ETH zürich

Surrogate-based reliability analysis for noisy models

A. Pires, M. Moustapha, S. Marelli, B. Sudret

Chair of Risk, Safety and Uncertainty Quantification
Stefano-Franscini-Platz 5
CH-8093 Zürich

Risk, Safety & Uncertainty Quantification

Data Sheet

Journal:	Proc. 14th Int. Conf. Applications of Statistics and Probability in Civil Engineering (ICASP14), Dublin (Ireland)
Report Ref.:	RSUQ-2023-007
Arxiv Ref.:	
DOI:	-
Date submitted:	
Date accepted:	July 9, 2023

14th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP14
Dublin, Ireland, July 9-13, 2023

Surrogate-based reliability analysis for noisy models

Anderson Pires
Graduate Student, Chair of Risk, Safety and Uncertainty quantification, ETH Zurich, Zurich, Switzerland

Maliki Moustapha
Researcher, Chair of Risk, Safety and Uncertainty quantification, ETH Zurich, Zurich, Switzerland

Stefano Marelli
Researcher, Chair of Risk, Safety and Uncertainty quantification, ETH Zurich, Zurich, Switzerland

Bruno Sudret
Professor, Chair of Risk, Safety and Uncertainty quantification, ETH Zurich, Zurich, Switzerland

ABSTRACT: Reliability analysis provides a logical methodology for the estimation of the probability of failure of a system and often requires many runs of an expensive-to-evaluate limit state function. Surrogate models can be deployed to reduce the computational cost associated with such analysis, Kriging being arguably the best-known surrogate model for reliability analysis. Although replacing full-scale simulations with surrogate models is well-established for deterministic limit state functions, there is still the need to extend it to non-deterministic cases. To bridge this gap, we propose using regression-based surrogate models for reliability estimation of noisy limit state functions. The performance of this method is demonstrated on well-known reliability benchmark problems artificially corrupted with noise. Our results show that regression-based surrogate models can be used to effectively denoise these models and estimate the associated probability of failure.

1. INTRODUCTION

Structural reliability aims at quantifying the probability that the uncertainty in the input parameters of a system leads to performance failure. To compute such a probability of failure P_f, we consider a probabilistic framework where the input parameters are denoted by a random vector $\mathbf{X}$. Their respective uncertainties are entirely characterized by their joint probability distribution function (PDF) $f_{\mathbf{X}}$, defined on the domain $\mathscr{D}_{\mathbf{X}} \subset \mathbb{R}^M$. Moreover, the so-called *limit state function* $g(\mathbf{x})$, often based on an expensive-to-evaluate simulation model, takes as input the parameters described by $\mathbf{X}$ and outputs the state of the system, *i.e.* whether the physical system fails or not. By convention, failure occurs when $g(\mathbf{x}) \leqslant 0$. Consequently, the so-called failure domain $\mathscr{D}_f$ can be defined as $\{\mathbf{x} : g(\mathbf{x}) \leqslant 0\}$. The set of points $\mathbf{x} \in \mathscr{D}_{\mathbf{X}}$ such that $g(\mathbf{x}) = 0$ forms to the so-called *limit state surface*. Given this framework, it is possible to compute the probability of failure as follows (Melchers, 1989):

$$P_f = \mathbb{P}(g(\mathbf{X}) \leqslant 0) = \int_{\mathscr{D}_f} f_{\mathbf{X}}(\mathbf{x})\,\mathrm{d}\mathbf{x}. \qquad (1)$$

Because the integration domain in Eq. (1) is multi-dimensional and implicitly dependent on $g(\mathbf{x})$, it is rarely possible to evaluate Eq. (1) analytically.

Monte Carlo simulation (MCS) is a robust and straightforward method that can be used to estimate P_f. However, it requires circa 10^{k+2} simulations to estimate $P_f \approx 10^{-k}$. As P_f is small by design, the cost of performing MCS is often prohibitive.

Surrogate models (also known as metamodels) are cheap-to-evaluate mathematical approximations that circumvent the cost of running full-scale simulations. They are trained on a limited set of model responses, called *experimental design* (ED), and can be used as a proxy for expensive simulations. Because obtaining the ED is costly, as it depends on running expensive simulations, it is crucial to define a training set that balances its size and the accuracy of the prediction. To this aim, metamodels used in the context of reliability analysis often make use of *active learning* (Teixeira et al., 2021; Moustapha et al., 2022). This class of methods deploys so-called *learning functions* to identify the best points to introduce in the ED so as to accurately approximate the limit state surface.

Surrogate models can be split into two broad classes: *interpolation-* and *regression-* based. Interpolation methods assume that the training points are not affected by random noise. Consequently, they precisely interpolate through the points of the ED. Regression methods, on the other hand, assume that the training set is contaminated with a noise term. For this reason, they do not strictly match the training points. Instead, they aim at minimizing a global cost function, usually the squared error.

Because computational models are usually assumed to be deterministic, interpolation methods have been successfully employed in reliability analysis. However, not all models are deterministic. For instance, Forrester et al. (2006) commented on the so-called "numerical noise" for computational fluid dynamics (CFD) simulations. Paz et al. (2020); Ahmadisoleymani and Missoum (2021) also report a similar sort of noise, called "simulation noise", in crashworthiness simulations. In the context of reliability analysis, numerical noise has also been observed in geotechnical models (van den Eijnden et al., 2021). In these cases, features indirectly related to the simulation, such as numerical precision or the meshing procedure, cause the simulation to behave non-deterministically. For this particular type of noise, a perturbation around the mean trend of the output is observed. Employing interpolation approaches in this context can lead to overfitting, ultimately converging to an incorrect failure probability estimate (van den Eijnden et al., 2021).

We propose to extend reliability analysis to non-deterministic simulations, by using regression-based surrogate models to obtain the underlying noise-free probability of failure. In this contribution, we apply this approach to two well-known benchmark problems, relying either on polynomial chaos expansions (PCE) or Gaussian process regression (GPR) and different active learning strategies.

2. Active learning methods for estimating the probability of failure

Moustapha et al. (2022) show that active learning methods for reliability analysis comprise four main components, that work as depicted in Algorithm 1. The *surrogate model* is the first and most important component of active learning reliability, affecting the properties and convergence of the estimation.

Algorithm 1 Pseudo-code of active learning methods

Require: $\mathscr{E}^{(0)}$ (Initial ED)
  $i \leftarrow 0$
  repeat
    Build *surrogate model* $\hat{g}^{(i)}\left(\mathbf{x};\mathscr{E}^{(i)}\right)$
    Compute $P_f^{(i)}$ using $\hat{g}^{(i)}\left(\mathbf{x};\mathscr{E}^{(i)}\right)$ using a *reliability estimation method*
    Use a *learning function* to select $\mathscr{X}^{next}$
    Enrich ED $\mathscr{E}^{(i+1)} \leftarrow \mathscr{E}^{(i)} \cup \{\mathscr{X}^{next}, \tilde{g}(\mathscr{X}^{next})\}$
    $i \leftarrow i+1$
  until *stopping criterion* is fulfilled

The second component of the framework is the *reliability estimation method*, which defines how the probability of failure is estimated at each iteration. Possible methods for this purpose are MCS (Rubinstein and Kroese, 2016) or other advanced methods, such as importance sampling (Melchers, 1989) or subset simulation (Au and Beck, 2001).

The third component of the methodology is the *learning function* that sequentially identifies the best

points to introduce in the experimental design. This component synergises strongly with the surrogate model and the reliability estimation method, often requiring measures of uncertainty of the surrogate model predictions.

The last component of the method is the *stopping criterion*. It affects the termination of the algorithm, and is therefore related to its efficiency and robustness. Loose stopping criteria can lead to premature termination, whereas too strict ones can lead to an unnecessary costly ED.

2.1. Surrogate Models

2.1.1. PCE basics

Under the assumption that $g(\mathbf{X})$ has finite variance, it can be approximated as follows:

$$g(\mathbf{X}) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_\alpha \boldsymbol{\Psi}_{\boldsymbol{\alpha}}(\mathbf{X}) + \varepsilon_P, \tag{2}$$

where c_α are real coefficients and $\boldsymbol{\Psi}_\alpha(\mathbf{X})$ are multivariate polynomials orthonormal w.r.t. $f_{\mathbf{X}}$. $\mathscr{A}$ corresponds to the truncation set of the $\boldsymbol{\alpha}$ multi-index, which identifies the degree of the multivariate polynomial along each input variable. ε_P is the error introduced as a consequence of the truncation.

Given a noise-corrupted ED, defined as

$$\begin{aligned}\mathscr{E} = \Bigg\{ &\left(\mathscr{X}^{(i)}, \widetilde{\mathscr{Y}^{(i)}}\right) : \widetilde{\mathscr{Y}^{(i)}} = \tilde{g}\left(\mathscr{X}^{(i)}\right) \in \mathbb{R}, \\ &\mathscr{X}^{(i)} \in \mathscr{D}_{\mathbf{X}} \subset \mathbb{R}^M, i = 1,\dots,n \Bigg\},\end{aligned} \tag{3}$$

it is possible to compute the PCE coefficients via least-squares minimization:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{a}\in\mathbb{R}^P} \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{a}^T\boldsymbol{\Psi}\left(\mathbf{x}^{(i)}\right) - \tilde{g}\left(\mathbf{x}^{(i)}\right)\right)^2, \tag{4}$$

where $\boldsymbol{\Psi}\left(\mathbf{x}^{(i)}\right)$ is a vector of length $P = \operatorname{card}\mathscr{A}$ containing the values of the multivariate polynomials at point $\mathbf{x}^{(i)}$.

Advanced techniques for the solution of Eq. (4) that are robust to noise come from the compressive sensing literature. For an in-depth review of both truncation and regression techniques for PCE, the reader is referred to Lüthen et al. (2021, 2022).

2.1.2. Gaussian process regression basics

When employing GPR as a metamodel, the underlying assumption is that the output of the model, $\mathscr{M}(\mathbf{x})$, can be well approximated by a Gaussian process indexed by $\mathbf{x} \in \mathscr{D}_{\mathbf{X}}$. Under this premise, the surrogate is given as follows:

$$g^{GP}(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{f}(\mathbf{x}) + \sigma^2 Z(\mathbf{x},\omega). \tag{5}$$

The term $\boldsymbol{\beta}^T\mathbf{f}(\mathbf{x})$ corresponds to the mean (or *trend*) of the Gaussian process. The vector $\mathbf{f}(\mathbf{x})$ contains P pre-defined functions, and the vector $\boldsymbol{\beta}$ consists of the P coefficients associated with the functions. $Z(\mathbf{x},\omega)$ is a zero-mean, unit-variance and stationary Gaussian process, and the parameter σ^2 represents the variance of $g^{GP}(\mathbf{x})$. The parameter ω represents the underlying process variability. Also, because $Z(\mathbf{x},\omega)$ is a Gaussian process, it can be completely defined by its autocorrelation function, $R(\mathbf{x},\mathbf{x}';\boldsymbol{\theta})$, that depends on $(\mathbf{x}-\mathbf{x}')$ and hyperparameters $\boldsymbol{\theta}$.

Due to the Gaussian nature of the method, the predictor $g^{GP}(\mathbf{x})$ results in a normally distributed random variable. Its mean value $\mu_{\hat{g}}(\mathbf{x})$ and variance $\sigma_{\hat{g}}(\mathbf{x})$ are computed by conditioning $g^{GP}(\mathbf{x})$ to the observations in the ED, leading to the following expressions (Rasmussen and Williams, 2006):

$$\mu_{\hat{g}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T\hat{\boldsymbol{\beta}} + \tilde{\mathbf{r}}(\mathbf{x})^T\widetilde{\mathbf{R}}^{-1}(\widetilde{\mathscr{Y}} - \mathbf{F}\hat{\boldsymbol{\beta}}), \tag{6}$$

$$\sigma^2_{\hat{g}}(\mathbf{x}) = \sigma^2_{\text{total}}\left(1 - \tilde{\mathbf{r}}^T(\mathbf{x})\widetilde{\mathbf{R}}^{-1}\tilde{\mathbf{r}}(\mathbf{x}) + \mathbf{u}^T(\mathbf{x})\left(\mathbf{F}^T\widetilde{\mathbf{R}}^{-1}\mathbf{F}\right)^{-1}\mathbf{u}(\mathbf{x})\right), \tag{7}$$

where $\mathbf{F}$ is the design matrix, whose components are $F_{ij} = f_j\left(x^i\right)$ for $i = \{1,\dots,n\}$ and $j = \{1,\dots,P\}$. $\hat{\boldsymbol{\beta}} = \left(\mathbf{F}^T\widetilde{\mathbf{R}}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^T\widetilde{\mathbf{R}}^{-1}\widetilde{\mathscr{Y}}$ is the generalized least-squares estimate of the coefficients β, and $\mathbf{u}(\mathbf{x}) = \mathbf{F}^T\widetilde{\mathbf{R}}^{-1}\tilde{\mathbf{r}}(\mathbf{x}) - \mathbf{f}(\mathbf{x})$ is introduced for readability.

Additionally, to obtain a closed-form solution for Eq. (6) and Eq. (7), the noise component is cast as normally distributed and homoskedastic, *i.e.* $\varepsilon(\mathbf{x}) \sim \mathscr{N}\left(0, \sigma_O^2\mathbf{I}\right)$. Then,

$$\sigma^2_{\text{total}} = \sigma^2 + \sigma_O^2, \quad \tau = \frac{\sigma_O^2}{\sigma^2_{\text{total}}} \tag{8}$$

and

$$\tilde{\mathbf{r}} = (1-\tau)\,\mathbf{r}, \quad \widetilde{\mathbf{R}} = (1-\tau)\,\mathbf{R} + \tau\mathbf{I}, \tag{9}$$

where the parameter σ_O^2 corresponds to the variance of the noise. Finally, $\mathbf{R}$ is the correlation matrix, whose components are $R_{i,j} = R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta}\right), i,j = \{1,\ldots,n\}$, and $\mathbf{r}$ corresponds to the vector of cross-correlations with $r_i = R\left(\mathbf{x}, \mathbf{x}^{(i)}; \boldsymbol{\theta}\right), i = \{1,\ldots,n\}$. $\tilde{\mathbf{r}}(\mathbf{x})$ is the vector of cross-correlation between the points of the ED and the predicted point.

2.2. Reliability estimation method

As the surrogate models provide an inexpensive-to-evaluate approximation of the limit state function, it is possible to estimate P_f using MCS. Given a sample set of size N of the input random vector $\mathbf{X}$, $\mathscr{X}_{\mathrm{MC}} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right\}$, the MCS estimator reads:

$$P_{f,\mathrm{MC}} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{1}_{\mathscr{D}_f}\left(\mathbf{x}^{(k)}\right) \tag{10}$$

where $\mathbf{1}_{\mathscr{D}_f}(\mathbf{x})$ is an indicator function that takes 1 for $g(\mathbf{x}) \leqslant 0$, and 0 otherwise. Additionally, it is possible to define a discrete random variable $\mathscr{I} = \mathbf{1}_{\mathscr{D}_f}(\mathbf{X})$ with Bernoulli distribution and probability mass function $\mathbb{P}(\mathscr{I}=1) = P_f$ and $\mathbb{P}(\mathscr{I}=0) = 1 - P_f$. As corollary, the coefficient of variation of the estimator reads:

$$CV_{P_{f,\mathrm{MC}}} = \sqrt{\frac{1-P_{f,\mathrm{MC}}}{N \cdot P_{f,\mathrm{MC}}}} \tag{11}$$

The $CV_{P_{f,\mathrm{MC}}}$ indicates the accuracy of the Monte Carlo estimator and it can be used to assess whether N is large enough.

2.3. Learning functions

2.3.1. Failed bootstrap replicates

This learning function is compatible with PCE and its rationale is to introduce into the ED the point that is most likely misclassified by the surrogate model. To this aim, Marelli and Sudret (2018) propose using the bootstrap resampling technique. The ED is resampled with substitution B times leading to B different EDs, *i.e.* $\left\{\mathscr{E}^{(1)}, \ldots, \mathscr{E}^{(B)}\right\}$ such that $\mathscr{E}^{(1)} \subset \mathscr{E}, \ldots, \mathscr{E}^{(B)} \subset \mathscr{E}$. Each replication contains the same number of samples as the original ED and is used to build B different PCEs based on their respective resampled ED. Consequently, for each point $\mathbf{x}$, B PCE predictions are available. Assuming that B is large enough, it is possible to estimate the misclassification probability as follows:

$$U_{FBR}(\mathbf{x}) = \frac{|B_S(\mathbf{x}) - B_F(\mathbf{x})|}{B}, \tag{12}$$

where $B_S(\mathbf{x})$ (resp. $B_F(\mathbf{x})$) corresponds to the number of bootstrap responses that belongs to the safe (resp. failure) domain, and $B_S(\mathbf{x}) + B_F(\mathbf{x}) = B$.

Based on the utility function described by Eq. (12), it is possible to cast the learning function as follows:

$$\mathbf{x}^{\mathrm{next}} = \underset{\mathbf{x}\in\mathscr{X}_{\mathrm{MC}}}{\arg\min}\ U_{FBR}(\mathbf{x}) \tag{13}$$

Because Eq. (12) is cheap to evaluate, the optimization problem described by Eq. (13) is cast in a discrete manner. That is, U_{FBR} is computed for the so-called candidate points, a set of points from which x^{next} is selected. Herein the candidate points coincide with the Monte Carlo sample set $\mathscr{X}_{\mathrm{MC}}$. The benefits of this strategy are twofold: first, it ensures that the selected points will have significant density. Second, it dismisses the need for an optimization algorithm.

2.3.2. The learning function U

Echard et al. (2011) capitalized on the Kriging variance $\sigma_{\hat{g}}^2(\mathbf{x})$, a by-product of the method that can be seen a local error estimator, and proposed the learning function U. This function allows us to identify the point that is most likely misclassified by the surrogate model and relies on the so-called *deviation number*, defined as follows:

$$U(\mathbf{x}) = \frac{\left|\mu_{\hat{g}}(\mathbf{x})\right|}{\sigma_{\hat{g}}(\mathbf{x})}. \tag{14}$$

Thanks to the Gaussian nature of Kriging, it is possible to define the probability of misclassification $P_m(\mathbf{x})$ as a function of U, as follows:

$$P_m(\mathbf{x}) = \Phi(-U(\mathbf{x})), \tag{15}$$

where Φ is the standard Gaussian cumulative distribution function (CDF).

Finally, the learning function can be cast as follows:

$$x^{\mathrm{next}} = \underset{\mathbf{x}\in\mathscr{X}_{\mathrm{MC}}}{\arg\min}\ U(\mathbf{x}) = \underset{\mathbf{x}\in\mathscr{X}_{\mathrm{MC}}}{\arg\max}\ P_m(\mathbf{x}). \tag{16}$$

Similarly to the previous learning function, the optimisation is performed pointwise through a set of

candidate points that correspond to the Monte Carlo samples used for the reliability estimation.

2.4. Stopping criteria

Many stopping criteria for PCE and GPR have been proposed for noise-free cases and they can be split into two groups: one that depends on the stability of the reliability index/probability of failure, such as the one proposed by Marelli and Sudret (2018), whereas the other one relies on the parameters of the learning functions, *e.g.* the one suggested by Echard et al. (2011). However, due to noise, the convergence curve also becomes noisy. In this case, using stability as stopping criteria would lead to a non-stop algorithm. Similarly, the deviance number cannot be used as a stopping criterion, as because of noise, there will always be a baseline misclassification probability depending on the unknown noise of the model. In this case, pre-defining a fixed threshold for U is impossible. Although a parametric threshold depending on the estimated noise level is, in principle, possible, we opted to define a maximum budget as a stopping criterion, as the main target of this study is to obtain the underlying noise-free probability of failure. Thus, a budget of 3,000 and 600 points was considered for PCE and GPR, respectively.

3. ENRICHMENT STRATEGY

The behaviour of the learning functions can be classified as either explorative or exploitative. The former refers to the ability of the learning function to discover different failure regions throughout the domain. The latter is related to the capacity of sampling points that will lead to an accurate surrogate of the already known limit state surfaces. Consequently, learning functions with a too explorative behaviour tend to sample points in regions where the input PDF is irrelevant and fail to describe the limit state surface accurately. On the other hand, learning functions with a too exploitative behaviour tend to sample points very close to one another, accurately describing a particular region of the limit state surface but often failing to find all failure regions.

Although optimal learning functions should balance both behaviours, observing the ideal trade-off between the behaviours is not always possible. Indeed, these cases occur when the employed learning function is not optimal for the problem (Chevalier et al., 2014). If the optimal learning function is unknown, a possible workaround is to boost the disfavoured behaviour of the learning function artificially. For instance, employing the Monte Carlo sample set as candidate points is a strategy for pruning a too-explorative behaviour.

Conversely, a possible solution to increase the explorative behaviour of learning functions is to enrich the metamodel with multiple points at every iteration. The K-means clustering technique (Zaki and Meira, 2014) is one of the methods that enables doing this. In this method, the candidate set is partitioned into K different groups, found by minimising the squared Euclidean distance between points within the clusters. As solving such an optimisation problem can be computationally expensive, only a subset of the candidate points is considered. For the FBR learning function, only points with $U_{FBR}(\mathbf{x}) \leqslant 0.5$ are considered. For the learning function U only the subset $\mathscr{S} = \left\{\mathbf{x} \in \mathscr{S} : -1.96 \cdot \sigma_{\hat{g}}(\mathbf{x}) \leqslant \mu_{\hat{g}}(\mathbf{x}) \leqslant 1.96 \cdot \sigma_{\hat{g}}(\mathbf{x})\right\}$ is considered. Finally, the optimisation process described by Eq. (13) and Eq. (16) is performed for each cluster, enabling the enrichment of K different points.

4. EFFECTS OF NOISE ON THE ESTIMATION OF THE PROBABILITY OF FAILURE

4.1. Monte Carlo simulation

We employ the so-called R-S problem, one of the few cases when analytical computation of the P_f is possible, to showcase the effects of noise in MCS. The random vector of independent inputs for this problem is given by $\mathbf{X} = \{R, S\}$, where $R \sim \mathscr{N}\left(\mu_R, \sigma_R^2\right)$ represents the resistance, and $S \sim \mathscr{N}\left(\mu_S, \sigma_S^2\right)$ denotes the demand on an engineering system. The associated noise-free and noisy limit state functions can be cast as follows:

$$g(\mathbf{x}) = r - s \qquad \text{and} \qquad \tilde{g}(\mathbf{x}) = r - s + \varepsilon, \quad (17)$$

where ε is introduced to account for the simulation noise. In this contribution, as standard practice, we corrupt the noise-free model by casting $\varepsilon \sim \mathscr{N}\left(0, \sigma_\varepsilon^2\right)$. Then, the analytical probability of failure to the noise-free and noisy R-S problems read:

$$P_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}}\right) \text{ and } \tilde{P}_f = \Phi\left(-\frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2 + \sigma_\varepsilon^2}}\right). \quad (18)$$

Figure 1 depicts boxplots of $P_{f,\mathrm{MC}}$ for several noise levels, *i.e.* different values of σ_ε. The boxplots were obtained using 15 repetitions of the analysis to account for the statistical variability of $P_{f,\mathrm{MC}}$. Convergence is considered achieved when $CV_{P_{f,\mathrm{MC}}} \leqslant 1\%$.

The results presented in Figure 1 demonstrate that MCS fails to converge to the noise-free probability of failure. Instead, they converge to a different value which coincides with the noisy probability of failure $\tilde{P}_f$ shown in Eq. (18).

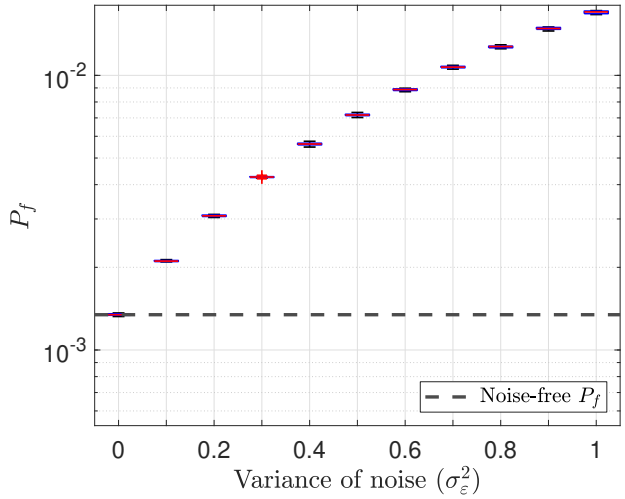


Figure 1: MC estimation of the probability of failure for the R-S problem

These results indicate an undesirable characteristic of MCS when dealing with noisy models. Since this method lacks denoising features, its outcome is affected by the artificial variability introduced by noise, leading to larger failure probabilities and potentially influencing decision-making. For these reasons, we consider MCS inappropriate for coping with noisy models. Instead, aiming at retrieving the associated noise-free probability of failure, we propose using regression-based surrogate models as a denoising tool. The rationale is that these metamodels converge to the noise-free limit state surface when the noise is unbiased. Finally, the limit state surface estimated by the surrogate model can be used for performing the reliability analysis through MCS.

5. Denoising with regression-based surrogate models

5.1. Benchmark functions

5.1.1. Four-branch function

The so-called four-branch function is a common benchmark for reliability analysis. It consists of an analytical model of a series system with four distinct failure regions. Multiple failure regions make the convergence to the correct probability of failure a non-trivial task for many classical algorithms. The noise-free limit state function reads:

$$g(\mathbf{x}) = \min \left\{ \begin{array}{c} 3+0.1\,(x_1+x_2)^2 - \frac{x_1+x_2}{\sqrt{2}} \\ 3+0.1\,(x_1+x_2)^2 + \frac{x_1+x_2}{\sqrt{2}} \\ (x_1-x_2)+\frac{6}{\sqrt{2}} \\ (x_2-x_1)+\frac{6}{\sqrt{2}} \end{array} \right\}. \quad (19)$$

The two input random variables are modelled as independent standard normal distributions, *i.e.* $X_1, X_2 \sim \mathcal{N}(0,1)$. The reference noise-free probability of failure for this function is $4.516 \cdot 10^{-3}$ $\left(CV_{P_{f,\mathrm{MC}}} \leqslant 1\%\right)$.

5.1.2. Hat function

The Hat function is another commonly used benchmark problem for reliability analysis. It consists of a polynomial function that reads:

$$g(\mathbf{x}) = 12 - (x_1 - x_2)^2 - 8\,(x_1 + x_2 - 4)^3. \quad (20)$$

The input random vector consists of two independent normally distributed random variables, $X_1, X_2 \sim \mathcal{N}(0.25, 1)$. The reference noise-free probability of failure is $9.761 \cdot 10^{-4}$ $\left(CV_{P_{f,\mathrm{MC}}} \leqslant 1\%\right)$.

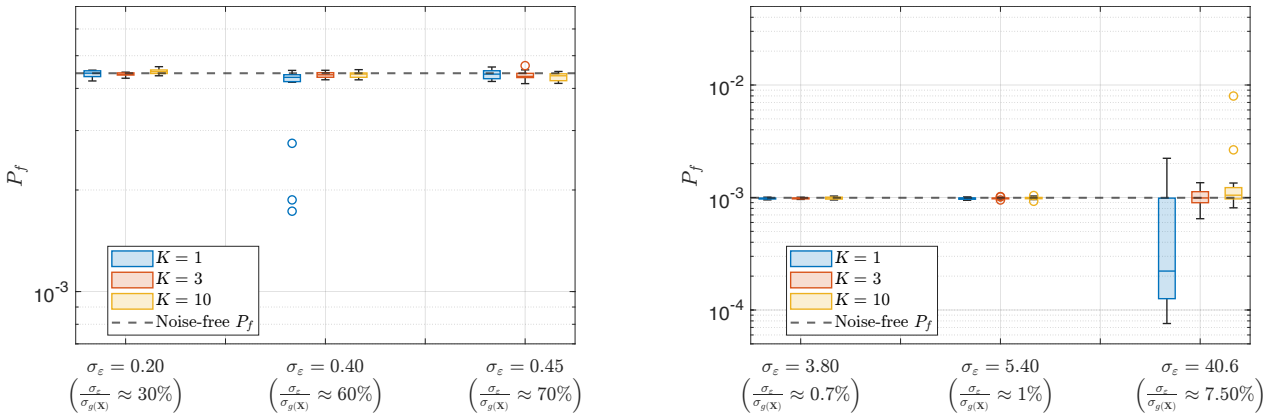
This benchmark problem was chosen because its topology features flat gradients around the limit state surface and steep gradients in other regions. Such a feature complicates the denoising of the problem, as it favours the misclassification of points around the limit state surface.

5.2. Results – PCE

Consistently with Marelli and Sudret (2018), the initial experimental design was obtained by a space-filling LHS sample of $N_{ini} = 10$ points. The number of bootstrap replications was $B = 100$. The degree of the sparse adaptively selected polynomial basis ranged in $p = 1, \ldots, 10$, with q-norm $\in \{0.5, 1\}$. The optimal base was identified through the least angle regression method (Blatman and Sudret, 2011; Lüthen et al., 2023). Finally, all analyses were performed using the UQLab software (Marelli and Sudret, 2014).

Figure 2 depicts the results obtained employing the discussed methodology for different noise levels. The ratio between the noise level, σ_ε and the standard deviation of the noise-free associated model $\sigma_{g(\mathbf{X})}$ is shown for reference. In Figures 2a and 2b, different boxplots represent distinct enrichment strategies, *i.e.* $K = 1, 3$ or 10. Each experiment was repeated 15 times to account for the statistical variability. The stopping criterion was consistent among the experiments, independently of the enrichment strategy.

The results show that, in contrast to MCS (Figure 1), the methodology converges to the noise-free reference probability of failure. Indeed, for the four-branch limit state function, the method reveals robust, converging to the reference probability of failure even for extremely noisy models. Despite this, Figure 2a reveals several outliers, suggesting that the learning function did not always find all failure regions. Aiming at improving the explorative behaviour of the learning function, tests with multiple enrichment points were carried out, leading to a



(a) Four-branch limit state function *(b) Hat limit state function*

Figure 2: Boxplots depicting results of the denosing performed with PCE.

significant improvement in the performance of the algorithm.

5.3. Results – GPR

We initialized the GPR surrogate model with the same initial ED used for the previous analysis. Moreover, the metamodel was modelled with a constant trend (ordinary Kriging) and an anisotropic ellipsoidal Matérn ($\nu = 5/2$) correlation function.

Figure 3 shows the results obtained using the GPR surrogate. Similar to previous results, Figure 3a depicts inaccurate convergence for the four-branch function when not all failure regions have been encountered, in this case, for $K = 1$ or 3. However, for $K = 10$, the explorative behaviour of the learning function is boosted, and accurate convergence is observed. For this reason, we assess that the methodology can denoise the problem, but the introduction of noise significantly increases the tendency for the learning function to get stuck in specific areas of the domain, leading to inaccurate convergence.

In contrast, Figure 3b shows that accurate results are obtained even when $K = 1$. For this example, no improvement in the estimation is observed when the enrichment is carried out with more than one point simultaneously, which is the expected behaviour, as discussed in Sec. 3. We attribute the difference between the results of the two experiments to the different topologies of the limit state functions. As the four-branch function comprises four failure regions, it demands more explorative behaviour. On the other hand, the hat function contains a unique failure region, accurately discovered through exploitative loops.

6. Conclusions

We have introduced a new class of problems in the field of reliability analysis, represented by non-deterministic limit state functions corrupted with simulation noise. Such a feature, hinders the assessment of the associated noise-free probability of failure through usual simulation techniques, such as MCS. In addition, we have shown that the latter converges to a different probability of failure, in some cases considerably larger depending on the noise level.

Moreover, we show that well-established surrogate modelling techniques allow retrieving the actual probability of failure. However, the tested learning functions depict a suboptimal behaviour, as they tend to get stuck in exploitative loops, ultimately failing to explore some regions of the input space. Thus, to efficiently deal with noise-corrupted models, the development of dedicated learning functions is needed. Besides, suitable convergence criteria must be investigated, as the noise significantly affects the convergence curve. Despite that, from the current results, an academic validation of the methodology was observed. Applying the developed methodology to real problems considering credible noise levels and high-dimensional inputs remains to be done.

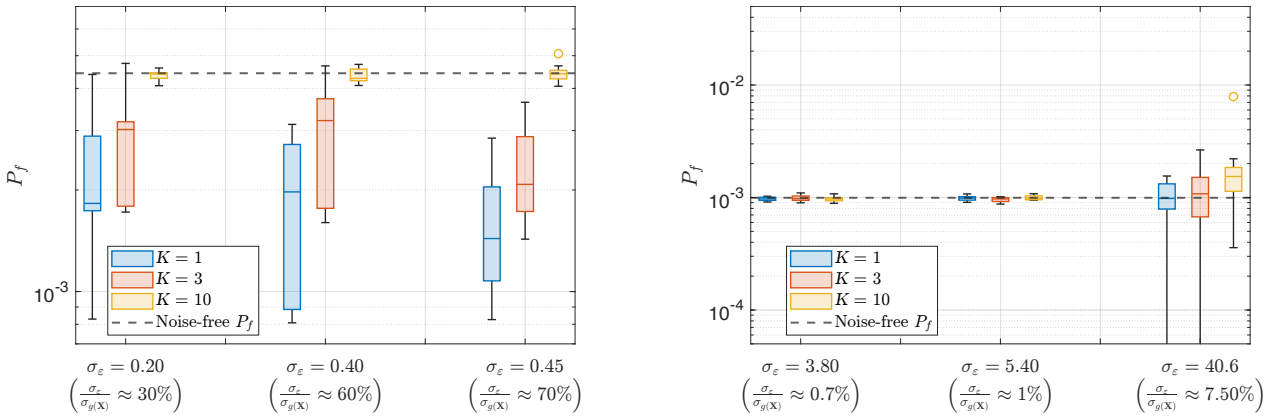
7. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 955393.

8. References

Ahmadisoleymani, S. S. and Missoum, S. (2021). "Stochastic crashworthiness optimization accounting for simulation noise." *Journal of Mechanical Design*, 144(5), 1–15.

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



(a) Four-branch limit state function

(b) Hat limit state function

Figure 3: Boxplots depicting results of the denoising performed with GPR.

Blatman, G. and Sudret, B. (2011). "Adaptive sparse polynomial chaos expansion based on least angle regression." *Journal of Computational Physics*, 230(6), 2345–2367.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014). "Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set." *Technometrics*, 56(4), 455–465.

Echard, B., Gayton, N., and Lemaire, M. (2011). "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation." *Structural Safety*, 33(2), 145–154.

Forrester, A., Keane, A., and Bressloff, N. (2006). "Design and analysis of "noisy" computer experiments." *American Institute of Aeronautics and Astronautics (AIAA)*, 44, 2331–2339.

Lüthen, N., Marelli, S., and Sudret, B. (2021). "Sparse polynomial chaos expansions: Literature survey and benchmark." *SIAM/ASA Journal on Uncertainty Quantification*, 9(2), 593–649.

Lüthen, N., Marelli, S., and Sudret, B. (2022). "Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications." *International Journal for Uncertainty Quantification*, 12(3), 49–74.

Lüthen, N., Marelli, S., and Sudret, B. (2023). "A spectral surrogate model for stochastic simulators computed from trajectory samples." *Computer Methods in Applied Mechanics and Engineering*, 406(115875), 1–29.

Marelli, S. and Sudret, B. (2014). "UQLab: A framework for uncertainty quantification in Matlab." *Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom)*, 2554–2563.

Marelli, S. and Sudret, B. (2018). "An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis." *Structural Safety*, 75, 67–74.

Melchers, R. E. (1989). "Importance sampling in structural systems." *Structural Safety*, 6, 3–10.

Moustapha, M., Marelli, S., and Sudret, B. (2022). "Active learning for structural reliability: Survey, general framework and benchmark." *Structural Safety*, 96, 102714.

Paz, J., Díaz, J., and Romera, L. (2020). "Analytical and numerical crashworthiness uncertainty quantification of metallic thin-walled energy absorbers." *Thin-Walled Structures*, 157, 107022.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Massachusetts, Internet edition.

Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc. (November).

Teixeira, R., Nogal, M., and O'Connor, A. (2021). "Adaptive approaches in metamodel-based reliability analysis: A review." *Structural Safety*, 89, 102019.

van den Eijnden, A. P., Schweckendiek, T., and Hicks, M. A. (2021). "Metamodelling for geotechnical reliability analysis with noisy and incomplete models." *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 16(3), 518–535.

Zaki, M. J. and Meira, W. J. (2014). *Data Mining and Analysis: fundamental Concepts and Algorithms*. Cambridge University Press.