14th International Conference on Structural Safety and Reliability - ICOSSAR'25
1-6 June, 2025 - Los Angeles California, USA
R. Ghanem, E. Johnson, S. Masri and A. Olivier (Eds)

RELIABILITY ANALYSIS OF PERFORMANCE FUNCTIONS VIA ADAPTIVE SEQUENTIAL SAMPLING WITH DETECTION OF FAILURE SURFACES

MIROSLAV VOŘECHOVSKÝ

Institute of Structural Mechanics, Brno University of Technology
Veveří 95, 602 00, Brno, Czech Republic
miroslav.vorechovsky@vut.cz, https://www.fce.vutbr.cz/STM/vorechovsky.m/

Key words: ψ criterion, failure probability, active learning, gradient-free optimization, Kriging, importance sampling

Abstract. We propose an improved method for estimating rare event probabilities in computational models with smooth performance functions. Building on a previously developed robust strategy for generally non-smooth or discrete-state performance functions, we enhance scalability and efficiency by replacing the original nearest-neighbor surrogate with a Gaussian process regression model. This surrogate leverages numerical limit state values to preselect potential candidates in an active learning scheme that balances exploration and exploitation. The resulting method significantly reduces the number of required evaluations, particularly in low-dimensional problems, while extending applicability to higher-dimensional settings.

1 INTRODUCTION

We present a new, efficient, and robust method for estimating rare event probabilities in computational models of engineering products and processes. The standard problem setting is a limit state function $g(\boldsymbol{x})$ (also called a performance function or computational model) with continuously distributed random inputs $\boldsymbol{X}$ forming the random vector with dimension $N_{\rm var}$. The probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$ is assumed as known. Vector $\boldsymbol{X}$ is the input to the performance function of a system/process whose reliability is to be evaluated. Assume that for any realization (a point from the *design domain* or input domain, $\mathcal{D}$), $\boldsymbol{x} \in \mathcal{D}$, we can obtain, information as to whether or not the system fails, or generally about the system performance. The probability of failure $p_{\mathcal{F}}$ is then defined as

$$p_{\mathcal{F}} \equiv \int \cdots \int_{\mathcal{F}} f_{\boldsymbol{X}}(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = \int \cdots \int_{\mathcal{D}} \mathbf{1}_{\mathcal{F}}(\boldsymbol{x}) \, f_{\boldsymbol{X}}(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x}. \tag{1}$$

where the failure set (domain) $\mathcal{F}$ as a union of all regions within the design domain in which failure occurs, and limiting the integral to the failure domain is equivalent to using an indicator function $\mathbf{1}_{\mathcal{F}}(\boldsymbol{x})$ that equals one for $\boldsymbol{x} \in \mathcal{F}$ (failure event), and zero otherwise.

Note that if the classification function $\mathbf{1}_{\mathcal{F}}(\boldsymbol{x})$, or its approximation via some surrogate model, can be evaluated very fast, the failure probability in Eq. (1) can be estimated sampling (such as importance sampling); see, e.g., [1]. In order to build an approximate representation of a true performance

function, which is inexpensive to analyze via sampling strategies, and provides also estimation of the prediction uncertainty, we adopt the Gaussian process regression (a.k.a. Kriging) [2, 3].

The spectrum of existing methods for reliability estimation is very rich and they can be classified based on the way how and how much they use and also make targeted modifications to the three groups of input input information (i) the information about the limit state function $g(\boldsymbol{x})$, (ii) the density of the input vector $\boldsymbol{X}$, and (iii) the geometry and topology of the input space. Crude Monte Carlo sampling estimation of probabilistic integrals does not modify or make special use of any of these inputs which makes it very robust method, but also quite inefficient. Other methods utilize more or less one or more pieces of information and also make various kinds of assumptions about these inputs to improve their efficiency.

This work builds upon a previously developed, robust methodology designed for discrete-state performance functions presented in [1]. Our earlier approach utilized a crude surrogate model based on a nearest-neighbor strategy. To enhance this method, we proposed a novel active learning criterion that selects candidates from both exploratory and exploitation point sets. The exploratory points are strategically chosen to span selected probability levels, while the exploitation points target regions near existing design points with known different classifications (safe, failure, or no response). Although this algorithm is exceptionally robust and efficient in low-dimensional spaces, it faces scalability challenges: as the input dimension grows, the number of exploratory points required to achieve similar classification accuracy increases dramatically.

In this paper, we significantly improve the scalability and performance of this approach by employing a Gaussian process regression-based surrogate model. By incorporating the numerical values of the limit state function, we greatly enhance candidate selection and extend the applicability of our strategy to higher-dimensional problems. Furthermore, in low-dimensional settings, the improved surrogate further reduces the number of required sample points for accurate failure probability estimation and sensitivity analysis.

2 SEQUENTIAL ADAPTIVE *EXTENSION* OF THE EXPERIMENTAL DESIGN

The premise behind the development of the presented algorithm is that each evaluation of the system performance (limit state function) is very expensive. Crude Monte Carlo type sampling strategies simply throw sampling points (= integration "nodes") and do not necessarily reflect the structure of the problem being solved at all.

The set of N_{sim} points at which the true performance function has been evaluated so far is referred to as the *experimental design* (ED). The ED consists of a table with N_{sim} points, each described by N_{var} coordinates, along with a corresponding vector of N_{sim} real-valued responses obtained from the performance function. The algorithms proposed in this paper are designed to operate solely on this information. Specifically, no additional input is required for the two fundamental steps: (i) *extension* of the ED by selecting the most suitable candidate for evaluation using the expensive performance function, and (ii) *estimation* of the current failure probability using a temporary binary surrogate model (or a surrogate with discrete output states, as discussed earlier).

These two steps are illustrated in Fig. 1, which outlines the overall structure of the proposed method. It relies on the construction of a surrogate model via Gaussian process (GP) regression, which yields two outputs at any point $\boldsymbol{x}$: the predicted mean value $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$. The trained GP model serves two independent purposes: (i) it enables the filtering (or censoring) of the candidate set using the ratio $\sigma(\boldsymbol{x})/|\mu(\boldsymbol{x})|$, aiding in the selection of the next evaluation point, and (ii) it supports rapid classification (safe/failure) during the sampling-based estimation of failure probability

using the sign of $\mu(\boldsymbol{x})$.

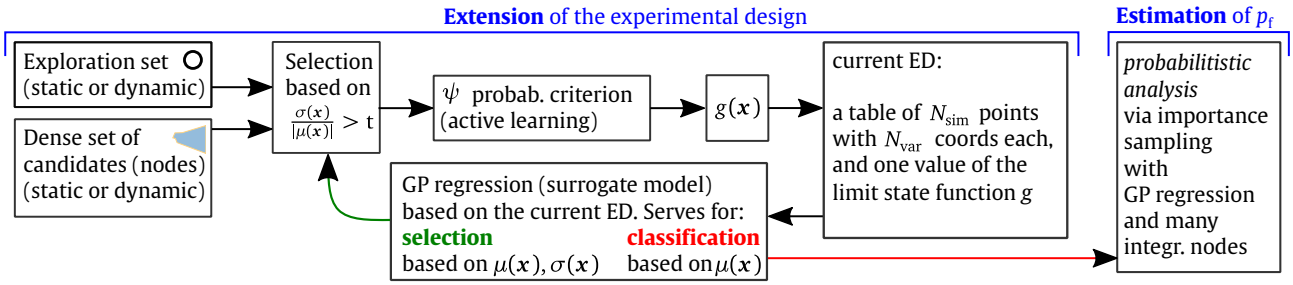

Figure 1: Flowchart of the proposed methodology consisting of two basic tasks: extension and estimation.

2.1 Exploration set (global)

We expect that a rare event (failure) region can appear anywhere. The local probability density of any potential failure region size decreases with increasing distance from the origin in the standard Gaussian space. Therefore, in a global search for failure domains, it makes sense to explore the space in a controlled fashion by checking various directions with an increasing distance from the origin. Therefore, we use our previously proposed [1] construction of a sequence of nested N_{var}-dimensional balls in the Gaussian space whose radii increase towards infinity and which occupy the probability in prescribed levels. In other words, we suggest preparing an *exploration set* for "onion-like" layers; see Fig. 2 left. Each layer is numbered by a level number i, and the corresponding N_{var}-ball is to be covered by a prescribed number of points n_i corresponding to this layer. The boundary of each layer is an N_{var}-ball, and the n_i points selected from the surface of this ball should be uniformly dispersed and it is preferable to avoid directionally collapsible candidates (all the candidates should have different radii vectors).

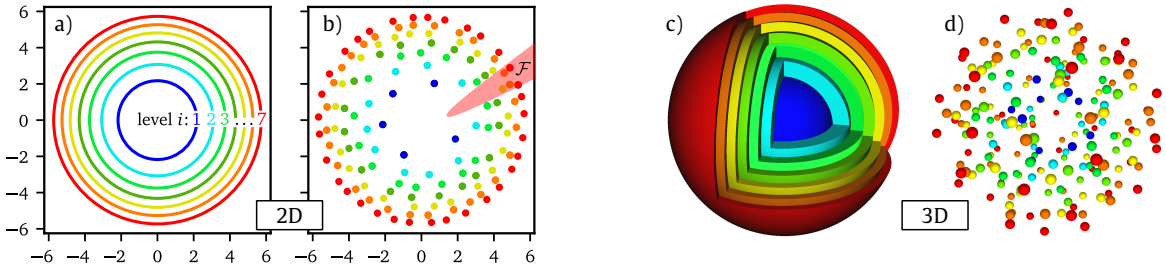

Figure 2: The first seven levels of exploration sets (layers) in $N_{\text{var}} = 2$ and $N_{\text{var}} = 3$ dimensional standard Gaussian space. Panels a,c: nested N_{var}-balls. Panels b,d: realizations of random exploration sets.

The probability content of the N_{var}-dimensional ball corresponding to layer i is chosen to be one tenth of that of the preceding ball. In the simplest case, the exploration set is pre-generated in advance for a given problem dimension. However, we note that the exploration set can be modified at any point during the execution of the proposed method. Detailed information on the construction of the exploration set can be found in [1].

2.2 Exploitation set (local)

A dense set of candidate points for future censoring and active learning selection can be generated using, for example, Quasi-Monte Carlo sampling methods (such as scrambled Sobol or Halton sequences) or conventional Monte Carlo sampling. These candidates can later also serve as integration nodes for failure probability estimation.

The candidate pool may be pre-generated at the beginning and kept fixed throughout the analysis. Alternatively, it can be dynamically enriched after each newly discovered failure point, in the spirit of importance sampling, using a Gaussian sampling density $h_i(\boldsymbol{x})$ centered at each rare-event ED point (with mean $\mu_i = \boldsymbol{x}_i$). The standard deviation is then determined by the problem dimension, set as $\sigma_i = \sqrt{N_{\text{var}} - 1}$

2.3 Censoring candidates via Gaussian process surrogate model

The aggregated pool of candidates consists of *exploration* points that provide controlled coverage of the entire domain, and *exploitation* points that form a dense set—potentially enriched near the failure region. From this large pool, it is necessary to identify points that meet one of the following criteria: (i) they lie near the failure surface, i.e., the boundary where the surrogate must approximate the true model accurately, or (ii) they are located in regions where the surrogate model is potentially inaccurate and would benefit from additional evaluations of the true performance function.

The first objective is addressed by selecting points with small absolute mean predictions, $|\mu(\boldsymbol{x})|$, while the second is met by targeting points with large prediction variance, $\sigma(\boldsymbol{x})$. To balance these objectives, we retain all candidates for which the ratio of the standard deviation to the absolute mean exceeds a threshold t:

$$\text{cov}_{\hat{g}}(\boldsymbol{x}) = \frac{\sigma(\boldsymbol{x})}{|\mu(\boldsymbol{x})|} > t\,, \tag{2}$$

where a typical choice is $t = 1$. Parameter t is the only free parameter affecting the behavior of the proposed method. Interpreting the GP surrogate prediction $\hat{g}(\boldsymbol{x})$ as a random variable, this criterion retains points where the absolute coefficient of variation exceeds one. This ensures the retained candidates are either close to the predicted failure surface or lie in regions with high uncertainty, away from previously evaluated ED points.

This censoring rule is closely related to the active learning function used in the AK-MCS method by Echard et al. [2], where the inverse of $\text{cov}_{\hat{g}}$ is employed directly for guiding the learning process. Specifically, they define:

$$U(\boldsymbol{x}) = \frac{|\mu(\boldsymbol{x})|}{\sigma(\boldsymbol{x})} \tag{3}$$

and select the candidate that minimizes this U function.

In our approach, the ratio is used solely for preselecting candidates for the next step—ranking by the ψ criterion described below, which is grounded in probabilistic arguments.

2.4 ψ criterion for candidate selection (active learnign)

Suppose the combined and censored exploration–exploitation set of candidates is available. The task is to select the candidate at which the (potentially costly) evaluation of the performance function yields the greatest gain in terms of probability classification—that is, in improving the separation of the input domain into failure and safe regions. This gain can result either from refining the current estimate of the failure boundary $\partial\mathcal{F}$, or from discovering a previously unexplored portion of it.

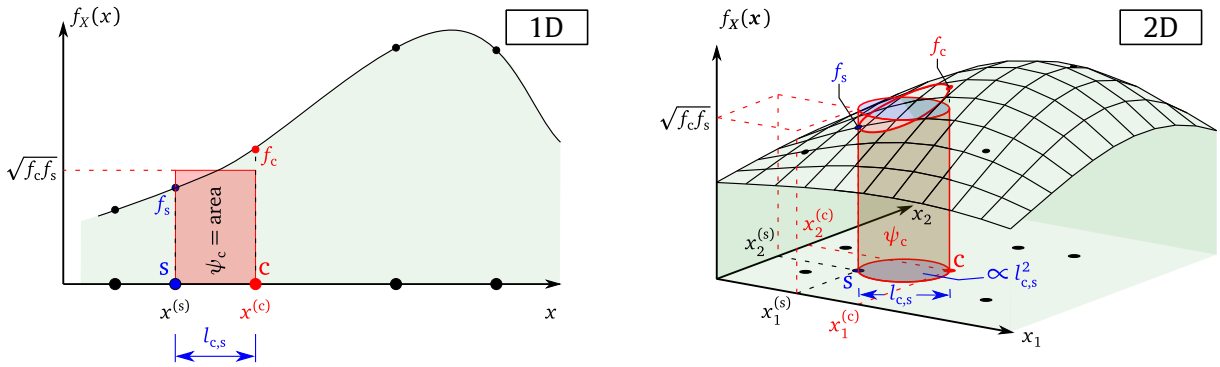

Figure 3: Geometrical meaning of the proposed ψ criterion for a candidate "c" in $N_{\text{var}} = 1$ and $N_{\text{var}} = 2$ dimensions. The black solid circles are the existing ED points and the blue circle "s" is the nearest neighbor to candidate "c". ψ_c is therefore proportional to the probability content of the red domain close to the candidate.

A key motivation behind the proposed algorithm is the principle that no two ED points should be placed close to one another unless doing so provides significant information in terms of probability content—specifically, by improving the boundary approximation between regions associated with different event types, such as the failure surface $\partial\mathcal{F}$. Furthermore, assuming the limit state function is smooth, a smooth interpolator (such as a Gaussian process) can approximate it well if the supporting points are not densely clustered, but instead are distributed along the failure surface in proportion to the local probability density.

For this purpose, we employ the previously proposed ψ criterion [1] to select the best candidate point. Maximizing this criterion approximately maximizes the instantaneous gain in probability content, since its value represents an estimate of the probability mass that would be newly classified by evaluating $g(\boldsymbol{x})$ at the candidate point.

The values of ψ can be interpreted as "bites" of probability with a clear geometric meaning: each candidate corresponds to a region in the design space (its neighborhood), and the volume of this region—when multiplied by the average probability density—yields the approximate probability content:

$$\psi_{\text{c}} = \underbrace{\sqrt{f_{\text{c}} f_{\text{s}}}}_{\text{ave probability}} \underbrace{(l_{\text{c,s}})^{N_{\text{var}}}}_{\propto \text{vol.}}, \tag{4}$$

where $\sqrt{f_{\text{c}} f_{\text{s}}}$ denotes the geometric mean of the (Gaussian) probability densities at the candidate point c and its nearest neighbor in the experimental design (ED), denoted s.

The geometric mean is used instead of the arithmetic mean to avoid overestimating the density contribution from candidates located far away in the Gaussian space: such remote points may have near-zero probability density, and an arithmetic mean could still yield a misleadingly nonzero value when combined with a higher-density ED point. This behavior would result in unreasonably large ψ values for distant candidates, leading to their selection—despite the fact that global exploration is already handled by the exploration set. Using the geometric mean naturally downweights such distant candidates and thus promotes more meaningful local refinement.

The term $l_{\text{c,s}}$ denotes the distance between a candidate point c and its nearest existing ED point s. In practice, the distances between each candidate and all N_{sim} currently evaluated ED points are computed, and each candidate is paired with its closest neighbor. To calculate these distances, a

suitable metric must be defined. In this paper, we use the Euclidean distance, given by

$$l_{\mathrm{c,s}} = \sqrt{\sum_{v=1}^{N_{\mathrm{var}}} \left| \boldsymbol{x}_v^{(\mathrm{c})} - \boldsymbol{x}_v^{(\mathrm{s})} \right|^2}. \tag{5}$$

When this distance is raised to the power of the problem dimension, i.e., $(l_{\mathrm{c,s}})^{N_{\mathrm{var}}}$, it becomes proportional to the volume of a geometrical region around the candidate. This region—whether interpreted as an N_{var}-ball, N_{var}-hypercube, or another shape—represents the spatial extent of the candidate's neighborhood in the design space. The exact shape of the region does not need to be specified, as all such volumes for different candidates are geometrically similar and share the same positive shape-dependent constant. This constant can be omitted from the criterion since it does not affect the relative ranking of candidates. The geometric interpretation of this criterion is illustrated in Fig. 3.

The criterion is formulated as the product of two independent terms:

- a volume term, which promotes rapid *exploration* of previously unvisited regions by expanding into areas of high uncertainty (reflected by a large numerator in the $\mathrm{cov}\hat{g}(\boldsymbol{x})$ term in Eq. (2)), and
- a probability term, which promotes *exploitation* of regions with high probability mass, particularly those near the origin and close to the failure surface (linked to the denominator in the $\mathrm{cov}\hat{g}(\boldsymbol{x})$ term).

This construction naturally balances exploration and exploitation. A candidate close to an existing ED point is only selected if its probability density is sufficiently high to yield greater gain than more distant candidates. Conversely, distant candidates are penalized due to their low density, unless their associated uncertainty justifies selection.

The criterion favors candidates that contribute to rapid classification of the design space. At any stage of the algorithm, the candidate selected for true function evaluation is the one with the highest value of ψ_{c} among all candidates under consideration.

It is beneficial to monitor the history of the ψ values during the sequential extension of the ED. The value of ψ provides a rough indication of how much the next evaluation might alter the current estimate of $p_{\mathcal{F}}$. This insight can be leveraged to formulate a *stopping criterion*. For instance, if ψ drops below a given threshold (e.g., $p_{\mathcal{F}}/100$), it may be reasonable to terminate the algorithm or switch to a different analysis phase. Empirical observations suggest that ψ values continue to decline as the computation proceeds, and the rate of decrease typically slows, indicating diminishing returns from further refinements of the boundary.

3 FAILURE PROBABILITY *ESTIMATION* VIA SAMPLING

At any stage of the ED extension process, the desired probabilities can be estimated based on pointwise information, i.e., the current ED with known $g(\boldsymbol{x})$ outcomes; see the right-hand part of Fig. 1. Indeed, a true computationally expensive model (a simulator) can be substituted by a computationally cheaper model. The surrogate model is constructed in a solely non-intrusive way with respect to the original simulator, i.e., it is purely data-driven. In our case, the GP surrogate model may used not only to guide the censoring of sample, but also to classify any point from $\mathcal{D}$ as either safe of failure event. In principle, one can built any other surrogate model based on the pointwise information in the ED not necessarily being able to provide also the uncertainty about the prediction, but reusing the mean prediction from the GP model does not require any additional computation.

The related event probability defined by Eq. (1) can today be estimated very efficiently, e.g., via importance sampling (IS). In IS the original joint density $f_{\boldsymbol{X}}(\boldsymbol{x})$ for sample selection is replaced by the importance sampling density $h(\boldsymbol{x})$, which we consider to be jointly Gaussian with independent marginals. Generating millions of realizations of Gaussian random variables takes fraction of a second on contemporary computers, and implementations are provided ready to use in various free numerical packages (for example the `NumPy` package [4] for powerful operations over multidimensional dimensional arrays in combination with vectorized functions and routines [5] from `scipy` available in `Python`).

The probability of an event type $\mathcal{T}$ occurring when the indicator function $\mathbf{1}_{\mathcal{T}}(\boldsymbol{X})$ signals it is defined as the expectation: $p_{\mathcal{T}} = \mathbb{E}\left[\mathbf{1}_{\mathcal{T}}(\boldsymbol{X})\right] = \int \cdots \int_{\mathcal{D}} \mathbf{1}_{\mathcal{T}}(\boldsymbol{x})\, f_{\boldsymbol{X}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$. Let $h(\boldsymbol{x})$ be the IS density which is positive wherever event $\mathcal{T}$ occurs. The probability of event $\mathcal{T}$ can be rewritten as

$$p_{\mathcal{T}} = \int_{\mathcal{D}} \frac{\mathbf{1}_{\mathcal{T}}(\boldsymbol{x})\, f_{\boldsymbol{X}}(\boldsymbol{x})}{h(\boldsymbol{x})} h(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \mathbb{E}_h\left[\frac{\mathbf{1}_{\mathcal{T}}(\boldsymbol{X})\, f_{\boldsymbol{X}}(\boldsymbol{X})}{h(\boldsymbol{X})}\right], \tag{6}$$

where $\mathbb{E}_h[\cdot]$ denotes the expectation for $\boldsymbol{X}$ being distributed according to h: $\boldsymbol{X} \sim h$. The IS estimation of $p_{\mathcal{T}}$ based on the current approximation of $\mathbf{1}_{\mathcal{T}}^{(N_{\mathrm{sim}})}(\boldsymbol{x})$ of the true indicator function $\mathbf{1}_{\mathcal{T}}(\boldsymbol{x})$ is made with n_{IS} integration nodes via the arithmetic average

$$p_{\mathcal{T}}^{(N_{\mathrm{sim}})} \approx \frac{1}{n_{\mathsf{IS}}} \sum_{i=1}^{n_{\mathsf{IS}}} \mathbf{1}_{\mathcal{T}}^{(N_{\mathrm{sim}})}(\boldsymbol{x}_i) \frac{f_{\boldsymbol{X}}(\boldsymbol{x}_i)}{h(\boldsymbol{x}_i)}, \quad \boldsymbol{X}_i \sim h. \tag{7}$$

The IS estimator is unbiased by construction. The variance of the IS estimator follows the standard definition

$$\mathrm{Var}_h[p_{\mathcal{T}}] = \mathbb{E}_h\left[\left(\frac{\mathbf{1}_{\mathcal{T}}(\boldsymbol{x})\, f_{\boldsymbol{X}}(\boldsymbol{x})}{h(\boldsymbol{x})}\right)^2\right] - {p_{\mathcal{T}}}^2 = \int_{\mathcal{D}} \left(\frac{\mathbf{1}_{\mathcal{T}}(\boldsymbol{x})\, f_{\boldsymbol{X}}(\boldsymbol{x})}{h(\boldsymbol{x})}\right)^2 h(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} - {p_{\mathcal{T}}}^2 \tag{8}$$

Its estimation using n_{IS} integration nodes $\boldsymbol{X}_i$ sampled from h, $i = 1, \ldots, n_{\mathsf{IS}}$, can again be performed using an arithmetic average

$$\mathrm{Var}_h[p_{\mathcal{T}}^{(N_{\mathrm{sim}})}] \approx \frac{1}{n_{\mathsf{IS}}} \left\{ \left[\frac{1}{n_{\mathsf{IS}}} \sum_{i=1}^{n_{\mathsf{IS}}} \mathbf{1}_{\mathcal{T}}^{(N_{\mathrm{sim}})}(\boldsymbol{x}_i) \frac{f_{\boldsymbol{X}}^2(\boldsymbol{x}_i)}{h^2(\boldsymbol{x}_i)}\right] - \left(p_{\mathcal{T}}^{(N_{\mathrm{sim}})}\right)^2 \right\} \tag{9}$$

and finally, the coefficient of variation of the estimation reads

$$\mathrm{CoV}_h[p_{\mathcal{T}}^{(N_{\mathrm{sim}})}] \approx \frac{\sqrt{\mathrm{Var}[p_{\mathcal{T}}^{(N_{\mathrm{sim}})}]}}{p_{\mathcal{T}}^{(N_{\mathrm{sim}})}} \tag{10}$$

A straightforward application of IS estimation is to use a Gaussian sampling density centered at the origin (i.e., with zero mean) and an inflated standard deviation, scaled according to the smallest distance between the failed ED point and the origin. While this approach is robust, it is not particularly efficient, as many integration nodes fall outside the region associated with the target event. Another drawback is the need to compute a weight ratio for each integration node.

In [1], we proposed two more advanced constructions of the sampling density, one of which is even proportional to the original Gaussian density but excludes a "safe hyperball" from the domain $\mathcal{D}$. From this point onward, we assume that the IS estimation task is solved, and we shift our focus to the efficiency of active learning in estimating failure probabilities with the fewest possible evaluations of the true limit state function $g(\boldsymbol{x})$.

4 DISCUSSION

The proposed methods fall into the category of active learning strategies for the sequential improvement of the limit state function approximation, particularly near the failure surface. While many methods have been introduced in recent literature, the most relevant for comparison is the AK-MCS method with the U criterion [2], as it employs a Kriging surrogate and leverages prediction uncertainty.

The primary distinction between the two methods lies in the active learning criterion and its (in)dependence on the density of proposed candidates. We argue that the methodology presented here automatically selects the density of experimental design (ED) points near the failure surface in proportion to the input density $f_{\boldsymbol{X}}(\boldsymbol{x})$, even when the candidate set densely covers the input space.

Our criticism of the AK-MCS method with the U criterion is that, when provided with an overly dense set of candidates, it tends to repeatedly select points for which the Kriging mean prediction $\mu(\boldsymbol{x})$ is nearly zero. This behavior can result in placing too many ED points in already-explored regions, incurring unnecessary costs for evaluating the true model $g(\boldsymbol{x})$.

In contrast, our method uses the local Kriging variance only for candidate preselection (or censoring), while the final ED points are chosen to maximize the associated probability expressed in the ψ criterion. As a result, the final distribution of ED points becomes independent of the density of the offered candidate set.

5 NUMERICAL EXAMPLE

In this conference contribution, we briefly present a single 2D example. The accompanying presentation will feature additional numerical examples chosen to illustrate various classes of problems that pose unique challenges, with a particular focus on scaling the method to higher-dimensional settings.

The failure surface used to demonstrate the *extension* and *estimation* steps in Fig. 4 is a sine wave of amplitude $a = 1$ superposed onto a circle of average radius $r_{\text{ave}} = 4$. The corresponding limit state function can be formulated as, e.g.,

$$g\left(x_1, x_2\right) = r_{\text{ave}} + a\sin(k\varphi_{\boldsymbol{x}}) - \|\boldsymbol{x}\|, \tag{11}$$

where $\|\boldsymbol{x}\| = \sqrt{x_1^2 + x_2^2}$ is the Euclidean distance from the origin, $\varphi_{\boldsymbol{x}} = \text{atan}_2(x_2, x_1) = \arctan(x_2/x_1)$ is the angle of vector $\boldsymbol{x}$ from axis x_1, and $k = 7$ is the number of design points. The sum of the first two terms represents the maximum safe distance from the origin, and the rare event (failure) occurs "behind" the wavy boundary, i.e., when $g \leq 0$. The presented algorithm uses only binary information (failure or success). The seven most central failures are located at an identical distance of $r_{\text{ave}} - a = 3$. Figs. 4 and Figs. 5 present the situation after $N_{\text{sim}} = 82$ evaluations of the limit state function. The estimation of failure probability converges towards the exact failure probability result $p_{\mathcal{F}} \approx 2.582 \cdot 10^{-3}$.

We compare the proposed method with the AK-MCS method using the U criterion [2]. Both methods wwere started from an intial design with $N_{\text{sim}} = 5N_{\text{var}} = 10$ points selected from the Gaussian density. While both methods progressively identify all seven regions that contribute most significantly to the failure probability, a closer examination reveals that the proposed method is more effective at placing evaluation points within high-probability regions. As a result, it achieves a more accurate failure probability estimate for the same number of function evaluations.

This improved efficiency is due to the use of the ψ criterion, which guides the selection of candidates in a way that naturally alternates between the most relevant regions, with frequencies propor-

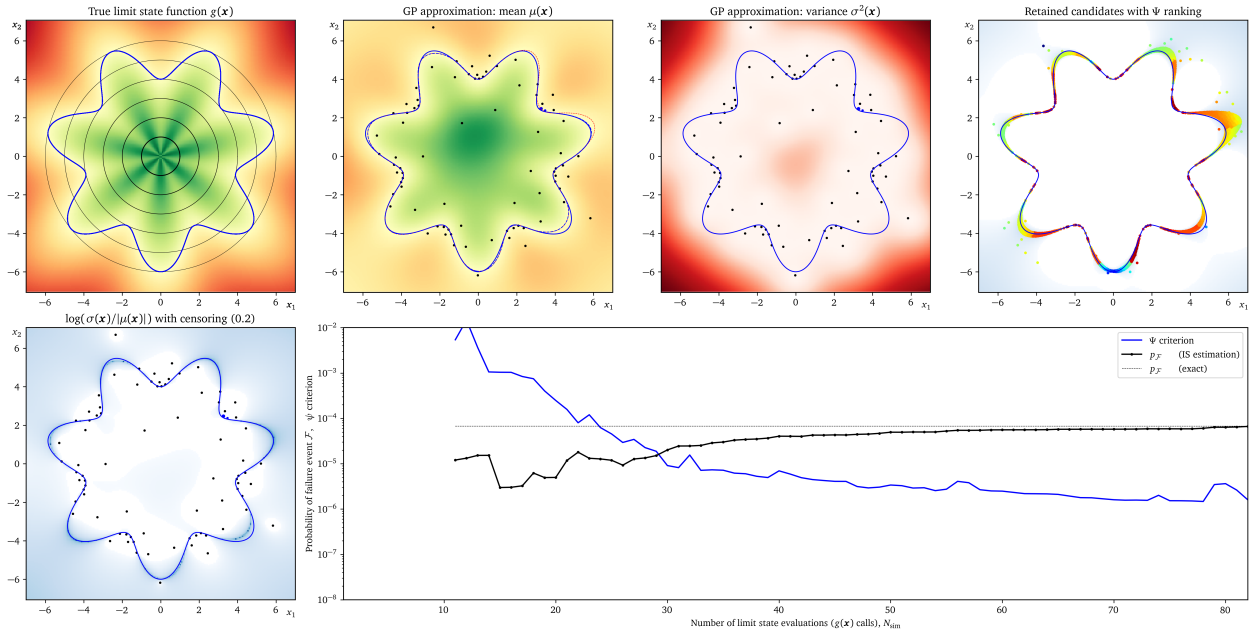

Figure 4: State after $N_{\text{sim}} = 82$ function evaluations in the proposed method. Top row: on-the-fly classification of territories. Top left panel: true limit state function via green to red color with the exact failure surface by blue color. The two following panels are the mean prediction and the prediction variance and the top right panel show the currently preselected candidate via cov, colored by with their ψ criterion. Bottom left panel shows the censoring by cov threshold set to $t = 0.2$. The large bottom panel is the history of estimation of failure probability and ψ criterion. quantifying the approximate "probability bites" by selected candidates.

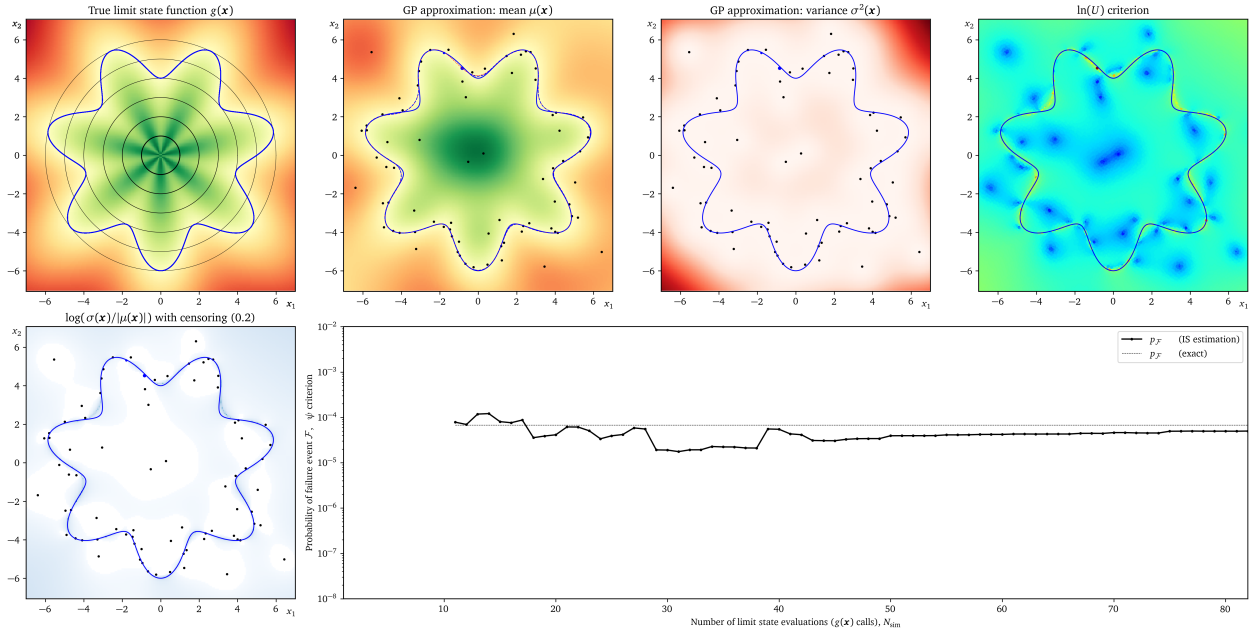

Figure 5: State after $N_{\text{sim}} = 82$ function evaluations in ACK-MS method with U criterion. The only difference in panel content is in the top right panel where we plot the current value of U criterion for all candidates.

tional to the Gaussian input density $f_{\boldsymbol{X}}(\boldsymbol{x})$.

We used OpenTURNS software [6] to run various techniques for failure probability estimation. The gradient-based design point search for FORM analysis initiated at a random location needs, after some help, 33 model evaluations to discover one of the design points, and thus FORM approximates the probability incorrectly as $\Phi(-3) \approx 1.35 \cdot 10^{-3}$. The SORM analysis increases the number of limit state function calls because it computes the failure surface curvature about the design point and decreases the estimation by about $3.2 \cdot 10^{-4}$ (the values slightly differ depending on the method used: Tvedt, Hohenbichler, and Breitung).

The results obtained with SuS implemented in OpenTURNS very much depend on the computational budget. If the number of limit state function evaluations reserved for each probability level is sufficiently high (a few thousand), the stochastic gradient optimization is able to correctly locate all seven failure regions and estimate the failure probability accurately. When, however, the total number of function calls drops below roughly one thousand, the estimate becomes incorrect. Similar statements are true about results obtained from the Adaptive Directional Stratification Algorithm in OpenTURNS: the number of function evaluations must be in the thousands for the method to provide sufficiently good results. For this function, the proposed technique provides better efficiency than the above methods because it provides more accurate estimations with considerably fewer function calls.

6 CONCLUSIONS

This paper presents an innovative method for rare event probability estimation, combining the strengths of Gaussian process (GP) surrogate modeling with a novel, probabilistically motivated criterion for active learning. Unlike existing methods such as AK-MCS, which rely on direct functions of prediction uncertainty, the proposed approach uses prediction uncertainty only for preselecting candidates, while the final selection is driven by the ψ criterion—a new formulation grounded in geometric and probabilistic arguments. This results in highly effective learning of the GP surrogate with significantly fewer function evaluations.

The method represents a major advancement over our previous work [1], which addressed a more challenging setting where the limit state function returned only discrete outputs (e.g., binary safe/failure or non-responses). That earlier method, while robust, suffered from scalability issues in higher dimensions due to the lack of informative surrogate structure.

The approach proposed here directly addresses that limitation. By leveraging the smoothness of the GP model, it achieves excellent efficiency, requiring a remarkably small number of function evaluations even in high-dimensional problems. The ψ criterion automatically determines optimal ED point placement without any dependence on the density of the candidate point set, enabling robust and efficient performance even in expensive black-box simulations.

This method is particularly well-suited for computationally expensive models with smooth performance functions—such as those involving finite element simulations—where traditional sampling-based approaches like Subset Simulation or Importance Sampling become computationally prohibitive.

ACKNOWLEDGEMENT

This work was supported by Ministry of Education, Youth and Sports of the Czech Republic under project no. LUAUS24260.

References

[1] M. Vořechovský, Reliability analysis of discrete-state performance functions via adaptive sequential sampling with detection of failure surfaces, *Computer Methods in Applied Mechanics and Engineering*, (2022) 401:115606. doi:10.1016/j.cma.2022.115606.

[2] B. Echard, N. Gayton, M. Lemaire, AK-MCS: An active learning reliability method combining Kriging and Monte Carlo simulation, *Structural Safety*, (2011) 33 (2):145–154. doi:10.1016/j.strusafe.2011.01.002.

[3] J. Wang, G. Xu, Y. Li, A. Kareem, AKSE: A novel adaptive Kriging method combining sampling region scheme and error-based stopping criterion for structural reliability analysis, *Reliability Engineering & System Safety*, (2022) 219:108214. doi:10.1016/j.ress.2021.108214.

[4] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, *Nature*, (2020) 585 (7825):357–362. doi:10.1038/s41586-020-2649-2.

[5] R. Chudoba, V. Sadílek, R. Rypl, M. Vořechovský, Using Python for scientific computing: an efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs, *Computer Physics Communications*, (2013) 184 (2):414–427. doi:10.1016/j.cpc.2012.08.021.

[6] M. Baudin, A. Dutfoy, B. Iooss, A.-L. Popelin, OpenTURNS: An industrial software for uncertainty quantification in simulation, in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer International Publishing, Cham, 2016, pp. 1–38. doi:10.1007/978-3-319-11259-6_64-1.