Point process-based approaches for the reliability analysis of systems modeled by costly simulators

Guillaume Perrin

HAL Id: hal-03228196

https://hal.science/hal-03228196

Submitted on 17 May 2021

Point process-based approaches for the reliability analysis of systems modeled by costly simulators

G. Perrin[a]

[a] *COSYS, Université Gustave Eiffel, 77420 Champs-sur-Marne, France*

Abstract

This paper addresses the issue of guaranteeing the good functioning of physical systems using expensive simulators. More precisely, it is interested in the construction of bounds allowing to majorise with a specified confidence the probability of occurrence of undesired events. In this context, this paper presents two algorithms: a first one allowing to build a bound higher than this probability at a fixed number of simulator evaluations; a second one allowing to reduce as much as possible this bound by adding in an optimized way new simulator evaluations. The efficiency of these algorithms is finally illustrated through the analysis of several test functions.

Keywords: risk analysis, surrogate modeling, Gaussian process regression, quantile estimation, error control

1. Introduction

The reliability analysis of complex systems is increasingly based on simulation. In this work, we focus on the analysis of one of these systems, whose description can be characterized by a vector of d continuous parameters, noted $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{X} \subset \mathbb{R}^d$. We assume that there are uncertainties about these parameters, which may be epistemic or stochastic, and which can be dealt with by probability theory. Thus, the input vector is modeled by a random vector $\boldsymbol{X}$, and we assume that its distribution has a known probability density function (PDF) $f_{\boldsymbol{X}}$ with respect to the Lebesgue's measure. On the other hand, we note y the measurable function such that the system's failure domain can be characterized by $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{X} \mid y(\boldsymbol{x}) < 0\}$. By construction, the probability of system failure, denoted p_f, is given by:

$$p_f := \mathbb{P}_{\boldsymbol{X}}(y(\boldsymbol{X}) < 0) = \int_{\mathcal{F}} f_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}. \tag{1}$$

In this paper, function y is seen as the output of a numerically expensive deterministic "black box": for every $\boldsymbol{x}$ in $\mathbb{X}$, the value of $y(\boldsymbol{x})$ is unique, and it can be calculated using a simulator that can take a long time to evaluate. In this type of configuration, the calculation

---

Email address: `guillaume.perrin@univ-eiffel.fr` (G. Perrin)

of p_f is usually based on sampling techniques. For instance, if $\{\boldsymbol{X}(\omega_1), \ldots, \boldsymbol{X}(\omega_\ell)\}$ denote ℓ independent realizations of $\boldsymbol{X}$, the Monte Carlo (MC) approach [26, 1] approximates p_f by its empirical estimate, $\ell^{-1} \sum_{i=1}^{\ell} 1_{y(\boldsymbol{X}(\omega_i))<0}$, where $1_{y(\boldsymbol{x})<0}$ is equal to 1 if $y(\boldsymbol{x}) < 0$ and zero otherwise. By construction, the squared value of the variation coefficient of this estimator is equal to $\ell^{-1}(1/p_f - 1)$. In other words, to guarantee an estimate of p_f with good precision, i.e. with a relatively low coefficient of variation, very large values of ℓ are needed when p_f is low (say $p_f < 10^{-3}$ for a rare event).

Over the last few decades, specific techniques have therefore been developed to estimate p_f from a limited number of code evaluations. First of all, the well-known first- or second-order reliability methods (FORM/SORM) directly approach the boundary of $\mathcal{F}$ in the form of a linear or polynomial second-order function [14, 24, 7, 20]. More advanced approximations of failure domains were also introduced, based on support vector machine (SVM) techniques [31, 29, 17] or on generalized least squares linear regression [30, 16] for instance. Nevertheless, given an approximation of $\mathcal{F}$, it is generally difficult to evaluate the difference between p_f and its approximation.

Methods based on reducing the variance of the estimator were also introduced. Among them, the importance sampling approach introduces another random vector in the Monte Carlo method, so that the mean of the estimator is still equal to p_f, but its variance is much smaller than that in the classical Monte Carlo case [26]. Alternatively, the splitting techniques [2, 8] divide the rare event into a series of less rare events, so that the variance of the aggregate estimator is much lower than that of the direct estimator. However, for these two techniques to be effective, practical problems must be solved: the choice of the random vector for importance sampling, and the conditional sampling for splitting techniques (see [3] for advice on the control of rare event probability estimators).

To bypass some of these problems, most techniques for estimating rare events are currently based on a surrogate model, i.e. an easy to calculate mathematical approximation of y. Among these surrogate models, the Gaussian process regression (GPR) method, or kriging, proposes to model y as a particular realization of a Gaussian stochastic process $Y \sim \mathrm{GP}(\mu, \Sigma)$, whose mean function μ and covariance function Σ are estimated on the basis of the responses of a Design of Experiments (DoE) computed using true function y [27, 19, 28, 13]. Under that formalism,

$$p_f = \mathbb{P}_{\boldsymbol{X}}\left(Y(\boldsymbol{X}) < 0 \mid Y = y\right), \tag{2}$$

which amounts at saying that p_f is a particular realization of the random variable $P_f^Y :=$ $\mathbb{P}_{\boldsymbol{X}}\left(Y(\boldsymbol{X}) < 0 \mid Y\right)$. Assuming that Y is a good approximation of y, p_f can then be approximated by the mean value $\widehat{p}_f$ of P_f^Y (or possibly by $\widetilde{p}_f := \mathbb{P}_{\boldsymbol{X}}\left(\mu(\boldsymbol{X}) < 0\right)$), which can be rewritten as:

$$\widehat{p}_f = \mathbb{E}_Y\left[P_f^Y\right] = \mathbb{E}_{\boldsymbol{X}}\left[\Phi\left(-\frac{\mu(\boldsymbol{X})}{\sqrt{\Sigma(\boldsymbol{X},\boldsymbol{X})}}\right)\right], \quad \Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{v^2}{2}\right) dv. \quad (3)$$

Sampling techniques can finally be used to estimate $\widehat{p}_f$ (or $\widetilde{p}_f$) without requiring any additional evaluation of expensive function y.

A key aspect in this estimation of p_f is the optimization of the DoE used to estimate the statistical properties of Y. When interested in estimating a failure probability, it is important to notice that Y is only used to predict if $y(\boldsymbol{x})$ is smaller or greater than 0. Thus, Y has to be particularly precise in areas where y is close to 0. Depending on the sampling method considered for the estimation of $\widehat{p}_f$ (or $\widetilde{p}_f$), this motivated the introduction of several adaptive approaches for the optimization and the enrichment of this DoE (see for instance [11, 23, 6, 25, 12] for more details on the coupling of GPR and Monte Carlo approach, and [18, 5, 36, 35] for more details on the coupling of GPR and splitting techniques).

However, replacing true function y by an accurate surrogate can still lead to an inaccurate estimation of p_f [21]. Thus, to correctly anticipate the risks of deterioration of the system, it was proposed in [10] (and then completed in [34, 15, 37, 38]) to only use the surrogate model to approximate the Optimal Importance Density. In that case, the estimator of p_f is written as a product of two terms. The first one is a direct estimator of the failure probability based on the surrogate model only, and the second one is a factor correcting the bias, which relies on additional evaluations of true function y. Nevertheless, the numerical cost of estimating this second term may be relatively large, which may limit the use of these approaches when the number of calls allowed for y is very low.

Alternatively, in line with the works achieved in [33], we propose in this paper to work on the construction of confidence bounds to failure probability estimates. More precisely, instead of working on the estimation of the mean value of P_f^Y, which may lead to a prejudicial underestimation of p_f, we are interested in the construction of a robust estimator $\widehat{Q}_{\alpha,\beta}$ of the (1-α) quantile of P_f^Y, noted q_α, so that:

$$\mathbb{P}_Y(P_f^Y < q_\alpha) = 1 - \alpha, \quad (4)$$

$$\mathbb{P}_{\widehat{Q}_{\alpha,\beta}}\left(\mathbb{P}_Y\left(P_f^Y \leq \widehat{Q}_{\alpha,\beta} \mid \widehat{Q}_{\alpha,\beta}\right) \geq 1 - \alpha\right) \geq 1 - \beta, \quad (5)$$

where $\beta \in (0,1)$ is a chosen constant. By construction, estimator $\widehat{Q}_{\alpha,\beta}$ depends on two constants: α characterizes the risk associated to the replacement of y by its surrogate Y, and β controls the fact that only finite-dimensional samples of $Y(\boldsymbol{x})$ are available for its construction (see Figure 1 for a graphical illustration of the meanings of α, β, q_α and $\widehat{Q}_{\alpha,\beta}$). Therefore, for $\alpha, \beta \in (0,1)$ and a fixed number of evaluations of y, the first objective of
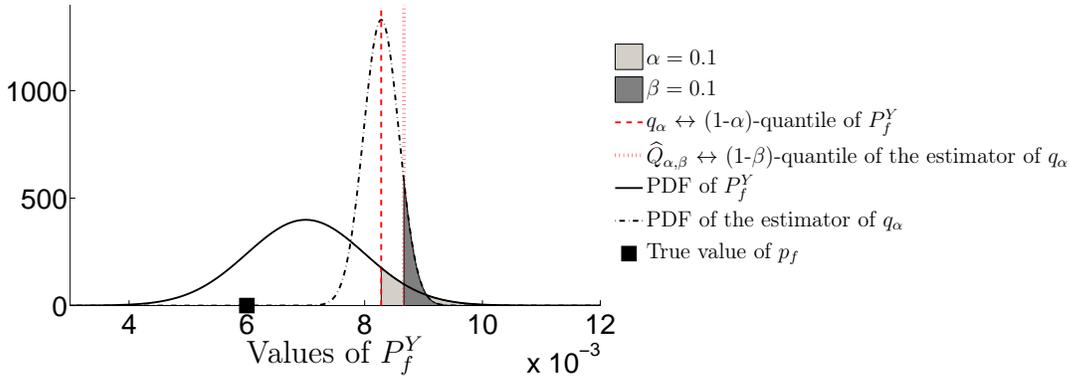
3

Figure 1: Graphical illustration of the meanings of α, β, q_α and $\widehat{Q}_{\alpha,\beta}$ on a synthetic example.

this paper is to propose an algorithm allowing us to construct this estimator. This algorithm couples results associated with (1) order statistics, (2) the Gaussian process regression formalism, and (3) a particular Marked Poisson Process.

The second objective of this paper is to propose a strategy adapted to the former algorithm to sequentially minimize the dependence of $\widehat{Q}_{\alpha,\beta}$ on the replacement of y by Y. In particular, we would like this enrichment to make it possible to manage two pathological configurations in rare events: the case where no point of the initial experimental design for the construction of Y belongs to the failure domain ; the case where the failure domain is multimodal.

The outline of this work is as follows. Section 2 presents the algorithm we propose for bounding the failure probability at a fixed budget. Then, Section 3 describes a stepwise uncertainty reduction procedure to sequentially improve the quality of the bound, and Section 4 presents numerical results. At last, Section 5 concludes the paper.

Notations

The notations that will be used in the rest of the paper are the following:

- a, b will correspond to deterministic scalars.

- $\boldsymbol{a}, \boldsymbol{b}$ will correspond to deterministic vectors.

- $(\Omega, \mathcal{A}, \mathbb{P})$ will correspond to a probability space.

- A, B will correspond to random variables.

- $\boldsymbol{A}, \boldsymbol{B}$ will correspond to random vectors.

- For all ω, ω' in Ω, $A(\omega)$ and $\boldsymbol{B}(\omega')$ will be particular realizations of A and $\boldsymbol{B}$.

4

- $\mathrm{GP}(\mu, \Sigma)$ will correspond to the distribution of a Gaussian process whose mean function is μ, and whose covariance function is Σ.

2. Bounding the failure probability at a fixed budget

The objective of this section is to propose an algorithm for constructing an upper bound with controlled precision for the probability of failure p_f at a fixed and limited budget. By fixed and limited budget, we mean that the maximum information for constructing this bound is a finite number ℓ of model evaluations, such that $\ell \ll 1/p_f$. In that case, direct approaches such as the Monte Carlo method cannot be applied, and indirect approaches based on the introduction of a surrogate model must generally be introduced. In that prospect, the formalism of the Gaussian process regression is considered in this work: y is seen as a sample path of a stochastic process defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, which is assumed Gaussian for the sake of tractability. By conditioning this Gaussian process by the ℓ available code evaluations, it is therefore possible to define very interesting predictors for the value of y in any non-observed point of $\mathbb{X}$. This predictor of function y at any $\boldsymbol{x}$ in $\mathbb{X}$, which is written $Y(\boldsymbol{x})$, is Gaussian by construction,

$$Y \sim \mathrm{GP}(\mu, \Sigma),$$

and we refer to [27, 28] for further details about the expressions of the conditioned mean, μ, and of the conditioned covariance function, Σ.

As explained in Introduction, under that formalism, p_f is a particular realization of the random variable $P_f^Y = \mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}) < 0 \mid Y)$. As a consequence, constructing a bound for p_f with confidence α amounts at computing the quantile of level $1-\alpha$ of P_f^Y. As the distribution of P_f^Y is non explicit, this quantile estimation is not easy, and generally requires having access to a set of independent realizations of P_f^Y. However, in our case, each realization of P_f^Y is defined by a failure probability associated with a particular realization of random process Y. Consequently, only estimates of these realizations are available in practice, which introduces another source of uncertainty that needs to be well controlled for p_f to be bounded with the desired precision. But the construction of these estimates is a problem in itself, because the projection of a particular realization of Y into a large number of input points is often made impossible for computational reasons.

Addressing these two issues is the subject of the two following subsections. Assuming that is is possible to construct an estimator for each realization of P_f^Y, section 2.1 will first show how order statistics can be used to construct an upper bound for p_f with controlled precision. Section 2.2 will then explain how the Poisson process theory can help us to construct these estimators from the projection of Y into a small number of input points.

2.1. Order statistics
For $\omega \in \Omega$, let $Y(\omega)$ be a realization of Y, $Y(\boldsymbol{x}; \omega)$ be the projection of $Y(\omega)$ in any input vector $\boldsymbol{x}$ in $\mathbb{X}$, and $P_f^Y(\omega) := \mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}; \omega) < 0)$ be the associated realization of P_f^Y.

In addition, let $\mathcal{X}^n$ be a random set of $n > 1$ points chosen (independently or not) in $\mathbb{X}$ and $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$ be an estimator of $P_f^Y(\omega)$ relying on the elements of $\mathcal{X}^n$. For instance, $\mathcal{X}^n$ can be chosen equal to $\{\boldsymbol{X}^{(i)},\ 1 \leq i \leq n\}$, where $\boldsymbol{X}^{(1)},\ldots,\boldsymbol{X}^{(n)}$ are n independent copies of $\boldsymbol{X}$, and $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$ can be defined by :

$$\widehat{P}_f^{Y,\mathcal{X}^n}(\omega) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{Y(\boldsymbol{X}^{(i)};\omega)<0}. \tag{6}$$

For $\theta \in \Omega$, this allows us to define $\widehat{P}_f^{Y,\mathcal{X}^n}$ as the random variable associated with Y and $\mathcal{X}^n$, so that for each realization $Y(\omega)$ of Y and each realization $\mathcal{X}^n(\theta)$ of $\mathcal{X}^n$, $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega,\theta)$ is a realization of $\widehat{P}_f^{Y,\mathcal{X}^n}$.

For $m > 1$, we now introduce m independent copies of Gaussian process Y, noted $Y_1,\ldots,Y_m$, and m independent copies of random set $\mathcal{X}^n$, written $\mathcal{X}_1^n,\ldots,\mathcal{X}_m^n$. For $1 \leq j \leq m$, we then write $\widehat{P}_j := \widehat{P}_f^{Y_j,\mathcal{X}_j^n}$ the associated independent random variables. These random variables are supposed to be sorted in ascending order:

$$\widehat{P}_1 \leq \widehat{P}_2 \leq \ldots \leq \widehat{P}_m. \tag{7}$$

From basic statistics, for $1 \leq j \leq m$ and $\alpha \in (0,1)$, we therefore have:

$$\mathbb{P}(\widehat{P}_j > q_\alpha) = \sum_{u=0}^{j-1}\binom{m}{u}(1-\gamma)^{m-u}\gamma^u, \tag{8}$$

where $\gamma := \mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq q_\alpha)$. The value of q_α, which is defined by Eq.(4), is unfortunately unknown, which makes it difficult to estimate γ. However, reminding that by definition of q_α, $\mathbb{P}(P_f^Y \leq q_\alpha)$ is equal to $1-\alpha$, it is possible to bound the value of γ by a quantity $\gamma_\star$ that depends on the conditioned probability $\mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq P_f^Y \mid P_f^Y \geq q_\alpha)$:

$$\begin{aligned}
\gamma &= \mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq q_\alpha \mid P_f^Y \leq q_\alpha)\mathbb{P}(P_f^Y \leq q_\alpha) + \mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq q_\alpha \leq P_f^Y) \\
&\leq 1 \times (1-\alpha) + \mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq P_f^Y \ \cap \ P_f^Y \geq q_\alpha) \\
&\leq 1 - \alpha(1 - \mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq P_f^Y \mid P_f^Y \geq q_\alpha)) =: \gamma_\star.
\end{aligned} \tag{9}$$

As q_α is still unknown, $\gamma_\star$ is also unknown in the general case. But depending on the choice of the estimator for $P_f^Y(\omega)$, asymptotic values can be proposed for $\gamma_\star$. For example, if we consider again for $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$ the Monte Carlo estimator provided by Eq. (6), the central limit theorem allows us to write that:

$$\sqrt{n}(\widehat{P}_f^{Y,\mathcal{X}_j^n}(\omega) - P_f^Y(\omega)) \overset{\mathcal{L}}{\rightarrow} \mathcal{N}\left(0, P_f^Y(\omega)(1 - P_f^Y(\omega))\right),\ \omega \in \Omega. \tag{10}$$

Hence, for this choice of estimator, $\mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}^n} \leq P_f^Y \mid P_f^Y \geq q_\alpha)$ becomes close to $1/2$ when n is high enough, and constant $\gamma_\star$ can be replaced by $1 - \alpha/2$.

Assuming that the value of $\gamma_\star$ is known, and noticing that $\gamma \mapsto \sum_{u=0}^{j-1} \binom{m}{u} (1-\gamma)^{m-u} \gamma^u$ is a decreasing function, we can now write:

$$\mathbb{P}(\widehat{P}_j > q_\alpha) \geq \sum_{u=0}^{j-1} \binom{m}{u} (1-\gamma_\star)^{m-u} \gamma_\star^u. \tag{11}$$

Hence, by choosing $j^\star(\alpha,\beta)$ as the minimal index such that $\sum_{u=0}^{j^\star(\alpha,\beta)-1} \binom{m}{u}(1-\gamma_\star)^{m-u}\gamma_\star^u \geq 1-\beta$, it can be seen that $\mathbb{P}(\widehat{P}_f^{Y_{j^\star(\alpha,\beta)},\mathcal{X}_{j^\star(\alpha,\beta)}^n} > q_\alpha) \geq 1-\beta$. Using Eq.(4), it comes:

$$\mathbb{P}_{\widehat{P}_f^{Y_{j^\star(\alpha,\beta)},\mathcal{X}_{j^\star(\alpha,\beta)}^n}} \left(\mathbb{P}_Y \left(P_f^Y \leq \widehat{P}_f^{Y_{j^\star(\alpha,\beta)},\mathcal{X}_{j^\star(\alpha,\beta)}^n} \mid \widehat{P}_f^{Y_{j^\star(\alpha,\beta)},\mathcal{X}_{j^\star(\alpha,\beta)}^n} \right) \geq 1-\alpha \right) \geq 1-\beta. \tag{12}$$

Thus, replacing $\widehat{P}_f^{Y_{j^\star(\alpha,\beta)},\mathcal{X}_{j^\star(\alpha,\beta)}^n}$ by $\widehat{Q}_{\alpha,\beta}$ in Eq.(12), we eventually obtain the searched result of Eq.(5).

2.2. Choice of the estimator

In the former section, the result of Eq.(12) has been illustrated on the case where $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$ corresponds to the classical Monte Carlo estimator of $P_f^Y(\omega)$. In that case, for any $\omega \in \Omega$, as the squared value of the variation coefficient of $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$ is proportional to $1/(P_f^Y(\omega)n)$, high values of n can be required for this estimator to be reasonable when considering low values of $P_f^Y(\omega)$. In practice, this would mean that to numerically calculate $\widehat{P}_f^{Y,\mathcal{X}^n}(\omega)$, we need to project $Y(\omega)$ in a very high number of points (between $100/P_f^Y(\omega)$ and $1000/P_f^Y(\omega)$ for instance), which is often not possible due to computational reasons (memory and conditioning problems).

It is then necessary to propose a new estimator for each realization of P_f^Y that would be based on the projection of each particular sample path of Y into a much smaller number of input vectors. For this purpose, following the works achieved in [32], it is interesting to notice (see Appendix for the proofs) that for $q > 1$, if $P_1, \dots, P_q$ are q independent copies of a Poisson process of parameter $-\log\left(\mathbb{P}_X(Y(\boldsymbol{X};\omega) < 0)\right)$,

$$\widehat{P}_f^{Y,\mathcal{X}_n}(\omega) := \left(1 - \frac{1}{q}\right)^{\sum_{k=1}^q P_k} \tag{13}$$

defines an estimator of $P_f^Y(\omega) = \mathbb{P}_X(Y(\boldsymbol{X};\omega) < 0)$, and it verifies

$$\mathbb{E}_{\mathcal{X}_n}\left[\widehat{P}_f^{Y,\mathcal{X}_n}(\omega)\right] = P_f^Y(\omega), \quad \mathrm{Var}_{\mathcal{X}_n}(\widehat{P}_f^{Y,\mathcal{X}_n}(\omega)) = P_f^Y(\omega)^2 (P_f^Y(\omega)^{-1/q} - 1). \tag{14}$$

Moreover, the Poisson distribution is known to be well approximated by a Gaussian distribution as soon as its parameter is greater than 10. Thus, assuming that $P_f^Y(\omega) \leq 0.1$, once $-q\log(P_f^Y(\omega)) > 10$, that is to say once $q \geq 5$, it is reasonable to assume that $\sum_{k=1}^q P_k$ follows a Gaussian distribution, from which we deduce that:

7

$$\log \left(\widehat{P}_f^{Y,\mathcal{X}}(\omega) \right) \sim_{q>5} \mathcal{N} \left(\begin{array}{c} -q \log(P_f^Y(\omega)) \log(1 - 1/q), \\ -q \log(P_f^Y(\omega)) \log(1 - 1/q)^2 \end{array} \right), \ \omega \in \Omega. \tag{15}$$

In the same manner than for the former Monte Carlo estimator, this means that $\mathbb{P}(\widehat{P}_f^{Y,\mathcal{X}_n} \leq P_f^Y \mid P_f^Y \geq q_\alpha)$ becomes close to $1/2$ when q is high enough, leading to a value of $\gamma_\star$ close to $1 - \alpha/2$. However, while $Y(\omega)$ was to be projected in approximately $100/P_f^Y(\omega)$ points for the Monte Carlo approach to give relevant results, it only needs to be projected in $\mathbb{E}\left[\sum_{k=1}^q P_k\right] = -q \log(P_f^Y(\omega))$ points in average, which is clearly much more accessible from a computational point of view.

2.3. Practical construction of the Poisson-based estimator

For any $t > 0$, let $P(t)$ be a Poisson process of parameter t. Hence, to construct the estimator presented in Eq. (13), we need to be able to generate realizations of a Poisson process of parameter $-\log\left(\mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X};\omega) < 0)\right)$. The result of Proposition 1 gives us a way to achieve this goal in a very natural way.

Proposition 1. *For any ω in Ω,*

$$P(-\log\left(\mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X};\omega) < 0)\right)) = \sup\{i; \ Z_i \geq 0\}, \tag{16}$$

where $(Z_i)_{i\geq 0}$ is the decreasing random walk so that for all $i \geq 1$ and $z \in \mathbb{R}$:

$$Z_0 = +\infty, \ \mathbb{P}(Z_{i+1} \leq z \mid Z_i) = \mathbb{P}(Z \leq z \mid Z \leq Z_i), \tag{17}$$

with $Z := Y(\boldsymbol{X};\omega)$.

In order to generate realizations of the estimator presented in Eq. (13), it is thus sufficient to launch in parallel on the same instance of the random process Y several draws of the random walk defined by Eq. (17). If we combine this generation method with the results of Section 2.1 concerning the order statistics, we obtain a procedure for constructing a particular realization of $\widehat{Q}_{\alpha,\beta}$, which is summarized in Algorithm 1. This algorithm is based on four constants : $\alpha, \beta \in (0,1)$ are chosen according to Eq.(12) to control the risk of underestimation of probability of interest p_f, ℓ is the dimension of the initial design of experiments for the construction of the GPR-based surrogate model of y, and q is the number of copies of the decreasing random walk. In the following, α and β will be fixed to 0.1. Noticing that

$$\sum_{u=0}^{45-1} \binom{45}{45-u}(1 - (1-0.1/2))^{45-u}(1-0.1/2)^u \approx 0.90056,$$

we then choose $m = j^\star(\alpha,\beta) = 45$ to minimize the number of tested sample paths. ℓ will be adapted to the dimension of the input space $\mathbb{X}$, and we propose to gradually increase q until

8

the maximal value of the coefficients of variation of the m estimators is less than a given constant c (in the application section, this leads to values of q between 100 and 1000).

Algorithm 1: Construction of a particular realization of $\widehat{Q}_{\alpha,\beta}$.

Choose $\alpha, \beta, \ell, q, n_{\max}$;
Compute $m = j^\star(\alpha, \beta)$;
Let $Y \sim \mathrm{GP}(\mu, \Sigma)$ be the GPR-based surrogate model associated with y based on ℓ evaluations of y ;
for $1 \leq j \leq m$ **do**
    Sample q independent realizations of $\boldsymbol{X}$, noted $\boldsymbol{X}(\omega_1), \ldots, \boldsymbol{X}(\omega_q)$;
    Sample one realization of the Gaussian vector $(Y(\boldsymbol{X}(\omega_1)), \ldots, Y(\boldsymbol{X}(\omega_q)))$, noted $(y_1, \ldots, y_q)$;
    Define $Y_j(\omega) := Y \mid Y(\boldsymbol{X}(\omega_k)) = y_k, \ 1 \leq k \leq q$;
    Initialize : $n_{\mathrm{iter}} = 0$, $\widehat{\mathcal{X}}^j = \{\boldsymbol{X}(\omega_1), \ldots, \boldsymbol{X}(\omega_q)\}$, $\widehat{\mathcal{Y}}^j = \{y_1, \ldots, y_q\}$;
    for $1 \leq k \leq q$ **do**
        $z = y_k, P_k^j = 0$;
        while $z > 0$ and $n_{iter} \leq n_{max}$ **do**
            increment $n_{\mathrm{iter}} = n_{\mathrm{iter}} + 1$;
            draw at random a realization of $\boldsymbol{X}$, denoted by $\boldsymbol{x}^\star$;
            draw at random a realization of $Y_j(\boldsymbol{x}^\star)$, denoted by $y^\star$;
            if $y^\star < z$ **then**
                $z = y^\star, P_k^j = P_k^j + 1$;
                $Y_j(\omega) = Y_j(\omega) \mid Y_j(\boldsymbol{x}^\star) = y^\star$;
                $\widehat{\mathcal{X}}^j = \widehat{\mathcal{X}}^j \cup \{\boldsymbol{x}^\star\}, \widehat{\mathcal{Y}}^j = \widehat{\mathcal{Y}}^j \cup \{y^\star\}$;
            end
        end
    end
    $\widetilde{p}_j := \left(1 - \frac{1}{q}\right)^{\sum_{k=1}^{q} P_k^j}$;
end
Return $\max_{1 \leq j \leq m} \widetilde{p}_j$.

Remarks on Algorithm 1.

- The m realizations $\widetilde{p}_j$ being independent, their construction can be made completely parallel.

- As we are only interested in the maximum of $\widetilde{p}_1, \ldots, \widetilde{p}_m$, once the loop j associated with Y_j has converged, it is possible to break each other loop j' as soon as $\sum_{k=1}^{q} P_k^{j'}$ is greater than $\sum_{k=1}^{q} P_k^j$. This can allow us to drastically reduce the total computational time associated with the construction of a realization of $\widehat{Q}_{\alpha,\beta}$.

- For the sake of simplicity, we propose in Algorithm 1 to directly generate the new candidates for $\boldsymbol{X}$ according to its density $f_{\boldsymbol{X}}$. When considering very small probability
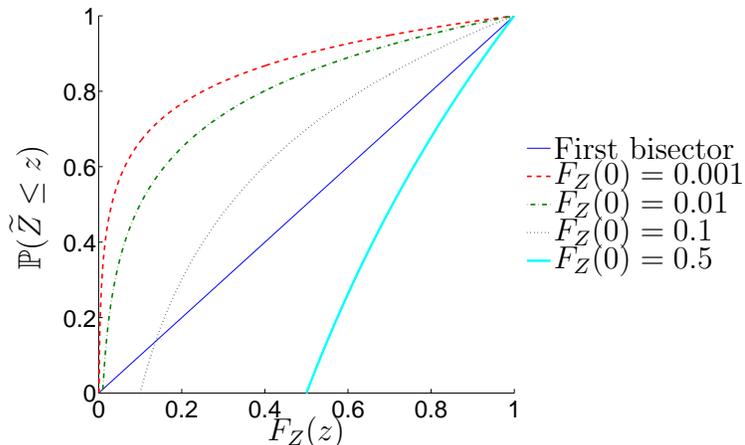
Figure 2: Concentration of the points generated by Algorithm 1 on the low values of y (with $n_{\max} = +\infty$).

of system failure, this can result in a very long execution time for the algorithm. In order to accelerate its convergence, it can be necessary to use sequential Monte Carlo simulation techniques instead [22].

- If the ℓ available values of y are strictly positive, it may very well happen that $\mathbb{E}_Y[P_f^Y]$ is much smaller than p_f. In this case, the risk of having a very (even too) high execution time for the presented algorithm is also strong. This explains the presence of the maximum number of iterations $n_{\max}$. By stopping a loop of the algorithm before its convergence, we obtain a value of P_k which will be smaller than what it should have been, leading to an overestimation of $\widehat{Q}_{\alpha,\beta}$, i.e. to a more protective bound for p_f.

2.4. Distribution of the generated points

In addition to the realization of $\widehat{Q}_{\alpha,\beta}$, Algorithm 1 provides a series of sets noted $\left(\widehat{\mathcal{X}}^j, \widehat{\mathcal{Y}}^j\right)_{j=1}^m$. For each j, the elements of $\widehat{\mathcal{Y}}^j$ are particular projections of Y in the elements of $\widehat{\mathcal{X}}^j$. Looking at Property 2, Corollary 1, and Figure 2, it is interesting to notice that the smaller p_f, the more concentrated around 0 the elements of $\widehat{\mathcal{Y}}^j$ are likely to be. In other words, for small values of p_f, the elements of $\widehat{\mathcal{X}}^j$ will be concentrated in areas where there is a high probability of obtaining a low value for Y. Hence, the elements of $\widehat{\mathcal{X}}^j$ will naturally show the regions of input space $\mathbb{X}$ that need to be particularly well covered by evaluations of y, in order to minimize the impact of replacing y by Y. As this consideration will play a central role in the next section, we denote by $\mathcal{L}(\mathbb{X})$ the (non-trivial) distribution over the input space of any element of these sets $\widehat{\mathcal{X}}^j$.

Proposition 2. *Let Z be a real-valued random variable with continuous CDF F_Z and $(Z_i)_{i\geq0}$ be the decreasing random walk defined by Eq.(17). If $\widetilde{Z}$ is a positive element of $(Z_i)_{i\geq0}$ chosen at random,*

10

$$\mathbb{P}(\widetilde{Z} \leq z) = 1 - \frac{\log(F_Z(z))}{\log(F_Z(0))}, \ z \geq 0. \tag{18}$$

Corollary 1. *For each $1 \leq j \leq m$, let $\widehat{y}$ be a point randomly chosen in $\widehat{\mathcal{Y}}^j$ such that $\widehat{y} \geq 0$. In the case $n_{max} = +\infty$, we have :*

$$\mathbb{P}(\widehat{y} \leq z) = 1 - \frac{\log\left(\mathbb{P}_{\boldsymbol{X}}(Y_j(\boldsymbol{X};\omega) \leq z)\right)}{\log\left(P_f^{Y_j}(\omega)\right)}. \tag{19}$$

3. Stepwise uncertainty reduction

Estimator $\widehat{Q}_{\alpha,\beta}$ aggregates at least two sources of uncertainty: the variability in the sample paths of Y due to the substitution of the computer code by its GPR-based surrogate model, and the variability in the different copies of the decreasing random walk associated with each sample path. While it is possible to control the second source of uncertainty by increasing the value of q, new code evaluations are needed to decrease the first source of uncertainty. In that prospect, different strategies can be found in the literature to sequentially choose the positions of the new evaluation points in order to minimize this first source of uncertainty. For instance, following [11], the new point can be chosen as the solution of the following optimization problem:

$$\boldsymbol{x}^{\star} := \arg\min_{\boldsymbol{x} \in \mathcal{S}(n)} \frac{|\mu(\boldsymbol{x})|}{\sqrt{\Sigma(\boldsymbol{x}, \boldsymbol{x})}}, \tag{20}$$

where $\mathcal{S}(n)$ gathers $n \gg 1$ iid samples of $\boldsymbol{X}$. However, it can be noticed that such a pointwise strategy does not take into account in its selection criteria the fact that the new evaluation point will bring additional information in its neighbourhood. In contrast, Stepwise Uncertainty Reduction (SUR) techniques propose to select the new evaluation point that minimizes the expected value of a well chosen measure of the uncertainty. For instance, noticing that the random variable $1_{Y(\boldsymbol{X})<0}$ has conditional variance $\alpha(\boldsymbol{X})(1 - \alpha(\boldsymbol{X}))$ with

$$\alpha(\boldsymbol{X}) := \mathbb{E}_Y\left[1_{Y(\boldsymbol{X})<0} \mid \boldsymbol{X}\right] = \Phi\left(\frac{-\mu(\boldsymbol{X})}{\sqrt{\Sigma(\boldsymbol{X}, \boldsymbol{X})}}\right), \tag{21}$$

the quantity $\mathbb{E}_{\boldsymbol{X}}[\alpha(\boldsymbol{X})(1 - \alpha(\boldsymbol{X}))]$ characterizes the uncertainty due to the replacement of y by its GPR-based surrogate model Y. Here, function Φ is the CDF of the standard Gaussian random variable introduced in Eq.(3). Hence, as proposed in [4], the new point $\boldsymbol{x}^{\star}$ can be searched as:

$$\boldsymbol{x}^{\star} = \arg\min_{\boldsymbol{x} \in \mathcal{S}(n)} \mathbb{E}_{\xi(\boldsymbol{x})}\left[\mathbb{E}_{\boldsymbol{X}}\left[\alpha^c(\boldsymbol{X}, \xi(\boldsymbol{x}))(1 - \alpha^c(\boldsymbol{X}, \xi(\boldsymbol{x}))) \mid \xi(\boldsymbol{x})\right]\right], \tag{22}$$

$$\alpha^c(\boldsymbol{X}, \xi(\boldsymbol{x})) := \mathbb{E}_Y\left[1_{Y(\boldsymbol{X})<0} \mid Y(\boldsymbol{x}) = \xi(\boldsymbol{x}), \xi(\boldsymbol{x}), \boldsymbol{X}\right],$$

with $\xi(\boldsymbol{x})$ a Gaussian random variable with mean $\mu(\boldsymbol{x})$ and variance $\Sigma(\boldsymbol{x}, \boldsymbol{x})$. In the following, we will refer to the selection criteria associated with Eqs. (20) and (22) as "Pointwise criterion" and "SUR criterion" respectively.

These two criteria have been and continue to be of great use in the sequential improvement of a GPR-based surrogate model. However, they are not particularly suited to the Poisson process that we use for the estimation of $\widehat{Q}_{\alpha,\beta}$. Indeed, they do not really take into account a central aspect of the proposed procedure, which is the ability of Gaussian process Y to correctly sort different values of the input vectors from their respective values of true function y. To better stick to this characteristic of Algorithm 1, we propose to search $\boldsymbol{x}^{\star}$ as the point that will minimize the risk of misclassification. That is the point such that the probability to have at the same time $Y(\boldsymbol{x}) > Y(\boldsymbol{x}')$ and $\mu(\boldsymbol{x}) < \mu(\boldsymbol{x}')$ is minimum for any value of $\boldsymbol{x}$ and $\boldsymbol{x}'$. And to focus the criterion on the sub-domain of $\mathbb{X}$ mainly explored by Algorithm 1, it is proposed that $\boldsymbol{x}$ and $\boldsymbol{x}'$ be drawn according to the distribution $\mathcal{L}(\mathbb{X})$ defined in Section 2.4:

$$\boldsymbol{x}^{\star} = \arg \min_{\widehat{\boldsymbol{x}} \in \mathcal{S}(n)} \mathbb{P}_{Y,\boldsymbol{X},\boldsymbol{X}'}\left(Y(\boldsymbol{X}) > Y(\boldsymbol{X}'), \mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \mid Y(\widehat{\boldsymbol{x}}) = \mu(\widehat{\boldsymbol{x}})\right), \quad \boldsymbol{X}, \boldsymbol{X}' \sim \mathcal{L}(\mathbb{X}).$$
(23)

In other words, by introducing this new criterion that will be referred to as "Proposed criterion" in the following, the objective is to give priority to the points that are difficult to classify in the high probability areas for the draws of the Poisson process under consideration. And by looking at the misclassification all along the sample path of the decreasing random walk, we also would like to avoid an over-sensitivity of the results to a poor initial construction of the surrogate model.

Proposition 3. *For all $\widehat{\boldsymbol{x}}$ in $\mathbb{X}$,*

$$\mathbb{P}_{Y,\boldsymbol{X},\boldsymbol{X}'}\left(Y(\boldsymbol{X}) > Y(\boldsymbol{X}'), \mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \mid Y(\widehat{\boldsymbol{x}}) = \mu(\widehat{\boldsymbol{x}})\right)$$
$$= \frac{1}{2} \mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[\Phi\left(-\frac{|\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}')|}{s(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}})}\right)\right],$$
(24)

with:

$$s^2(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}}) := \Sigma(\boldsymbol{X}, \boldsymbol{X}) + \Sigma(\boldsymbol{X}', \boldsymbol{X}') - 2\Sigma(\boldsymbol{X}, \boldsymbol{X}') - \frac{(\Sigma(\boldsymbol{X}, \widehat{\boldsymbol{x}}) - \Sigma(\boldsymbol{X}', \widehat{\boldsymbol{x}}))^2}{\Sigma(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}})}.$$
(25)

Another important feature of the proposed criterion is that it can be seen as a direct post-processing of the points generated by Algorithm 1 at previous step, when the resolution of the SUR problem defined by Eq. (22) can be numerically costly (see [9] for more details on how to solve this optimization problem in practice). Indeed, using Proposition 3, if $\widehat{\mathcal{X}} := \bigcup_{j=1}^{m} \widehat{\mathcal{X}}^j$ gathers all the generated input points, new point $\boldsymbol{x}^{\star}$ can be searched as:

$$\boldsymbol{x}^{\star} = \arg\min_{\widehat{\boldsymbol{x}} \in \widehat{\mathcal{X}}} \sum_{\substack{\boldsymbol{x}, \boldsymbol{x}' \in \widehat{\mathcal{X}} \setminus \widehat{\boldsymbol{x}} \\ \boldsymbol{x} \neq \boldsymbol{x}'}} \Phi\left(-\frac{|\mu(\boldsymbol{x}) - \mu(\boldsymbol{x}')|}{s(\boldsymbol{x}, \boldsymbol{x}', \widehat{\boldsymbol{x}})}\right). \tag{26}$$

Finally, the procedure we propose for the sequential estimation of the failure probability bound $\widehat{Q}_{\alpha,\beta}$ is summarized in Algorithm 2. Here, the stopping criterion is a maximum number of code evaluations, which is a constraint that is often present when studying costly simulators. Alternative criteria could have been proposed, based on convergence analyses of the values of $\widehat{Q}_{\alpha,\beta}$ for instance.

Algorithm 2: Sequential estimation of the failure probability bound $\widehat{Q}_{\alpha,\beta}$.

Choose $\alpha, \beta, \ell, q, n_{\max}, n_{\mathrm{add}}$ and compute $m = j^{\star}(\alpha, \beta)$;
Let $Y \sim \mathrm{GP}(\mu, \Sigma)$ be the GPR-based surrogate model associated with y based on ℓ evaluations of y ;
for $1 \leq r \leq n_{add}$ **do**
  Run Algorithm 1, recover $q_r := \widehat{Q}_{\alpha,\beta}$ and $\widehat{\mathcal{X}} = \bigcup_{j=1}^{m} \widehat{\mathcal{X}}^{j}$;
  Solve optimization problem 26, recover $\boldsymbol{x}^{\star}$;
  Update the surrogate model : $Y = Y \mid Y(\boldsymbol{x}^{\star}) = y(\boldsymbol{x}^{\star})$;
end
Return $(q_1, \ldots, q_{n_{\mathrm{add}}})$.

Remark. The expected number of elements of $\widehat{\mathcal{X}}$ is $-mq\log(p_f)$. For $p_f = 10^{-6}$, $m = 45$ and $q = 100$ (which approximately corresponds to the values considered in the application section), this gives around 6.2×10^4 elements. To solve Eq. (26), this means that we need to evaluate function Φ in around $\frac{1}{2}(6.2 \times 10^4)^3 \approx 1.2 \times 10^{14}$ points, which is clearly prohibitive. In order to solve this problem for a more reasonable cost, we propose to consider only a subset of potential pairs for $\boldsymbol{x}$ and $\boldsymbol{x}'$, which leads to the simplified optimization problem:

$$\boldsymbol{x}^{\star} = \arg\min_{\widehat{\boldsymbol{x}} \in \widehat{\mathcal{X}}} \sum_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}(n; \boldsymbol{x}^{\star})} \Phi\left(-\frac{|\mu(\boldsymbol{x}) - \mu(\boldsymbol{x}')|}{s(\boldsymbol{x}, \boldsymbol{x}', \widehat{\boldsymbol{x}})}\right), \tag{27}$$

where $\mathcal{S}(n; \boldsymbol{x}^{\star})$ gathers n distinct pairs of elements randomly chosen in $\widehat{\mathcal{X}} \setminus \widehat{\boldsymbol{x}}$. In the application section that follows, this value of n will be chosen equal to 10^3.

4. Applications

We list at least four objectives for the application section. First, we would like to check on difficult numerical cases that $\widehat{Q}_{\alpha,\beta}$ is higher than failure probability p_f with high probability, when $\widehat{p}_f = \mathbb{E}_Y\left[P_f^Y\right] = \mathbb{E}_{\boldsymbol{X}}\left[\Phi\left(-\frac{\mu(\boldsymbol{X})}{\sqrt{\Sigma(\boldsymbol{X},\boldsymbol{X})}}\right)\right]$ may strongly underestimate it. To this end, we also introduce the following estimation of $\widehat{p}_f$:

13

Example	Function y	d	p_f	ℓ	n_{add}
1	Oscillating function	1	0.0109	10	12
2	Branin	2	0.00196	20	100
3	Hartmann	6	0.000949	300	100

Table 1: Characteristics of the three analyzed numerical functions (see Appendix for the expressions of the functions).

$$\widehat{m} := \frac{1}{n^{\mathrm{test}}} \sum_{i=1}^{n^{\mathrm{test}}} \Phi\left(-\frac{\mu(\boldsymbol{X}(\omega_i))}{\sqrt{\Sigma(\boldsymbol{X}(\omega_i), \boldsymbol{X}(\omega_i))}}\right), \tag{28}$$

where $\boldsymbol{X}(\omega_1), \ldots, \boldsymbol{X}(\omega_{n^{\mathrm{test}}})$ denote $n^{\mathrm{test}} \gg 1$ iid realizations of $\boldsymbol{X}$ (n^{test} will be chosen equal to 10^6 in the following). The second objective is to compare the trade-offs between exploration and exploitation for the three introduced section criteria. Then, this section aims at quantifying the efficiency of the proposed selection criteria to make bound $\widehat{Q}_{\alpha,\beta}$ converge to the true value of p_f from above. At last, the proposed algorithms are applied to a study close to real engineering applications, for which the true value of p_f is however unknown, in order to better highlight the potential contributions of such an approach.

4.1. Analytical examples

4.1.1. Presentation of the test cases

The adaptive estimation of bound $\widehat{Q}_{\alpha,\beta}$ will be first illustrated on three numerical applications, whose characteristics are listed in Table 4.1.1. None of the presented examples will actually be costly to evaluate, which makes possible the performance analysis of the proposed algorithms. In particular, the reference values for p_f are estimated using a crude Monte Carlo approach based on 10^6 code evaluations.

On purpose, each function presents several local minima, and the value of ℓ is in each case chosen much smaller than $1/p_f$, so that no training point belongs to the failure domain at the initial step (with high probability). To get sound comparisons between the different selection criteria, the results presented in the next sections are averaged over 10 repetitions of the whole procedures.

4.1.2. Analysis of the results

The comparison results for the oscillating and the Branin functions are summarized in Figures 3 and 4. Focusing on the graphs of Figure 3, strong differences can be noticed between the results associated with the three selection criteria. For the oscillating function, when the pointwise strategy seems to focus on a unique region around the global minimum, a better balance between exploration and exploitation is observed for the SUR criterion, and even more for the proposed procedure, which adds points in priority in low value areas for y but also where there may be a high risk of misclassification. Similar observations can be made for
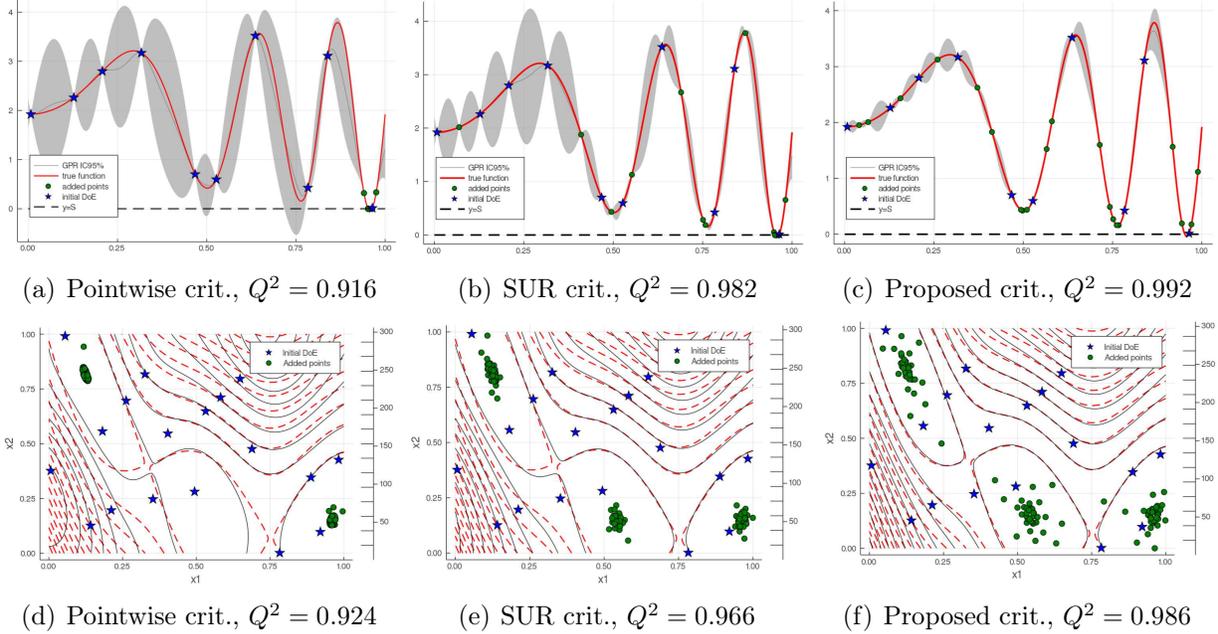
(a) Pointwise crit., $Q^2 = 0.916$ (b) SUR crit., $Q^2 = 0.982$ (c) Proposed crit., $Q^2 = 0.992$

(d) Pointwise crit., $Q^2 = 0.924$ (e) SUR crit., $Q^2 = 0.966$ (f) Proposed crit., $Q^2 = 0.986$

Figure 3: Comparison of the GPR-based surrogate models after adding n_{add} new code evaluations. In each subfigure, function y is represented in red (in solid line for the oscillating function and in dotted line for the Branin function), the blue stars indicate the positions of the ℓ initial training points, and the green circles indicate the positions of the new code evaluations. For the oscillating function, the grey area corresponds to the 95% prediction interval provided by the surrogate model, when the contour lines associated with the conditioned mean are in black solid line for the Branin function. Here, Q^2 is the classical metric of learning performance on test data calculated as one minus the predictive residual error sum of squares (PRESS) divided by the total sum of squares (TSS).

the Branin function. Depending on the initial design, only one or two out of the three regions of failure are found on average with the pointwise strategy, when these three regions are well covered by the SUR and the proposed selection criterion. And once again, we observe that the proposed criterion tends to spread out the evaluation points more widely in the input space in order to allow a better global classification.

The same kinds of observations can be made when looking in Figure 4 at the convergence of $\widehat{m}$ and $\widehat{Q}_{\alpha,\beta}$ with respect to n_{add}. Indeed, as the pointwise strategy is struggling to find all the minima, $\widehat{m}$ does not converge to p_f, while the values of $\widehat{Q}_{\alpha,\beta}$ remain very high. Different results are obtained for the SUR and the proposed selection criteria: $\widehat{m}$ converges to p_f and interesting values are quickly obtained for $\widehat{Q}_{\alpha,\beta}$. However, differences can also be noticed between these two approaches. As the SUR criterion explores less the input space to focus more on the contours of the failure domain, the convergence speed of $\widehat{m}$ is on average quicker. But the convergence of $\widehat{Q}_{\alpha,\beta}$ seems to be a bit quicker for the proposed criterion, which is not surprising, since this criterion was built specifically for this purpose.

The third example in dimension six allows us to emphasize the importance of exploration when analyzing multimodal functions. Indeed, for this example, we see in Figure 5 that
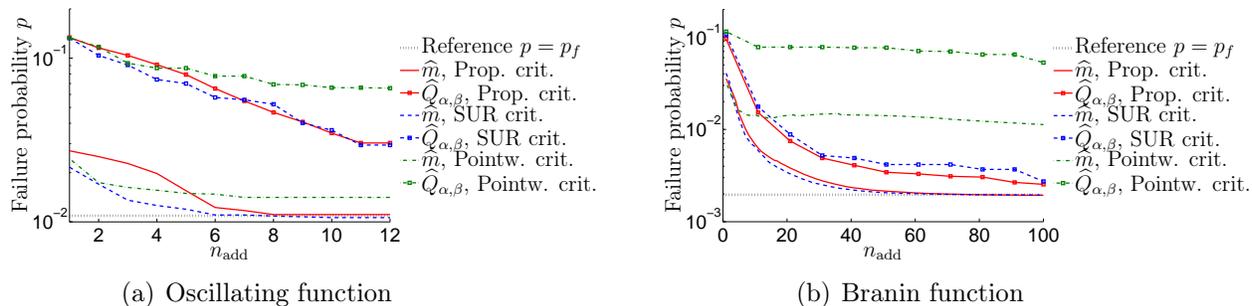
(a) Oscillating function

(b) Branin function

Figure 4: Convergence with respect to the number of added points n_{add} of the estimators $\widehat{m}$ and $\widehat{Q}_{\alpha,\beta}$ depending on the chosen selection criterion.
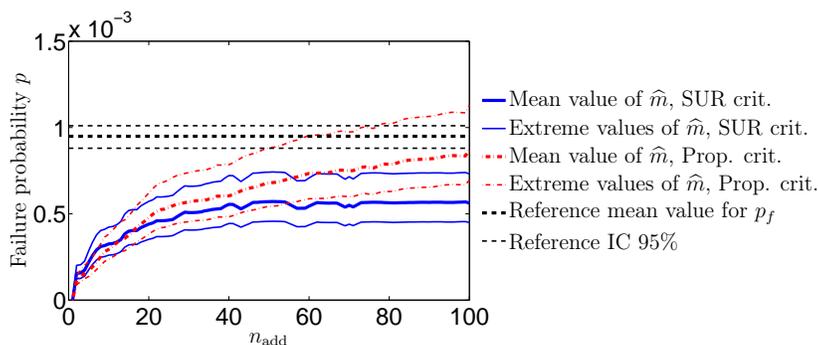

Figure 5: Convergence with respect to the number of added points n_{add} of the estimator $\widehat{m}$ depending on the chosen selection criterion. For the reference value estimated with a Monte Carlo approach, IC 95% stands for 95% confidence interval. For the two compared criteria, the mean and the extreme (minimal and maximum) values are associated with 10 repetitions of the whole sequential procedure.
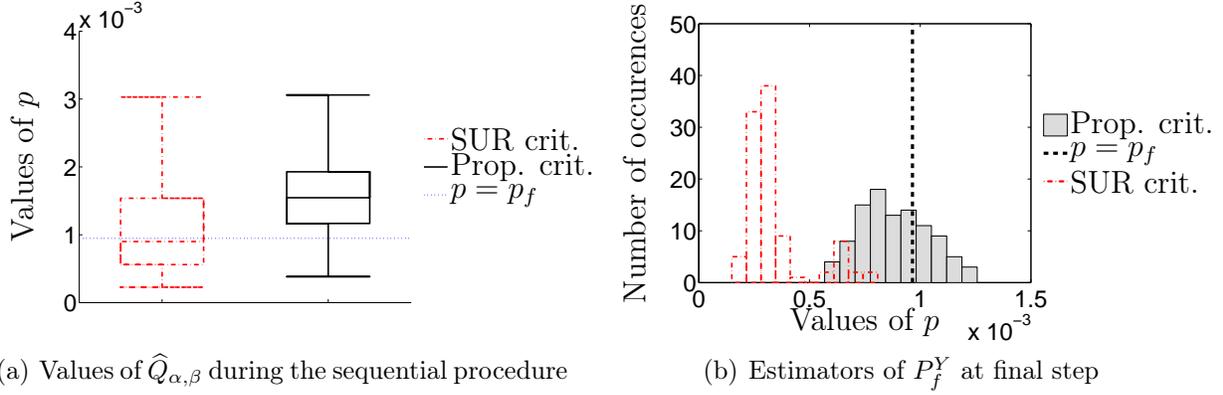
16

(a) Values of $\widehat{Q}_{\alpha,\beta}$ during the sequential procedure

(b) Estimators of P_f^Y at final step

Figure 6: Distribution of the obtained values of $\widetilde{p}_j$ and $\widehat{Q}_{\alpha,\beta}$ obtained during the sequential procedure presented in Algorithms 1 and 2. Subfigure (a) compares for the SUR and the proposed criteria the distributions of all the values of $\widehat{Q}_{\alpha,\beta}$ obtained during the 10 repetitions of the sequential procedure. For a particular initial design of experiment, subfigure (b) compares the influence of the choice of the selection criterion on the distribution of 100 values of the estimators $\widetilde{p}_j$ of P_f^Y after having added n_{add} points.

the SUR strategy has difficulties to make the mean estimator converge, when the results are more satisfying for the proposed criterion (the results associated with the pointwise selection criterion being not relevant enough, we decided not to show them for this example). More prejudicial, if we look at Figure 6(a), we see that during the 10 repetitions of the n_{add} steps of the proposed sequential procedure, around 50% of the obtained values of $\widehat{Q}_{\alpha,\beta}$ are inferior to p_f. On the contrary, when considering the proposed selection criterion, the values of $\widehat{Q}_{\alpha,\beta}$ are, as required, higher than p_f with high probability during the whole sequential procedure. In addition, Figure 6(b) shows 100 independent realizations of the estimator of p_f that can be obtained using Algorithm 1 after having added n_{add} points to a particular initial design of experiments using Algorithm 2. In line with previous conclusions, we see in this figure a nice dispersion of these realizations around the true value p_f when the proposed criterion is considered, which is not the case when considering the SUR criterion.

4.2. Industrial example

The fourth application deals with the reliability analysis of a spherical containment vessel, in which detonation experiments are carried out. More precisely, a spherical explosive charge is placed at the geometric center of the vessel. Once it is ignited, a non regular transient pressure is observed on the inner surface of the containment vessel, which puts the system into vibration. We then monitor the displacement during the experiment of each point of the system, and record the maximum value of this displacement as the quantity of interest. For confidential reasons, only few numerical values will be provided for this last example.

Two codes are coupled to simulate this experiment. The first code is a simplified version of a detonation code. It takes as inputs the radius of the explosive charge, and three constants associated with the behavior law of the explosive. The difference between the true value of

17

the radius and its nominal value is noted x_1, while the uncertainties on the parameters of the behavior law are noted x_2, x_3, x_4. Given values for x_1 to x_4, the detonation code then provides the time evolution of the inner pressure, which is denoted by $y_{\text{press}}(x_1, x_2, x_3, x_4)$. The second code is a damped oscillator characterized by the following second order linear differential equation:

$$\ddot{u} + 2(\overline{x}_5 + x_5)(\overline{x}_6 + x_6)\dot{u} + (\overline{x}_6 + x_6)^2 u = y_{\text{press}}(x_1, x_2, x_3, x_4). \qquad (29)$$

The quantities $\overline{x}_5$ and $\overline{x}_6$ correspond to the nominal value of the attenuation coefficient and to the natural pulsation of the structure, when x_5 and x_6 are potential fluctuations around these nominal values. As an illustration, time evolutions of y_{press} and u can be found in Figure 7 for several values of the input vector $\boldsymbol{x} := (x_1, \dots, x_6)$. The quantity of interest is therefore given by:

$$y(\boldsymbol{x}) := s - \max_{t \geq 0} u(t; \boldsymbol{x}), \qquad (30)$$

with s a given threshold. Vector $\boldsymbol{x}$ is seen as a random vector to model the fact that its values are known in an uncertain way, and we then try to quantify the risk that u is greater than s, i.e. to calculate the probability that y is smaller than 0 to stick to the notations of the previous sections. To this end, we first computed the value of y in $\ell = 50$ points uniformly chosen in the input space. None of these values of y was below 0. We then launched Algorithm 1 with $m = 100$ and $q = 100$, which allowed us to obtain $m = 100$ approximations of p_f. The smallest and largest values obtained for these 100 realizations were 0.00144 and 0.0053 respectively. In order to reduce this uncertainty on p_f, Algorithm 2 was run, with $\alpha = \beta = 0.1$, $n_{\max} = 3000$, $q = 100$ and $n_{\text{add}} = 50$. Thus, looking at Figure 8, which represents the dispersions of $m = 100$ approximations of p_f with respect to several values of n_{add}, we notice that adding new code evaluations changes the mean of the estimates but also the range of admissible values for p_f. The proximity between the boxplots at $n_{\text{add}} = 25$ and $n_{\text{add}} = 50$ then seems to indicate that the algorithm has converged relatively well in terms of reducing the uncertainty associated with the surrogate model. At this stage, the main contributor to the uncertainty seems to be linked to the dispersion of the estimator itself. By increasing q from 100 to 1000 (third and fourth boxplots from the left respectively), we can then strongly reduce this uncertainty, to obtain a quite reasonable dispersion of values for p_f. In particular, after post-processing the realizations of the Poisson process associated with the estimation of p_f using Eq. (11), we can say that $\mathbb{P}(0.00505 > q_{\alpha=0.1}) \geq 90\%$, where we remind that $q_{\alpha=0.1}$ is the 90%-quantile of P_f^Y. It is important to note that this increase in q does not require any additional calls to the code. Finally, to go further in terms of dispersion reduction, it would be necessary to increase q again, but probably also to add new evaluations of y.

5. Conclusion and prospects

This paper focuses on the guarantee by simulation of the correct functioning of complex systems. These guarantees are most often based on the fact of being able to ensure that
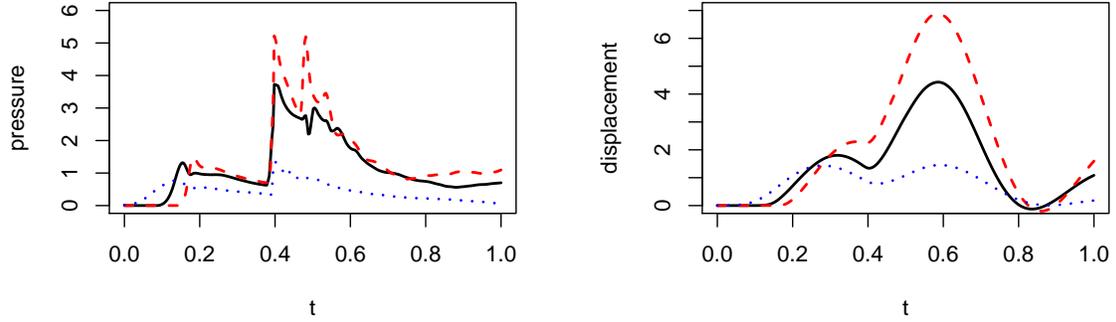
(a) Time evolution of the pressure

(b) Time evolution of the displacement

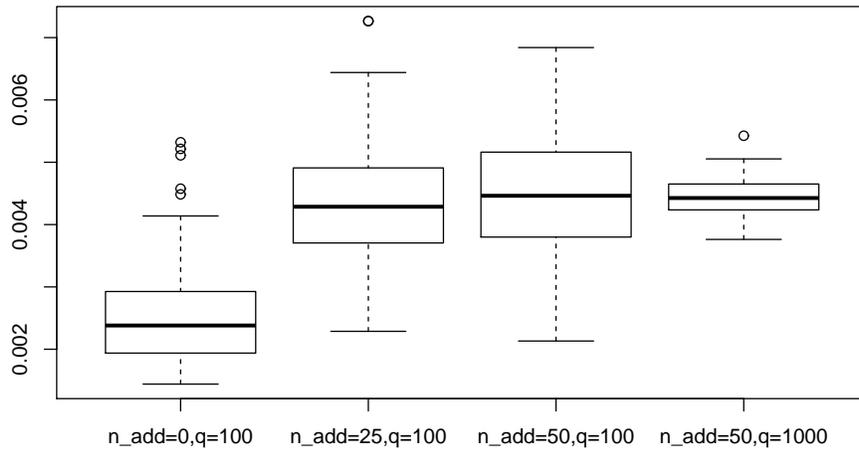Figure 7: Evolution of y_{press} and u for three different values of $\boldsymbol{x}$.

Figure 8: Comparison of the dispersions obtained on the estimates of p_f as a function of the number of points added n_{add} and the value of q.

the probability of occurrence of undesired events is lower than a risk that is considered acceptable. With this objective, this paper proposes a method to bound this probability with a specified confidence, while requiring only a reduced number of calls to the code used to simulate the behavior of the studied system. This method is based on the substitution of the potentially numerically costly code by a GPR-based surrogate model, the use of order statistics, and the exploitation of a dedicated Poisson process. One point that should be emphasized is that the method is intuitive, which makes it easy to implement.

In addition, a sequential strategy is proposed, allowing us to refine this boundary by adding, in an optimized way, new learning points for the surrogate model. In particular, it should be noted that this strategy proposes a natural compromise between exploration and exploitation, allowing to avoid a too strong sensitivity of the results to the fact that the predictive capacities of the initial surrogate model may not be very good. The interest of such an approach compared to already existing methods is finally illustrated on a series of numerical examples.

When the total number of code evaluations is fixed, it would nevertheless seem interesting, in terms of perspectives for this work, to progress on the choice of the number of code evaluations dedicated to initial learning, noted ℓ in this paper, and the number of code evaluations dedicated to sequential improvement, noted n_{add}.

Appendix

Proof of Proposition 1

Let $(Z_i)_{i \geq 0}$ be a decreasing random walk so that for all $i \geq 1$ and $z \in \mathbb{R}$:

$$Z_0 = +\infty, \ \mathbb{P}(Z_{i+1} \leq z \mid Z_i) = \mathbb{P}(Z \leq z \mid Z \leq Z_i), \tag{31}$$

with Z a real-valued random variable with continuous CDF F_Z. We can then associate to this random walk the results of Proposition 4 and Corollary 2.

Proposition 4. *For all $t > 0$,*

$$P(t) := \sup \{i; \ -\log \left(\mathbb{P}(Z \leq Z_i | Z_i) \right) \leq t\} \tag{32}$$

is a Poisson random process with parameter t.

□ **Proof:** By construction, for all $i \geq 0$, $\mathbb{P}\left(Z \leq Z_{i+1} \mid Z \leq Z_i\right)$ is a random variable that is uniformly distributed over $[0, 1]$, such that $S_{i+1} := -\log(\mathbb{P}(Z \leq Z_{i+1} | Z \leq Z_i))$ follows an exponential law with parameter 1. Therefore, for all $t > 0$, $P(t) := \sup \{i; \ S_1 + \ldots + S_i \leq t\}$ is a Poisson process with parameter t. Noticing that:

$$S_1 + \ldots + S_i = -\sum_{k=1}^{i} \log(\mathbb{P}(Z \leq Z_k | Z \leq Z_{k-1}))$$

$$= -\log\left(\prod_{k=1}^{i} \mathbb{P}(Z \leq Z_k | Z \leq Z_{k-1})\right) \tag{33}$$

$$= -\log\left(\mathbb{P}(Z \leq Z_i \mid Z_i)\right),$$

we eventually find the searched result. $\square$

Corollary 2. *For $t > 0$ and $q \geq 1$, if $P_1, \ldots, P_q$ denote q independent copies of $P(t)$, the random variable $\Pi_q := \left(1 - \frac{1}{q}\right)^{\sum_{k=1}^{q} P_k}$ verifies:*

$$\mathbb{E}\left[\Pi_q\right] = \mathrm{e}^{-t}, \quad \mathrm{Var}(\Pi_q) = \mathrm{e}^{-2t}(\mathrm{e}^{t/q} - 1). \tag{34}$$

$\square$ **Proof:**
We have:

$$\mathbb{E}\left[\Pi_q\right] = \left(\mathbb{E}\left[\left(1 - \frac{1}{q}\right)^{P(t)}\right]\right)^q$$

$$= \left(\sum_{k=0}^{+\infty} \left(1 - \frac{1}{q}\right)^k \frac{\mathrm{e}^{-t} t^k}{k!}\right)^q$$

$$= \mathrm{e}^{-t}\left(\sum_{k=0}^{+\infty} \left(\left(1 - \frac{1}{q}\right) t\right)^k \frac{\mathrm{e}^{-\left(1 - \frac{1}{q}\right)t}}{k!}\right)^q$$

$$= \mathrm{e}^{-t}.$$

$$\mathbb{E}\left[\Pi_q^2\right] = \left(\mathbb{E}\left[\left(1 - \frac{1}{q}\right)^{2P(t)}\right]\right)^q$$

$$= \left(\sum_{k=0}^{+\infty} \left(1 - \frac{1}{q}\right)^{2k} \frac{\mathrm{e}^{-t} t^k}{k!}\right)^q$$

$$= \mathrm{e}^{qt\left(-1 + \left(1 - \frac{1}{q}\right)^2\right)}\left(\sum_{k=0}^{+\infty} \left(\left(1 - \frac{1}{q}\right)^2 t\right)^k \frac{\mathrm{e}^{-\left(1 - \frac{1}{q}\right)^2 t}}{k!}\right)^q$$

$$= \mathrm{e}^{-2t + t/q},$$

from which we deduce the searched result.

□

Applying the results of Property 4 and Corollary 2 with $Z = Y(\boldsymbol{X}; \omega)$ and $t = -\log\left(\mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}; \omega) < 0)\right)$, we obtain:

$$P(-\log\left(\mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}; \omega) < 0)\right)) = \sup\{i; \; Z_i \geq 0\}. \tag{35}$$

Hence, if $P_1, \ldots, P_q$ denote q independent copies of Poisson process $P(-\log\left(\mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}; \omega) < 0)\right))$, which are based on the projection of deterministic function $Y(\omega)$ in a total number of n points of $\mathbb{X}$ gathered in $\mathcal{X}_n$, $\widehat{P}_f^{Y, \mathcal{X}_n}(\omega) := \left(1 - \frac{1}{q}\right)^{\sum_{k=1}^{q} P_k}$ defines an estimator of $P_f^Y(\omega) = \mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X}; \omega) < 0)$, and it verifies the equalities of Eq. (14).

Proof of Proposition 2

Lemma 1. *Let Z be a real-valued random variable with continuous CDF F_Z. If $(Z_i)_{i \geq 0}$ is the decreasing random walk defined by Eq.(17), for all $z \in \mathbb{R}, i \geq 1$, we have:*

$$\mathbb{P}(Z_i \leq z) = F_Z(z) \sum_{k=0}^{i-1} \frac{|\log(F_Z(z))|^k}{k!}. \tag{36}$$

Proof of Lemma 1.
The result is straightforward for $i = 1$.
Let us assume that the expression is true for all $1 \leq k \leq i$. For all $z, z' \in \mathbb{R}$, we have:

$$\mathbb{P}(Z_{i+1} \leq z \mid Z_i = z') = \begin{cases} 1 & \text{if } z' \leq z, \\ \frac{F_Z(z)}{F_Z(z')} & \text{otherwise.} \end{cases} \tag{37}$$

Hence, if f_{Z_i} is the PDF of Z_i, we have:

$$\begin{aligned} f_{Z_i}(z) =& f_Z(Z) \left(\sum_{k=0}^{i-1} \frac{|\log(F_Z(z))|^k}{k!} - \sum_{k=1}^{i-1} \frac{|\log(F_Z(z))|^{k-1}}{(k-1)!} \right) \\ =& f_Z(z) \frac{|\log(F_Z(z))|^{i-1}}{(i-1)!}. \end{aligned}$$

Then, using Eq. (37), it comes:

$$\mathbb{P}(Z_{i+1} \le z) = \mathbb{E}\left[\mathbb{P}\left(Z_{i+1} \le z \mid Z_i\right)\right]$$

$$= \int_{-\infty}^{z} \mathbb{P}(Z_{i+1} \le z | Z_i = y) f_{Z_i}(y) dy + \int_{z}^{+\infty} \mathbb{P}(Z_{i+1} \le z | Z_i = y) f_{Z_i}(y) dy$$

$$= \int_{-\infty}^{z} f_{Z_i}(y) dy + \int_{z}^{+\infty} \frac{F_Z(z)}{F_Z(y)} f_{Z_i}(y) dy$$

$$= \mathbb{P}(Z_i \le z) + F_Z(z) \int_{z}^{+\infty} \frac{f_Z(y)}{F_Z(y)} \frac{|\log(F_Z(y))|^{i-1}}{(i-1)!} dy$$

$$= \mathbb{P}(Z_i \le z) + \frac{F_Z(z)}{i!} \left[-|\log(F_Z(y))|^n\right]_z^{+\infty}$$

$$= F_Z(z) \sum_{k=0}^{i-1} \frac{|\log(F_Z(z))|^k}{k!} + F_Z(z) \frac{|\log(F_Z(z))|^i}{i!}$$

$$= F_Z(z) \sum_{k=0}^{i} \frac{|\log(F_Z(z))|^k}{k!}.$$

Proof of Property 2.
Let $\widehat{Z}$ be an element of $(Z_i)_{i \ge 0}$ chosen at random. It comes:

$$\mathbb{P}(\widetilde{Z} \le z) = 1 - \mathbb{P}(\widehat{Z} \ge z \mid \widehat{Z} \ge 0)$$

$$= 1 - \frac{\mathbb{P}(\widehat{Z} \ge z)}{\mathbb{P}(\widehat{Z} \ge 0)} \tag{38}$$

$$= 1 - \frac{\sum_{i=1}^{+\infty} \mathbb{P}(Z_i \ge z)}{\sum_{i=1}^{+\infty} \mathbb{P}(Z_i \ge 0)}.$$

Then, for all $z \ge 0$,

$$\sum_{i=1}^{+\infty} \mathbb{P}(Z_i \ge z) = \sum_{i=1}^{+\infty} \sum_{k=i}^{+\infty} F_Z(z) \frac{|\log(F_Z(z))|^k}{k!}$$

$$= \sum_{k=1}^{+\infty} k \times F_Z(z) \frac{|\log(F_Z(z))|^k}{k!}$$

$$= |\log(F_Z(z))| \sum_{k=1}^{+\infty} F_Z(z) \frac{|\log(F_Z(z))|^{k-1}}{(k-1)!} \tag{39}$$

$$= |\log(F_Z(z))| \sum_{k=0}^{+\infty} F_Z(z) \frac{|\log(F_Z(z))|^k}{k!}$$

$$= |\log(F_Z(z))|,$$

which completes the proof.

Proof of Corollary 1

The result of Corollary 1 is a direct consequence of Property 2 when replacing Z by $Y(\boldsymbol{X};\omega)$ and writing $P_f^Y(\omega) = \mathbb{P}_{\boldsymbol{X}}(Y(\boldsymbol{X};\omega) \le 0)$.

Proof of Property 3

For $\boldsymbol{x}, \boldsymbol{x}', \widehat{\boldsymbol{x}}$ in $\mathbb{X}$, we remind that:

$$\begin{pmatrix} Y(\boldsymbol{x}) \\ Y(\boldsymbol{x}') \\ Y(\widehat{\boldsymbol{x}}) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu(\boldsymbol{x}) \\ \mu(\boldsymbol{x}') \\ \mu(\widehat{\boldsymbol{x}}) \end{pmatrix}, \begin{bmatrix} \Sigma(\boldsymbol{x}, \boldsymbol{x}) & \Sigma(\boldsymbol{x}, \boldsymbol{x}') & \Sigma(\boldsymbol{x}, \widehat{\boldsymbol{x}}) \\ \Sigma(\boldsymbol{x}', \boldsymbol{x}) & \Sigma(\boldsymbol{x}', \boldsymbol{x}') & \Sigma(\boldsymbol{x}', \widehat{\boldsymbol{x}}) \\ \Sigma(\widehat{\boldsymbol{x}}, \boldsymbol{x}) & \Sigma(\widehat{\boldsymbol{x}}, \boldsymbol{x}') & \Sigma(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}}) \end{bmatrix} \right). \tag{40}$$

It comes:

$$Y(\boldsymbol{x}') - Y(\boldsymbol{x}) \mid Y(\widehat{\boldsymbol{x}}) = \mu(\widehat{\boldsymbol{x}}) \sim \mathcal{N}\left(\mu(\boldsymbol{x}') - \mu(\boldsymbol{x}), s^2(\boldsymbol{x}, \boldsymbol{x}', \widehat{\boldsymbol{x}}) \right), \tag{41}$$

$$s^2(\boldsymbol{x}, \boldsymbol{x}', \widehat{\boldsymbol{x}}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) + \Sigma(\boldsymbol{x}', \boldsymbol{x}') - 2\Sigma(\boldsymbol{x}, \boldsymbol{x}') - \frac{(\Sigma(\boldsymbol{x}, \widehat{\boldsymbol{x}}) - \Sigma(\boldsymbol{x}', \widehat{\boldsymbol{x}}))^2}{\Sigma(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{x}})}, \tag{42}$$

so that:

$$\begin{aligned}
&\mathbb{P}_{Y,\boldsymbol{X},\boldsymbol{X}'}\left(Y(\boldsymbol{X}) > Y(\boldsymbol{X}'), \mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \mid Y(\widehat{\boldsymbol{x}}) = \mu(\widehat{\boldsymbol{x}}) \right), \\
&= \mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[1_{\mu(\boldsymbol{X})<\mu(\boldsymbol{X}')} \mathbb{P}_Y\left(Y(\boldsymbol{X}') - Y(\boldsymbol{X}) < 0 \mid \boldsymbol{X}, \boldsymbol{X}', Y(\widehat{\boldsymbol{x}}) = \mu(\widehat{\boldsymbol{x}}) \right) \right], \\
&= \mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[1_{\mu(\boldsymbol{X})<\mu(\boldsymbol{X}')} \Phi\left(\frac{\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}')}{s(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}})} \right) \right], \\
&= \mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[\Phi\left(\frac{\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}')}{s(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}})} \right) \mid \mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \right] \mathbb{P}_{\boldsymbol{X},\boldsymbol{X}'}\left(\mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \right), \quad (43) \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[\Phi\left(\frac{\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}')}{s(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}})} \right) \mid \mu(\boldsymbol{X}) < \mu(\boldsymbol{X}') \right], \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{X},\boldsymbol{X}'}\left[\Phi\left(-\frac{|\mu(\boldsymbol{X}) - \mu(\boldsymbol{X}')|}{s(\boldsymbol{X}, \boldsymbol{X}', \widehat{\boldsymbol{x}})} \right) \right]
\end{aligned}$$

Expression of the analyzed numerical functions

The oscillating function corresponds to the function:

$$y : \begin{cases} [0,1] & \to & \mathbb{R} \\ x & \mapsto & (x+1)\sin(6\pi x^2) + 1.92. \end{cases} \tag{44}$$

The Branin function corresponds to the function:

$$y : \begin{cases} [0,1]^2 & \to & \mathbb{R} \\ \boldsymbol{x} & \mapsto & \left(15x_2 - \frac{5}{4\pi^2}(15x_1 - 5)^2 + \frac{3x_1 - 1}{\pi} - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi} \right)\cos(15x_1 - 5) + 10. \end{cases} \tag{45}$$

The Hartmann function corresponds to the function:

$$
y : \begin{cases} [0,1]^6 & \to & \mathbb{R} \\ \boldsymbol{x} & \mapsto & -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right), \end{cases} \tag{46}
$$

$$
\boldsymbol{\alpha} = (1.0, 1.2, 3.0, 3.2),
$$

$$
A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix},
$$

$$
P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}.
$$

[1] Andres Alban, Hardik A. Darji, Atsuki Imamura, and Marvin K. Nakayama. Efficient monte carlo methods for estimating failure probabilities. *Reliability Engineering & System Safety*, 165:376 – 394, 2017.

[2] 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.

[3] Mathieu Balesdent, Jérôme Morio, and Julien Marzat. Recommendations for the tuning of rare event probability estimators. *Reliability Engineering & System Safety*, 133:68 – 78, 2015.

[4] 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.

[5] Julien Bect, Ling Li, and Emmanuel Vazquez. Bayesian subset simulation. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):762–786, 2017.

[6] Francesco Cadini, Gian Luca Agliardi, and Enrico Zio. Estimation of rare event probabilities in power transmission networks subject to cascading failures. *Reliability Engineering & System Safety*, 158:9 – 20, 2017. Special Sections : Reliability and Safety Certification of Software-Intensive Systems.

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

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

[9] 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.

[10] V. Dubourg, B. Sudret, and F. Deheeger. Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, 33:47 – 57, 2013.

[11] 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.

[12] Abdul-Kader El Haj and Abdul-Hamid Soubra. Improved active learning probabilistic approach for the computation of failure probability. *Structural Safety*, 88:102011, 2021.

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

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

[15] Qing Guo, Yongshou Liu, Bingqian Chen, and Yuzhen Zhao. An active learning kriging model combined with directional importance sampling method for efficient reliability analysis. *Probabilistic Engineering Mechanics*, 60:103054, 2020.

[16] F Hayashi. *Econometrics.* Princeton University Press, 2000.

[17] J. Hortado. *Structural reliability - Statistical learning perspectives*, volume 17. Lectures notes in applied and computational mechanics, Springer, 2002.

[18] Xiaoxu Huang, Jianqiao Chen, and Hongping Zhu. Assessing small failure probabilities by ak-ss: An active learning method combining kriging and subset simulation. *Structural Safety*, 59:86 – 95, 2016.

[19] M C Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal of the royal statistical society*, 63:425–464, 2001.

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

[21] Jing Li and Dongbin Xiu. Evaluation of failure probability via surrogate models. *Journal of Computational Physics*, 229(23):8966 – 8980, 2010.

[22] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68:411–436, 2006.

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

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

[25] Nassim Razaaly and Pietro Marco Congedo. Extension of AK-MCS for the efficient computation of very small failure probabilities. *Reliability Engineering & System Safety*, 203:107084, 2020.

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

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

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

[29] B. Scholkopf and A.J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Boston, 2002.

[30] A Takeshi. *Advanced econometrics*. Harvard University Press, 1985.

[31] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

[32] 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.

[33] Zeyu Wang and Abdollah Shafieezadeh. On confidence intervals for failure probability estimates in kriging-based reliability analysis. *Reliability Engineering & System Safety*, 196:106758, 2020.

[34] Sinan Xiao, Sergey Oladyshkin, and Wolfgang Nowak. Reliability analysis with stratified importance sampling based on adaptive kriging. *Reliability Engineering & System Safety*, 197:106852, 2020.

[35] Chunlong Xu, Weidong Chen, Jingxin Ma, Yaqin Shi, and Shengzhuo Lu. AK-MSS: An adaptation of the AK-MCS method for small failure probabilities. *Structural Safety*, 86:101971, 2020.

[36] Jinhao Zhang, Mi Xiao, and Liang Gao. An active learning reliability method combining kriging constructed with exploration and exploitation of failure region and subset simulation. *Reliability Engineering & System Safety*, 188:90 – 102, 2019.

[37] Xufang Zhang, Lei Wang, and John Dalsgaard Sørensen. Akois: An adaptive kriging oriented importance sampling method for structural system reliability analysis. *Structural Safety*, 82:101876, 2020.

[38] M. Munoz Zuniga, A. Murangira, and T. Perdrizet. Structural reliability assessment through surrogate based importance sampling with dimension reduction. *Reliability Engineering & System Safety*, 207:107289, 2021.