. Nevertheless, whatever the "physical" distribution of x_i before an eventual isoprobabilist transform (uniform, beta, exponential, and so on), the meanings of indices ς_i^δ and τ_i^δ do not change. They still quantify the influence on p of small changes of the mean and the standard deviation of x_i. Moreover, if for each f_i in $\mathcal{F}(\delta)$, μ_i and ε_i are the two constants so that:

$$f_i(x_i) = \frac{1}{\sqrt{2\pi}(1 + \varepsilon_i)} \exp\left(-\frac{(x_i - \mu_i)^2}{2(1 + \varepsilon_i)^2}\right), \tag{34}$$

the perturbed PDF in the "physical" space that corresponds to f_i can be approximated by $\frac{d}{dx_i}\Phi_{\mu_i,\varepsilon_i+1} \circ \Phi_{0,1}^{-1} \circ F_{x_i}$, where for all a, b in $\mathbb{R} \times \mathbb{R}^{+*}$,

$$\Phi_{a,b}(x) := \int_{-\infty}^{x} \varphi(z; a, 1+b)dz, \quad x \in \mathbb{R}. \tag{35}$$

As an illustration, Figure 1 compares the evolutions of f_{x_i} and $\frac{d}{dx}\Phi_{\mu_i,\varepsilon_i+1} \circ \Phi_{0,1}^{-1} \circ F_{x_i}$ for several values of μ_i and ε_i, and different classes for F_{x_i}.

- In theory, there is no restriction for the definition of the PDF perturbations. But if we are interested in analysing the effect on p due to changes of higher statistical moments (kurtosis, skewness,...), the iso-probabilist transform may not be the most appropriate transformation of the model inputs.

- As the gradient of any function in $\mathcal{F}(\delta)$ with respect to its mean and standard deviation is explicit, the optimization problems associated with Eqs. (26), (27) and (28) can be efficiently solved using any multi-start gradient-based solver. However, noticing that:

$$\prod_{i=1}^{D} \frac{f_i(x_i)}{f_{x_i}(x_i)} = 1 + \sum_{i=1}^{D} \mu_i x_i + (\varepsilon_i + \frac{\mu_i^2}{2})(x_i^2 - 1) + \frac{\varepsilon_i^2}{2}(2 - 5x_i^2 + x_i^4) + \varepsilon_i \mu_i(x_i^3 - 3x_i)$$
$$+ \sum_{1 \leq i < j \leq D} (\varepsilon_i(x_i^2 - 1) + \mu_i x_i)(\varepsilon_j(x_j^2 - 1) + \mu_j x_j) + o(\delta^2),$$
$$\tag{36}$$

it follows that:

$$h(f_{x_1}, \ldots, f_{x_{i-1}}, f_i, f_{x_{i+1}}, \ldots, f_{x_D})$$
$$= p + a_i \mu_i + b_i(\varepsilon_i + \frac{\mu_i^2}{2}) + c_i\frac{\varepsilon_i^2}{2} + d_i\varepsilon_i\mu_i, \tag{37}$$

$$a_i := \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{1}_{y(\boldsymbol{x})>q}x_i\right], \quad b_i := \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{1}_{y(\boldsymbol{x})>q}(x_i^2 - 1)\right], \tag{38}$$

$$c_i := \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{1}_{y(\boldsymbol{x})>q}(2 - 5x_i^2 + x_i^4)\right], \quad d_i := \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{1}_{y(\boldsymbol{x})>q}(x_i^3 - 3x_i)\right]. \tag{39}$$

(a) $x_i \sim U(0,1)$
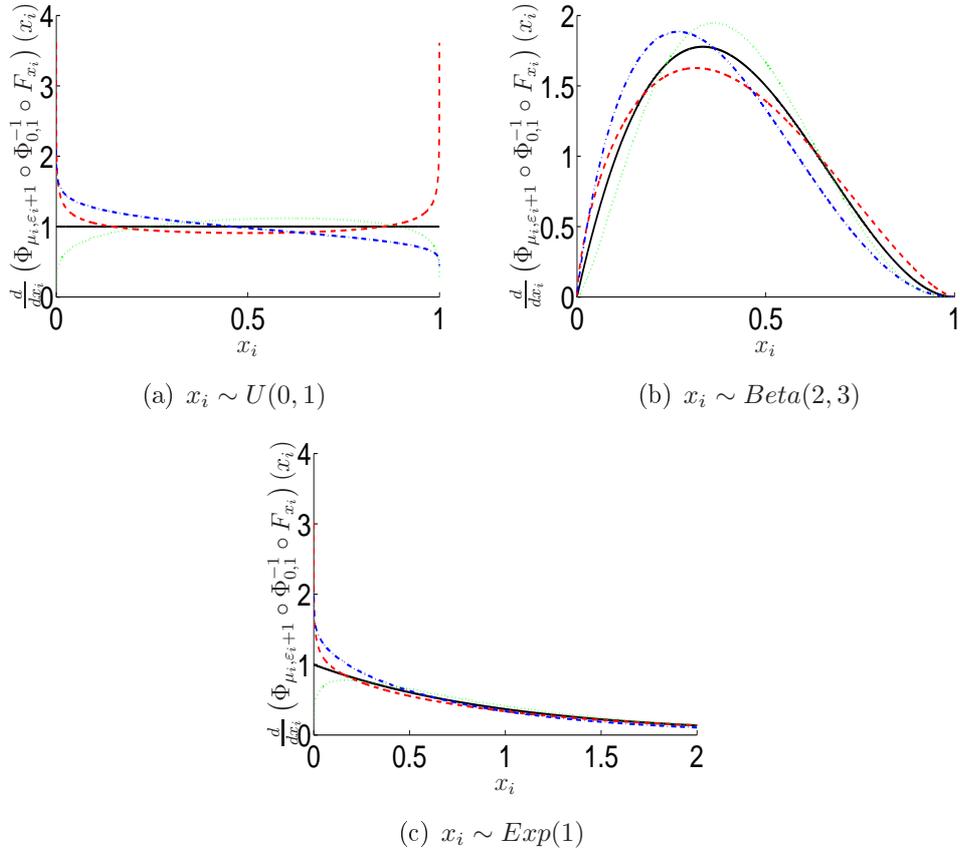
(b) $x_i \sim Beta(2,3)$

(c) $x_i \sim Exp(1)$

Figure 1: Evolution of $\frac{d}{dx_i}\left(\Phi_{\mu_i,\varepsilon_i+1} \circ \Phi_{0,1}^{-1} \circ F_{x_i}\right)(x_i)$ with respect to x_i for several values of μ_i and ε_i, and different classes for F_{x_i}. Black continuous line $\leftrightarrow (\mu_i,\varepsilon_i) = (0,0)$. Red dashed line $\leftrightarrow (\mu_i,\varepsilon_i) = (0,0.1)$. Blue dashed-dotted line $\leftrightarrow (\mu_i,\varepsilon_i) = (-0.2,0)$. Green dotted line $\leftrightarrow (\mu_i,\varepsilon_i) = (0.05,-0.1)$.

Thus, for each $1 \leq i \leq D$, initializing μ_i and ε_i at the points in $[0, \delta] \times [0, \delta]$ that maximize the second-order polynomial $(\mu_i, \varepsilon_i) \mapsto a_i \mu_i + b_i(\varepsilon_i + \frac{\mu_i^2}{2}) + c_i \frac{\varepsilon_i^2}{2} + d_i \varepsilon_i \mu_i$ is a good mean to accelerate the convergence of these algorithms.

- The set $\mathcal{F}(\delta)$ is parametrized by the scalar δ, which characterizes the maximal amplitude of the perturbation of the PDF of $\boldsymbol{x}$. For instance, the value $\delta = 0.1$ indicates that the uncertainty on the mean and the standard deviation of each component of $\boldsymbol{x}$ is less than 10% of the nominal value of the standard deviation. Alternatively, the value of δ can be associated with a chosen increase of the probability of exceeding q. For instance, in the applications that will be presented in Section 5, δ is chosen so that $p^\delta = 2 \times p$.

5. Surrogate model-based sensitivity indices

In the previous section, the estimations of p and of the different reliability-oriented sensitivity indices are based on evaluations of the true code. When small values of p are considered, this implies the code to be evaluated in a very high number of input values. If the computational cost associated with one evaluation of the code is high, this may not be possible in practice. Surrogate models have therefore to be constructed to emulate the code answer from a limited number of calls to the true code. As presented in Introduction, we propose to focus on the Gaussian process regression (GPR) to build this surrogate. To this end, performance function y is supposed to be a sample path of a particular stochastic process $\widetilde{y}$, whose mean function and covariance function are denoted by $\widetilde{\mu}$ and $\widetilde{c}$:

$$\widetilde{y} \sim \mathrm{GP}(\widetilde{\mu}, \widetilde{c}). \tag{40}$$

Conditioning this stochastic process by $R \geq 1$ code evaluations gathered in the set $\mathcal{S}^{\mathrm{learn}}(R) := \left\{ (\widetilde{\boldsymbol{x}}^{(r)}, y(\widetilde{\boldsymbol{x}}^{(r)})), \ 1 \leq r \leq R \right\}$, a very interesting predictor for the value of y in any unobserved point $\boldsymbol{x}$ in $\mathbb{X}$ can be obtained. This predictor is denoted by $\widetilde{y} \mid \mathcal{S}^{\mathrm{learn}}(R)$, and it verifies:

$$\widetilde{y} \mid \mathcal{S}^{\mathrm{learn}}(R) := \widetilde{y}_R \sim \mathrm{GP}(\widetilde{\mu}_R, \widetilde{c}_R). \tag{41}$$

The interested reader is referred to [35, 37] for further details about the choice and the optimisation of functions $\widetilde{\mu}$ and $\widetilde{c}$, and about the expres-

sions of the conditioned mean, $\widetilde{\mu}_R$, and the conditioned covariance, $\widetilde{c}_R$. For the applications presented in Section 6, $\widetilde{\mu}$ will be chosen as a second-order polynomial, and $\widetilde{c}$ will be chosen as an element of the 5/2-Matern class of covariance functions.

If we replace y by $\widetilde{y}_R$ in the evaluation of probability p, which is given by Eq. (5), the quantity

$$\widetilde{p} := \mathbb{P}_{\boldsymbol{x}}(\widetilde{y}_R(\boldsymbol{x}) > q) \tag{42}$$

is no longer deterministic but random, as for a fixed value of $\boldsymbol{x}$ in $\mathbb{X}$, $\widetilde{y}_R(\boldsymbol{x})$ is random. Let $\widetilde{y}_R(\cdot\,;\omega)$ denote one particular realization of $\widetilde{y}_R$. Using Algorithm 1, it is possible to gather in $\widetilde{D}_f(\omega)$ N^* points that are statistically independent, and whose distribution is $f_{\boldsymbol{x}\,|\widetilde{y}_R(\boldsymbol{x};\omega)>q}$. For this particular realization of $\widetilde{y}_R$, this algorithm also provides the estimate $\widetilde{p}(\omega)$, so that:

$$\widetilde{p}(\omega) \approx \mathbb{P}(\widetilde{y}_R(\boldsymbol{x};\omega) > q). \tag{43}$$

Hence, based on $\widetilde{D}_f(\omega)$ and $\widetilde{p}(\omega)$, it is possible to approximate the values of the different reliability-oriented sensitivity indices that were introduced in Section 3. Repeating this procedure several times, we eventually compute confidence intervals for these sensitivity indices, which aggregate two sources of uncertainties:

- the uncertainty that is due to the replacement of true function y by its surrogate model,

- the uncertainty that is due to the finite value of N^* for the estimation of $\widetilde{p}(\omega)$ and the estimation of the reliability-oriented sensitivity indices.

In practice, the uncertainty due to the surrogate model can be reduced by adding new code evaluations to the learning set. Such new points are generally chosen iteratively where the expected value of $\widetilde{y}_R$ is the closest to the threshold q, with the largest uncertainty. Hence, in the following applications, the construction of the surrogate models will be based on a two-step procedure. First, the code is evaluated in a small-dimensional space filling design of experiments (see [21, 26, 23, 12, 28]). Then, new points are added one by one using a stepwise uncertainty reduction (SUR) strategy

(see [2, 6] for more details about this particular selection criterion, and [4, 9, 13, 27] for alternative selection criteria) until the maximum computational budget is attained.

Remarks.

- Algorithm 1 is a particular application of the Moving Particle approach that is presented in Section 2. Hence, for this algorithm, it can be noticed that the mean number of points on which each realization of $\widetilde{y}_R$ has to be projected is equal to $R + N^*(1 - \log(p))$. Hence, the computational budget associated with the different conditionings of $\widetilde{y}$ in algorithm 1 stays reasonable, even for low values of p.

- Algorithm 1 is based on a simple Monte Carlo procedure for the generation of the realizations of $\boldsymbol{x}$. If very low values of p were considered, more efficient algorithms, such as Markov chain Monte Carlo (McMC) approaches, could be used to better concentrate the realizations of $\boldsymbol{x}$ in the region of potentially high values of $\widetilde{y}_R$.

6. Applications

Three examples are presented in this section to illustrate the interest of the former developments. The two first examples are based on analytical data, whereas the third one is based on an industrial case. The evaluations of the different failure probabilities are performed using the R package *mistral* [11].

6.1. Polynomial function

In this section, the quantity of interest is given by:

$$y : \begin{cases} \mathbb{R}^3 \to \mathbb{R} \\ \boldsymbol{x} = (x_1, x_2, x_3) \ \mapsto \ (1 + x_1)(5 + x_2)(10 + x_3) \end{cases} . \quad (44)$$

Here, x_1, x_2, x_3 are supposed to be three independent centred Gaussian random variables, whose variance is one, and we are interested in the probability that $y(\boldsymbol{x})$ exceeds the threshold $q = 250$. Three configurations associated with different available computational budgets are compared for the estimation of p and the computation of the different sensitivity indices.

1 Initialization : ;

2 let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N^*)}$ be N^* independent realizations of random vector $\boldsymbol{x}$;

3 let $(y(\boldsymbol{x}^{(1)}, \omega), \ldots, y(\boldsymbol{x}^{(N^*)}, \omega))$ be a particular realization of the random vector $(\widetilde{y}_R(\boldsymbol{x}^{(1)}), \ldots, \widetilde{y}_R(\boldsymbol{x}^{(N^*)}))$;

4 $\widetilde{y}_R^\star := \widetilde{y}_R \mid \widetilde{y}_R(\boldsymbol{x}^{(n)}) = y(\boldsymbol{x}^{(n)}, \omega)), \ 1 \leq n \leq N^*$;

5 $M = 0$;

6 $\widetilde{D}_f(\omega) = \emptyset$;

7 for $1 \leq n \leq N^*$ **do**

8 $\boldsymbol{x}^\star = \boldsymbol{x}^{(n)}$, $z = y(\boldsymbol{x}^{(n)}, \omega)$;

9   **while** $z < q$ **do**

10   draw at random a realization of $\boldsymbol{x}$, denoted by $\boldsymbol{x}^\star$;

11   draw at random a realization of $\widetilde{y}_R^\star(\boldsymbol{x}^\star)$, denoted by $y^\star(\omega)$;

12   **if** $y^\star(\omega) > z$ **then**

13 $z = y^\star(\omega)$, $M = M + 1$;

14 $\widetilde{y}_R^\star = \widetilde{y}_R^\star \mid \widetilde{y}_R^\star(\boldsymbol{x}^\star) = y^\star(\omega)$;

15   **end**

16   **end**

17 $\widetilde{D}_f(\omega) = \widetilde{D}_f(\omega) \cup \{\boldsymbol{x}^\star\}$;

18 end

19 $\widetilde{p}(\omega) = (1 - 1/N^*)^M$.

Algorithm 1: Generation of $N^\star$ failure points associated with one particular realization of Gaussian process $\widetilde{y}_R$, and estimation of the associated probability of exceeding q.

1. "Reference". A crude Monte Carlo based on 6×10^6 code evaluations is first considered as a reference. 5×10^6 evaluations will be used to compute the Sobol indices associated with the indicator function in an optimized sample-based approach using the R package "soboljansen". Although the Sobol indices are positive and inferior to one by construction, negative values can be obtained when the true values are very close to zero, and in the same manner, values greater than one can be obtained when the true value is very close to one. This is due to the finite number of code evaluations, and to the sample-based method that is used to compute these indices. The last 10^6 code evaluations are used to assess the value of $p = \mathbb{P}(y(\boldsymbol{x}) > q)$. Based on these evaluations, the interval $[8.0 \times 10^{-4}; 9.1 \times 10^{-4}]$ gives a 95% confidence interval for p.

2. "Code+MP". Then, the MP approach presented in Section 2 is considered, with $Q = 100$. It is coupled to a McMC approach with $T = 10$ for the conditional sampling. In that case, the sensitivity indices are computed from the sole simulations used to estimate p. It is recalled that the mean number of code evaluations for the estimation of p is $Q(1 - T \log(p)) \approx 7200$.

3. "GP+MP". The third approach replaces the true code output by a Gaussian surrogate. This surrogate is based on 50 code evaluations only (30 for the initial design of experiments, plus 20 for the sequential enrichment using a SUR criterion). The estimations of p and the sensitivity indices are based on the MP approach described in Algorithm 1. Hence, no additional code evaluation is required once the surrogate model is computed. The uncertainty due to the replacement of the true code by its surrogate can be empirically estimated by repeating the procedure several times.

The results are summarized in Table 1. For this simple example, each sensitivity index globally comes to the same conclusion: the most important role is played by x_1, whereas x_3 has the less significant impact on p. The importance of the combined effect of the three parameters can also be quantified by comparing the values of $\widehat{s}_i$ and $\widehat{t}_i$ on the one hand, and the values of $\widehat{\varsigma}_i^\delta$ and $\widehat{\tau}_i^\delta$ on the other hand. The fact that the total effects $\widehat{t}_i$ are always much stronger than the individual $\widehat{s}_i$ clearly indicates that y is sensitive to the

three model inputs, and that specific combinations are necessary to exceed q. This is not really surprising, and we could assume that, when q increases, the values of $\widehat{t_i}$ should tend to 1 whereas the values of $\widehat{s_i}$ should tend to 0. For indices $\widehat{\varsigma}_i^\delta$ and $\widehat{\tau}_i^\delta$, $1 \le i \le 3$, it is recalled that the value of δ is chosen so that $p^\delta = 2 \times p$. This corresponds to $\delta = 0.05$ for this example, which means that a 5% uncertainty on the means and the standard deviations of x_1, x_2, x_3 can multiply, in the worst case, the value of p by two. Looking at the values of $\widehat{\varsigma}_i^\delta$ and $\widehat{\tau}_i^\delta$, we notice that more than half of this increase could be due to the individual effect of x_1. This even more lay stress on the crucial role played by x_1.

Additionally, we notice that the values of the different sensitivity indices are very close whatever the considered configuration. Hence, passing from a crude Monte Carlo approach (the Reference configuration) to a more sophisticated splitting method (the Code + MP configuration) does not affect the results of the sensitivity analysis. In the same manner, the effect of replacing the true code by a surrogate model that is refined in the region where $y(\boldsymbol{x})$ is close to q has a reduced effect on the values of the sensitivity indices. Including the surrogate uncertainty in the estimation only tends to increase the amplitudes of the confidence intervals for these indices, but this is almost negligible compared to the uncertainty due to the fact that there are only few code evaluations in the failure domain. Finally, coupling the Gaussian process regression and the Moving Particle approach is a good mean to compute at a very reasonable computational cost the different sensitivity indices presented in Sections 3 and 4.

6.2. Non-linear oscillator

The second example corresponds to the test case in dimension $D = 6$ presented in [10, 3]. Here, $x_1, \dots, x_6$ are still independent standard Gaussian random variables, and we have:

$$
y : \begin{cases} \mathbb{R}^6 \to \mathbb{R} \\ \boldsymbol{x} \mapsto 0.15x_4 + \left| \dfrac{0.9 + 0.15x_5}{1.1 + 0.1x_2 + 0.01x_3} \sin\left(\sqrt{\dfrac{1.1 + 0.1x_2 + 0.01x_3}{1 + 0.05x_1}} \dfrac{1 + 0.2x_6}{2} \right) \right| \end{cases} \cdot
\tag{45}
$$

We focus on the threshold $q = 0.95$. Using a crude Monte Carlo approach based on 10^6 code evaluations, we found that p is around 3.7×10^{-3}. A reliability-oriented sensitivity analysis was then carried out, whose results

		$\widehat{s}_i(\%)$	$\widehat{t}_i(\%)$	$\widehat{\varsigma}_i^\delta(\%)$	$\widehat{\tau}_i^\delta(\%)$
i=1	Reference	[1;21]	[94;107]	[57;61]	[70;73]
	Code+MP	[6;18]	[97;99]	[54;64]	[67;75]
	GP+MP	[8;15]	[96;99]	[53;64]	[67;76]
i=2	Reference	[-14;10]	[74;87]	[18;21]	[28;31]
	Code+MP	[0;1]	[41;81]	[16;22]	[25;32]
	GP+MP	[0;1]	[66;91]	[15;23]	[24;35]
i=3	Reference	[-9;8]	[47;56]	[7;9]	[12;15]
	Code+MP	[0;0]	[41;78]	[6;11]	[10;18]
	GP+MP	[0;0]	[50;83]	[6;11]	[10;18]

Table 1: Values of the sensitivity indices for $q = 250$ (corresponding to $\widehat{p} = 8.5 \times 10^{-4}$ [$8.0 \times 10^{-4}; 9.1 \times 10^{-4}$]) for the polynomial function. 10^6 code evaluations have been used to compute the reference values. The "code+MP" values are based on around 7200 code evaluations (around 710 iterations were required for the Moving Particles algorithm, and the burn-in parameter was chosen equal to 10 for the conditional generation using a MCMC algorithm). The "GP+MP" values are based on only 50 code evaluations (30 codes evaluations for the initial DoE + 20 sequentially added code evaluations using a SUR criterion). The values between brackets correspond to 95% confidence intervals, which have been empirically computed from 100 repetitions of the whole procedure.

are gathered in Table 2. Once again, the results of the three configurations presented in the previous section are compared. According to the values of the different indices, we find that the most influential components of $\boldsymbol{x}$ are x_4 and x_5, which is completely coherent with the expression of y. For this example, it is interesting to notice that although there is *a priori* no interaction between x_4 and the other components of $\boldsymbol{x}$ in the expression of y, $\widehat{t_4}$ is much bigger than $\widehat{s_4}$. This underlines an important difference between classical Sobol indices, which focus on the mean answer of y, and the proposed reliability-oriented Sobol indices, which focus on high quantiles of y. Indeed, even if there is no explicit link between x_4 and the other components of $\boldsymbol{x}$, for $y(\boldsymbol{x})$ to exceed relatively high values of q, specific combinations of most of the components of $\boldsymbol{x}$ are required. For this example, to guarantee that $p^\delta = 2 \times p$, δ was chosen equal to 0.058. Hence, relaxing the values of the mean and the standard deviation of component x_4 by less than 6% can lead to an increase of more than 50% for p, when almost no effect would be noticed if the same relaxation was achieved for x_1 or x_3. Thus, to get a relevant estimation of p, a precise characterization of the distribution of x_4 is required.

In this example, the values for the three configurations (Reference, Code+MP and GP+MP) are once again very close, which tends to validate the coupling of the Gaussian process regression and the Moving Particle approach to carry out relevant reliability-oriented sensitivity analyses at a reduced computational cost. Nevertheless, some differences can be observed for the estimation of the Sobol total indices. This is due to the fact that the non-parametric estimations of the conditioned PDFs are based on $Q = 100$ points, which is few compared to the dimension of their definition domain (equal to $d-1 = 5$ for this example). Better estimates could be obtained by increasing Q, without any additional code evaluations.

6.3. Pressure tank under dynamic pressure

The third example is derived from an industrial application, and deals with the reliability analysis of a spherical containment vessel (including a tap in a different material) subject to an internal blast. A picture of this containment vessel is represented in Figure 2. The mechanical response (displacement, strain and stress tensors) of the vessel is modelled with a coupling of two independent numerical codes. First, a hydrodynamic code simulates the explosion of a bursting charge placed at the center of the tank using a 2-dimensional Eulerian scheme. The dynamical loading obtained at the inner

		$\widehat{s}_i(\%)$	$\widehat{t}_i(\%)$	$\widehat{\varsigma}_i^\delta(\%)$	$\widehat{\tau}_i^\delta(\%)$
i=1	Reference	[-4;3]	[11;14]	[1;2]	[2;4]
	Code+MP	[0;1]	[16;47]	[1;4]	[1;6]
	GP+MP	[0;1]	[14;36]	[1;4]	[1;7]
i=2	Reference	[-5;4]	[29;33]	[4;7]	[7;11]
	Code+MP	[0;2]	[15;52]	[3;9]	[5;15]
	GP+MP	[0;2]	[10;33]	[3;8]	[5;15]
i=3	Reference	[-4;3]	[3;4]	[0;1]	[0;3]
	Code+MP	[0;0]	[15:42]	[0;2]	[0;4]
	GP+MP	[0;1]	[10;33]	[0;2]	[1;4]
i=4	Reference	[19;28]	[95;101]	[54;61]	[66;71]
	Code+MP	[13;46]	[89;97]	[49;62]	[61;73]
	GP+MP	[17;62]	[89;98]	[51;62]	[62;72]
i=5	Reference	[-4;6]	[68;74]	[18;22]	[25;31]
	Code+MP	[1;4]	[39;82]	[16;24]	[23;34]
	GP+MP	[1;8]	[21;71]	[15;24]	[23;35]
i=6	Reference	[-4;4]	[15;17]	[1;3]	[3;5]
	Code+MP	[0;1]	[17;47]	[1;5]	[2;8]
	GP+MP	[0;0]	[12;31]	[1;4]	[2;7]

Table 2: Values of the sensitivity indices for $q = 0.95$ (corresponding to $\widehat{p} = 3.7 \times 10^{-3}$ [$3.6 \times 10^{-3}; 3.8 \times 10^{-3}$]) for the non linear oscillator. 10^6 code evaluations have been used to compute the reference values. The "code+MP" values are based on around 5700 code evaluations (around 560 iterations were required for the Moving Particles algorithm, and the burn-in parameter was chosen equal to 10 for the conditional generation using a MCMC algorithm). The "GP+MP" values are based on only 100 code evaluations (60 codes evaluations for the initial DoE + 40 sequentially added code evaluations using a SUR criterion). The values between brackets correspond to 95% confidence intervals, which have been computed from 100 repetitions of the whole procedure.
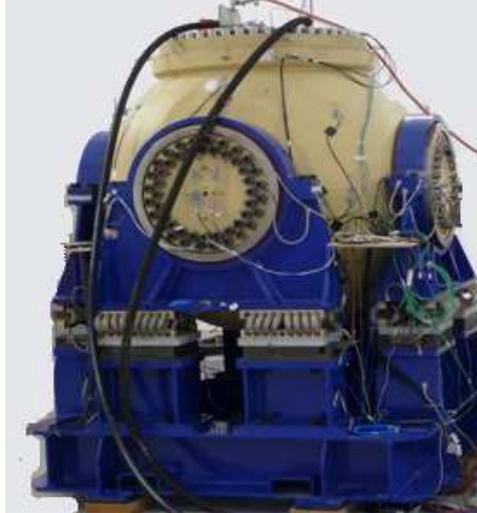
Figure 2: Picture of the studied containment vessel

surface of the tank is then applied as input of a structural code, which simulates the vibrations of the vessel under this dynamic excitation and computes the induced displacements at each time step in each position of the vessel. For this third example, we are interested in the cumulated equivalent plastic strain over a given time range. Let $\boldsymbol{z} = (z_1, \ldots, z_8)$ be eight uncertain parameters that characterize the nested simulation, whose properties are summarized in Table 3. Let $\boldsymbol{x}$ be the image of $\boldsymbol{z}$ by the isoprobabilist transform, and $y(\boldsymbol{x})$ be the maximum value in time and space of the cumulated equivalent plastic strain.

We focus on the case $q = 0.05$. Each evaluation of y being extremely time-demanding, the idea is to estimate p and carry out the reliability-oriented sensitivity analysis using as few calls to y as possible. In that case, nor the Monte-Carlo neither the Code+MP configurations are admissible. Thus, a surrogate model was constructed using 1286 code evaluations (500 codes evaluations for the initial DoE + 786 sequentially added code evaluations using a SUR criterion). The estimation of p and the sensitivity analysis was then carried out on the surrogate instead of the true code, but taking into account the uncertainties due to the substitution of the true code. The results are summarized in Table 4. These values are associated with an estimation of p close to 2.2×10^{-3} (a 95% confidence interval for this estimation is given by $[1.3 \times 10^{-3}; 3.4 \times 10^{-3}]$).

24

Model input	Meaning	Distribution	Parameters
z_1	Internal radius of the tank (m)	Normal	$(0.720, 0.005)$
z_2	Thickness (m)	Log-normal	$(0.073, 0.0015)$
z_3	Scaling factor on inner pressure	Weibull	$(24.95, 1.022)$
z_4	Scaling factor on time	Weibull	$(24.95, 1.022)$
z_5	Young modulus of the tank (Pa)	Log-normal	$(2.1 \times 10^{11}, 2.1 \times 10^{10})$
z_6	Elastic limit of the tank (Pa)	Normal	$(7 \times 10^8, 3 \times 10^7)$
z_7	Young modulus of the tap (Pa)	Log-normal	$(2.1 \times 10^{11}, 2.1 \times 10^{10})$
z_8	Elastic limit of the tap (Pa)	Normal	$(8.6 \times 10^8, 3 \times 10^7)$

Table 3: Stochastic model of the pressure tank. For Normal and Log-normal distributions, the two parameters corresponds to the mean and the standard deviation. For Weibull distribution, the two parameters are the shape and the scale parameters. Inputs z_3 and z_4 characterize the inner pressure history changing the impulsive load seen by the vessel. The different model inputs are statistically independent.

First, if we focus on the values of $\widehat{s}_i$ and $\widehat{t}_i$, we notice that, for this application, the fact that y exceeds q seems not to be due to the action of one unique parameter, but to a combined effect of all the parameters. This is typical for optimized systems, for which the values of each input parameters are generally chosen as a compromise between performance and cost limitation. However, if we look at the values of $\widehat{\varsigma}_i^{\delta}$ and $\widehat{\tau}_i^{\delta}$, it appears that the value of p is very sensitive to the PDF of z_5 (the Young modulus of the tank). For this application, the amplitude of the PDF perturbation is controlled by $\delta = 0.077$ to make p^{δ} be twice as high as p. To better realize that this value of δ only allows small perturbations of the input PDF, Figure 3 compares the marginal densities of the initial PDF of $\boldsymbol{z}$ to the marginal densities of the worst case PDF of $\boldsymbol{z}$. Hence, to guarantee a correct estimation of p, it is crucial to pay a particular attention to the definition of the PDF of z_5.

7. Conclusions

This work considers the challenging problem of carrying out at the same time the reliability and the sensitivity analyses of a complex system, whose behaviour can be modelled by a computationally demanding computer code. To this end, several reliability-oriented sensitivity indices were presented. These indices allow us to identify the model inputs whose variability has to be reduced in priority to minimize the failure probability p, but also the
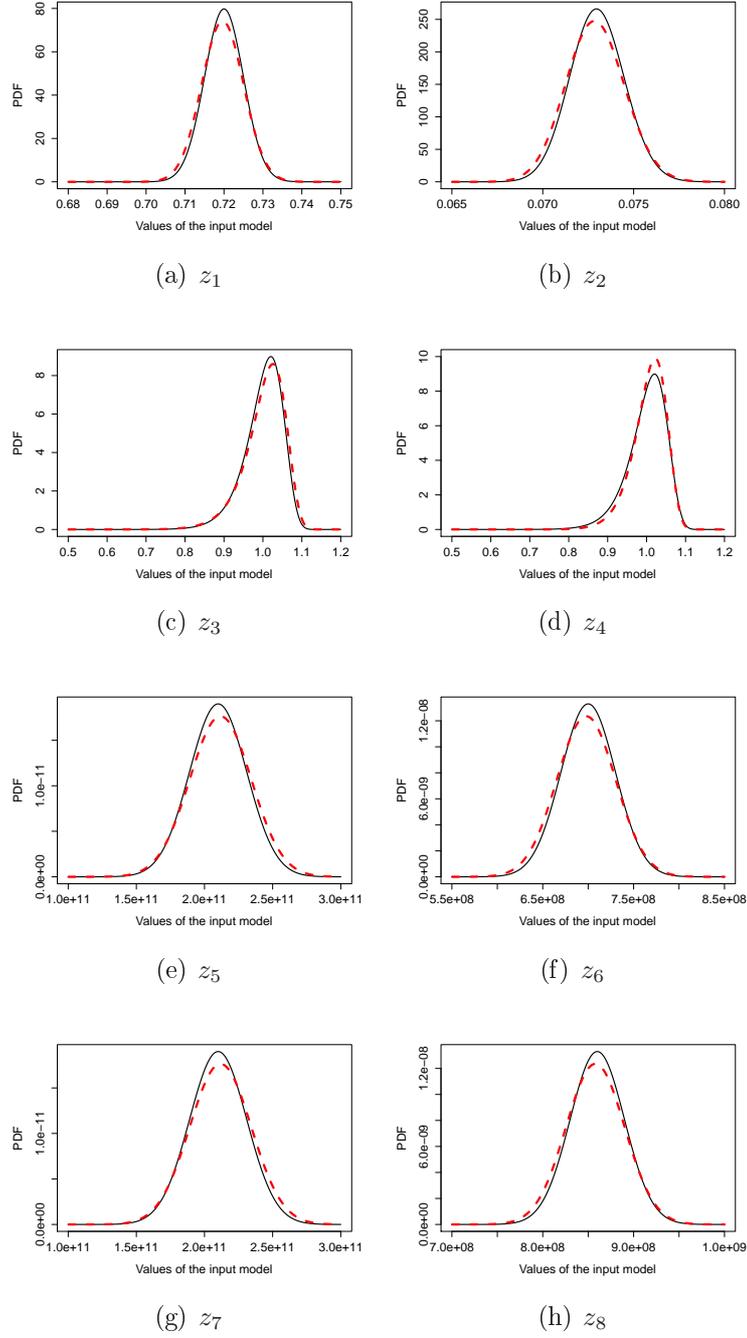
(a) z_1

(b) z_2

(c) z_3

(d) z_4

(e) z_5

(f) z_6

(g) z_7

(h) z_8

Figure 3: Comparison between the original and the perturbed PDFs of the components of $\boldsymbol{z}$. Considering the perturbed PDFs instead of the original ones multiply the value of p by a factor two. Black continuous line $\leftrightarrow$ original PDFs. Red dashed line $\leftrightarrow$ perturbed PDFs.

	$\widehat{s}_i(\%)$	$\widehat{t}_i(\%)$	$\widehat{\varsigma}_i^{\delta}(\%)$	$\widehat{\tau}_i^{\delta}(\%)$
i=1	[0;2]	[44;89]	[0;4]	[1;7]
i=2	[0;2]	[50;91]	[2;9]	[3;14]
i=3	[0;0]	[47;89]	[3;9]	[5;15]
i=4	[0;1]	[64;93]	[5;8]	[12;16]
i=5	[4;13]	[88;99]	[45;64]	[57;76]
i=6	[0;0]	[45;89]	[2;10]	[3;17]
i=7	[0;0]	[43;89]	[1;5]	[1;9]
i=8	[0;0]	[43;90]	[0;2]	[1;8]

Table 4: Values of the sensitivity indices for $q = 0.05$ (corresponding to $\widehat{p} = 2.2 \times 10^{-3}$ [$1.3 \times 10^{-3}; 3.4 \times 10^{-3}$]) for the pressure tank. The "GP+MP" values are based on 1286 code evaluations (500 codes evaluations for the initial DoE + 786 sequentially added code evaluations using a SUR criterion). The values between brackets correspond to 95% confidence intervals, which have been computed from 100 repetitions of the whole procedure, but for a unique surrogate model.

model inputs whose distribution has to be particularly well-characterized for a correct estimation of p. Based on the coupling of a Gaussian process regression and a Moving Particle approach, a method was proposed to efficiently estimate p and these sensitivity indices from a reduced set of code evaluations. The relevance of this approach was illustrated on three examples.

In the different applications, it has been shown that small perturbations (such as modifications of less than 5% of the mean and/or the standard deviation) of the input PDFs could strongly change the value of p. This questions the interest of computing very small failure probabilities for fixed input PDFs, when these PDFs are not perfectly known, as it is the case in many engineering applications. On the contrary, computing different values of p associated with different perturbations of the input PDF seems more appropriate. In this work, we limited these perturbations to small changes of the means and the standard deviations of the different model inputs. But working on more sophisticated perturbations seems an interesting perspective.

Appendix

Proof of Proposition 1

First, let us notice that $\mathbb{V}_{\boldsymbol{x}} \left[1_{y(\boldsymbol{x}) > s} \right] = p(1 - p)$. Then, using Bayes theorem and the equality given by Eq. (14), it follows that:

$$\mathbb{E}_{x_i}\left[\left(\mathbb{P}_{\boldsymbol{x}}(y(\boldsymbol{x})>q)-\mathbb{P}_{\boldsymbol{x}_{-i}}(y(\boldsymbol{x})>q\mid x_i)\right)^2\right]=\mathbb{E}_{x_i}\left[\left(p-p\frac{f_{x_i\mid y(\boldsymbol{x})>q}(x_i)}{f_{x_i}(x_i)}\right)^2\right]$$

$$=p^2\mathbb{E}_{x_i}\left[\left(1-\frac{f_{x_i\mid y(\boldsymbol{x})>q}(x_i)}{f_{x_i}(x_i)}\right)^2\right],$$

$$(46)$$

$$\mathbb{E}_{\boldsymbol{x}_{-i}}\left[\left(\mathbb{P}(y(\boldsymbol{x})>q)-\mathbb{P}(y(\boldsymbol{x})>q\mid \boldsymbol{x}_{-i})\right)^2\right]=p^2\mathbb{E}_{\boldsymbol{x}_{-i}}\left[\left(1-\frac{f_{\boldsymbol{x}_{-i}\mid y(\boldsymbol{x})>q}(\boldsymbol{x}_{-i})}{f_{\boldsymbol{x}_{-i}}(\boldsymbol{x}_{-i})}\right)^2\right].$$

$$(47)$$

As $\mathbb{E}_{x_i}\left[\frac{f_{x_i\mid y(\boldsymbol{x})>q}(x_i)}{f_{x_i}(x_i)}\right]=\mathbb{E}_{\boldsymbol{x}_{-i}}\left[\frac{f_{\boldsymbol{x}_{-i}\mid y(\boldsymbol{x})>q}(\boldsymbol{x}_{-i})}{f_{\boldsymbol{x}_{-i}}(\boldsymbol{x}_{-i})}\right]=1$, we eventually find the searched results.

References

[1] S.K. Au and J.L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16 (4):263–277, 2001.

[2] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vasquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22:773–797, 2012.

[3] J. Bect, L. Li, and E. Vasquez. Bayesian subset simulation. *arXiv preprint arXiv:1601.02337*, 2016.

[4] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan, and J.M. McFarland. Efficient global reliability analysis for non linear implicit performance functions. *AIAA Journal*, 46(10), 2008.

[5] E. Castillo, J.M. Sarabia, C. Solares, and P. Gomez. Uncertainty analyses in fault trees and bayesian networks using form/sorm methods. *Reliability Engineering and System Safety*, 65:29–40, 1999.

[6] C. Chevalier, J. Bect, D. Ginsburger, E. Vasquez, V. Picheny, and Y. Richet. Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56 (4):455–465, 2014.

[7] F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25 (2):417–443, 2007.

[8] F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential monte carlo for rare event simulation. *Statistics and Computing*, 22 (3):795–808, 2012.

[9] B. Echard, N. Gayton, and M. Lemaire. Ak-mcs: an active learning reliability method combining kriging and monte carlo simulation. *Structural Safety*, 22:145–154, 2011.

[10] B. Echard, N. Gayton, M. Lemaire, and N. Relun. A combined importane sampling and kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliability Engineering and System Safety*, 111:232–240, 2013.

[11] G. Defaux et al. mistral: Methods in structural reliability. *R package version 2.1-0*, 2014.

[12] K.T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall, Computer Science and Data Analysis Series, London, 2006.

[13] W. Fauriat and N. Gayton. Ak-sys: an adaptation of the ak-mcs method for system reliability. *Reliability Engineering and System Safety*, 123:137–144, 2014.

[14] A. N. Freudenthal. Safety and the probability of structural failure. *Transactions ASCE*, 121:1337–1397, 1956.

[15] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57 (1):97–109, 1970.

[16] H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, 12:27–30, 1951.

[17] R. Lebrun and A. Dutfoy. An innovating analysis of the nataf transformation from the copula viewpoint. *Prob. Eng. Mech.*, 24:312–320, 2009.

[18] M. Lemaire. *Structural reliability*. Wiley, 2009.

[19] P. Lemaitre, E. Sergienko, A. Arnaud, N. Bousquet, F. Gamboa, and B. Iooss. Density modification-based reliability sensitivity analysis. *Journal of Statistical Computation and Simulation*, 85 (6):1200–1223, 2015.

[20] C. Li and S. Mahadevan. An efficient modularized sample-based method to estimate the first-order sobol' index. *Reliability Engineering and System Safety*, 153:110–121, 2016.

[21] M McKay, R Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[22] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[23] M Morris and T Mitchell. Exploratory designs for computationnal experiments. *Journal of Statistical Planning and Inference*, 43:381–402, 1995.

[24] A. Nataf. Determination des distributions dont les marges sont donnees. *C. R. Acad. Sci. Paris*, 225:42–43, 1962.

[25] A. Owen. Better estimation of small sobol sensitivity indices. *ACM Trans. Model. Comput. Simul.*, 23:11, 2013.

[26] J. S. Park. Optimal latin-hypercube designs for computer experiments ? *Journal of Statistical Planning and Inference*, 39:95–111, 1994.

[27] G. Perrin. Active learning surrogate models for the conception of systems with multiple failure modes. *Reliability Engineering and System Safety*, 149:130–136, 2016.

[28] G. Perrin and C. Cannamela. A repulsion-based method for the definition and the enrichment of opotimized space filling designs in constrained input spaces. *Journal de la Société Française de Statistique*, 158(1):37–67, 2017.

[29] G. Perrin, C. Soize, D. Duhamel, and C. Funfschilling. Identification of polynomial chaos representations in high dimension from a set of realizations. *SIAM J. Sci. Comput.*, 34(6):2917–2945, 2012.

[30] G. Perrin, C. Soize, and N. Ouhbi. Data-driven kernel representations for sampling with an unknown block dependence structure under correlation constraints. *Journal of Computational Statistics and Data Analysis*, 119:139–154, 2018.

[31] R. Rackwitz and B. Fiessler. Structural reliability under combined load sequences. *Journal of computer structures*, 9:479–484, 1978.

[32] M. Rosenblatt. Remarks on a multivariate transformation. *Ann. Math. Stat.*, 23:470–472, 1952.

[33] R. T. Rubinstein. The score function approach for sensitivity analysis of computer simulation models. *Math. Comput. Simul.*, 28(5):351–379, 1986.

[34] R. T. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method.* John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.

[35] J. Sacks, W. Welch, T. Mitchell, and H. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.

[36] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, and D. Gatelli. *Global sensitivity analysis: the primer.* John Wiley & Sons, 2008.

[37] T. J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments.* Springer, New York, 2003.

[38] I.M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Math. Compu. Simul.*, 55:271–280, 2001.

[39] I.M. Sobol and E.E. Myshetskaya. Monte carlo estimators for small sensitivity indices. *Monte Carlo Methods Appl.*, 13:455–465, 2008.

[40] I.M. Sobol, S. Tarantola, D. Gatelli, S.S. Kucherenko, and W. Mauntz. Estimating the approximation error when fixing unessential factors in

global sensitivity analysis. *Reliability Engineering and System Safety*, 92:957–960, 2007.

[41] C. Walter. Moving particles: A parallel optimal multilevel splitting method with application in quantiles estimation and meta-model based algorithms. *Structural Safety*, 55:10–25, 2015.