Semi-Bayesian active learning quadrature for estimating extremely low failure probabilities

Chao Dang[a,*], Michael Beer[a,b,c]

[a] *Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany*
[b] *Institute for Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZF, United Kingdom*
[c] *International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, PR China*

Abstract

The Bayesian failure probability inference (BFPI) framework provides a sound basis for developing new Bayesian active learning reliability analysis methods. However, it is still computationally challenging to make use of the posterior variance of the failure probability. This study presents a novel method called 'semi-Bayesian active learning quadrature' (SBALQ) for estimating extremely low failure probabilities, which builds upon the BFPI framework. The key idea lies in only leveraging the posterior mean of the failure probability to design two crucial components for active learning - the stopping criterion and learning function. In this context, a new stopping criterion is introduced through exploring the structure of the posterior mean. Besides, we also develop a numerical integration technique named 'hyper-shell simulation' to estimate the analytically intractable integrals inherent in the stopping criterion. Furthermore, a new learning function is derived from the stopping criterion and by maximizing it a single point can be identified in each iteration of the active learning phase. To enable multi-point selection and facilitate parallel computing, the proposed learning function is modified by incorporating an influence function. Through five numerical examples, it is demonstrated that the proposed method can assess extremely small failure probabilities with desired efficiency and accuracy.

Keywords: Structural reliability analysis; Bayesian active learning; Stopping criterion; Learning function; Parallel computing

*Corresponding author
Email address: chao.dang@irz.uni-hannover.de (Chao Dang)

1. Introduction

In the context of probabilistic structural reliability analysis, a primary objective is to compute the so-called failure probability, which is defined through a multidimensional integral as follows:

$$P_f = \int_{\mathcal{X}} I(g(\boldsymbol{x}) \leq 0) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{X} = [X_1, X_2, \cdots, X_d] \in \mathcal{X} \subseteq \mathbb{R}^d$ is a vector of d basic random variables; $f_{\boldsymbol{X}}(\boldsymbol{x})$ denotes the joint probability density function (PDF) of $\boldsymbol{X}$; $g(\cdot)$ is the performance function (also known as the limit state function), which takes a non-positive value when a prescribed failure event occurs; $I(\cdot)$ is the indicator function: $I = 1$ if $g(\boldsymbol{x}) \leq 0$ and $I = 0$ otherwise. An analytical solution of Eq. (1) is always desirable, which, however, is rarely available in the majority of real-world scenarios. Therefore, the use of numerical approximation becomes essential to derive a failure probability estimate. A typical numerical solution scheme involves repeatedly eventuating the I-function (equivalently, the g-function) many times. However, it is worth noting that each evaluation of the g-function may take a long running time, which poses a significant computational challenge in probabilistic structural reliability analysis.

There exist many numerical methods for approximating the failure probability, such as crude Monte Carlo simulation (MCS) [1] and its variants (e.g., important sampling [2–4], subset simulation [5], directional simulation [6] and line sampling [7]), first- and second- order reliability methods [8–10] and surrogate-based methods (e.g., polynomial chaos expansions [11] and Kriging [12]), to name just a few. The reader is referred to [13] for a relatively comprehensive review of existing computational methods for probabilistic structural reliability analysis. Among the diverse paradigms, the active learning reliability approaches have attracted increasing attention in the last decade. In this context, two seminal works are the efficient global reliability analysis [14] and the active learning reliability method combining Kriging and MCS (AK-MCS) [15]. Since their inception, a large number of active learning reliability methods have been developed. To learn about those advancements, we direct the reader to [16–18]. It has been shown that active learning reliability analysis methods can yield accurate failure probability estimates with fewer g-function calls. This is advantageous, especially for computationally demanding problems.

In addition, another more recent category, collectively denoted as Bayesian active learning reliability analysis methods (e.g., [19–26]), has emerged in recent years. These methods feature a distinctive fusion of Bayesian statistical inference and active learning techniques, referred to simply as Bayesian active learning. A first attempt is the 'active learning probabilistic integration' (ALPI) method reported in [19]. This method frames the estimation of the failure probability integral as a Bayesian inference problem, and a learning function and stopping criterion are developed based on the posterior mean and an upper bound on the posterior variance of the failure probability. The ALPI method is further enhanced to assess small failure probabilities and use parallel computing, resulting in the 'parallel adaptive Bayesian quadrature' (PABQ) method [20]. In the work presented in [21], the authors introduce a principled Bayesian framework known as 'Bayesian Failure Probability Inference' (BFPI). A noteworthy contribution is that the posterior variance of the failure probability is derived. However, it cannot be expressed in a closed form and is expensive to evaluate numerically, making it challenging to use for active learning purposes. Consequently, efforts have been made to develop Bayesian active learning reliability analysis methods that do not rely solely on the posterior variance of the failure probability. For instance, parallel Bayesian probabilistic integration [25] and partially Bayesian active learning cubature (PBALC) [26] are such methods. Besides, the Bayesian active learning concept is also investigated in the context of line sampling for structural reliability analysis [22–24]. While considerable progress has been made, there is still significant room for improvement in making Bayesian active learning reliability analysis methods more effective tools for practical applications.

The objective of this paper is to develop a new Bayesian active learning reliability analysis method with the capability to assess extremely low failure probabilities and facilitate parallel computing. This method relies solely on the posterior mean of the failure probability, aligning with the idea introduced in the previous work [26]. The resulting method is termed 'semi-Bayesian active learning quadrature' (SBALQ). The primary contributions of this work can be summarized as follows. First, a novel stopping criterion is introduced based on exploring the structure of the posterior mean of the failure probability. Second, an effective numerical integration method called 'hyper-shell simulation' (HSS) is developed for approximating the two intractable integrals involved in the proposed stopping criterion. Third, new learning functions are

designed to enable multi-point section during the active learning phase, thus facilitating parallel distributed processing.

The rest of this paper is arranged as follows. In Section 2, we provide a brief review of two previous studies. The proposed SBALQ method is introduced in Section 3. Five numerical examples are investigated in Section 4 to illustrate the proposed method. This paper closes with some concluding remarks in Section 5.

2. Brief review of two related works

This section provides a brief overview of the BFPI framework [21] and the three PBALC methods [26], which are closely related to this work. Note that both the PBALC method and the proposed SBALQ method are set up in the standard normal space (called the $\mathcal{U}$ space). For consistency, we will reformulate the BFPI framework in the $\mathcal{U}$ space. To do so, let us first introduce a transformation T (e.g., Rosenblatt transformation and Nataf transformation) that can project the physical random vector $\boldsymbol{X}$ into a standard normal one $\boldsymbol{U}$, i.e., $\boldsymbol{U} = T(\boldsymbol{X})$, where $\boldsymbol{U} = [U_1, U_2, \cdots, U_d] \in \mathcal{U} \subseteq \mathbb{R}^d$ is a set of d i.i.d. standard normal variables. This allows us to define a transformed performance function $\mathcal{G}(\boldsymbol{U}) = g(T^{-1}(\boldsymbol{U}))$, where $\mathcal{G} = g \circ T^{-1}$ and T^{-1} represents the inverse transformation.

2.1. Bayesian failure probability inference

The BPFI framework recast in the $\mathcal{U}$ space begins by assigning a Gaussian process (GP) prior over the $\mathcal{G}$-function:

$$\mathcal{G}_0(\boldsymbol{u}) \sim \mathcal{GP}(m_{\mathcal{G}_0}(\boldsymbol{u}), k_{\mathcal{G}_0}(\boldsymbol{u}, \boldsymbol{u}')), \tag{2}$$

where $\mathcal{G}_0$ represents the prior distribution of $\mathcal{G}$ before seeing any observation data; $m_{\mathcal{G}_0}(\boldsymbol{u})$ and $k_{\mathcal{G}_0}(\boldsymbol{u}, \boldsymbol{u}')$ denote the prior mean and covariance functions, respectively, which can fully define the GP prior. Without loss of generality, the prior mean and covariance functions are assumed to be a constant and a squared exponential kernel, respectively:

$$m_{\mathcal{G}_0}(\boldsymbol{u}) = \beta, \tag{3}$$

$$k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}') = s_0^2 \exp\left(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{u}')\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{u}')^\top\right), \tag{4}$$

where $\beta \in \mathbb{R}$; $s_0 > 0$ is the standard deviation of the GP prior; $\boldsymbol{\Sigma} = \mathrm{diag}\left(l_1^2, l_2^2, \cdots, l_d^2\right)$, where diag and $l_i > 0$ are the diagonal operator and the length scale in the i-th dimension, respectively. The parameters collected in $\boldsymbol{\theta} = [\beta, s_0, l_1, l_2, \cdots, l_d]$ are refereed to as the hyper-parameters.

Now suppose that we obtain an observation dataset $\mathscr{D} = \{\mathscr{U}, \mathscr{Y}\}$, where $\mathscr{U} = \left\{\boldsymbol{u}^{(j)}\right\}_{j=1}^n$ is an $n \times d$ matrix with j-th row being $\boldsymbol{u}^{(j)} \in \mathcal{U}$ and $\mathscr{Y} = [y^{(1)}, y^{(2)}, \cdots, y^{(n)}]^\top$ is an $n \times 1$ vector with j-th element being $y^{(j)} = \mathcal{G}(\boldsymbol{u}^{(j)})$. Those hyper-parameters can be learned from $\mathscr{D}$ by maximizing the log-marginal likelihood:

$$\log p(\mathscr{Y}|\mathscr{U},\boldsymbol{\theta}) = -\frac{1}{2}\left[(\mathscr{Y}-\beta)^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}(\mathscr{Y}-\beta) + \log|\boldsymbol{K}_{\mathcal{G}_0}| + n\log 2\pi\right], \tag{5}$$

where $\boldsymbol{K}_{\mathcal{G}_0}$ denotes an $n \times n$ covariance matrix with its (i,j)-th entry being $k_{\mathcal{G}_0}(\boldsymbol{u}^{(i)}, \boldsymbol{u}^{(j)})$.

The posterior distribution of $\mathcal{G}$ conditional on $\mathscr{D}$ follows another GP:

$$\mathcal{G}_n(\boldsymbol{u}) \sim \mathcal{GP}(m_{\mathcal{G}_n}(\boldsymbol{u}), k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')), \tag{6}$$

where $\mathcal{G}_n$ represents the posterior distribution of $\mathcal{G}$ after seeing n observations; $m_{\mathcal{G}_n}(\boldsymbol{u})$ and $k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}')$ are the posterior mean and covariance functions, respectively, which can given by:

$$m_{\mathcal{G}_n}(\boldsymbol{u}) = m_{\mathcal{G}_0}(\boldsymbol{u}) + \boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\mathscr{U})^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}\left(\mathscr{Y} - \boldsymbol{m}_{\mathcal{G}_0}(\mathscr{U})\right), \tag{7}$$

$$k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') = k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}') - \boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\mathscr{U})^\top \boldsymbol{K}_{\mathcal{G}_0}^{-1}\boldsymbol{k}_{\mathcal{G}_0}(\mathscr{U},\boldsymbol{u}'), \tag{8}$$

where $\boldsymbol{m}_{\mathcal{G}_0}(\mathscr{U}) = [m_{\mathcal{G}_0}(\boldsymbol{u}^{(1)}), m_{\mathcal{G}_0}(\boldsymbol{u}^{(2)}), \cdots, m_{\mathcal{G}_0}(\boldsymbol{u}^{(n)})]^\top$; $\boldsymbol{k}_{\mathcal{G}_0}(\boldsymbol{u},\mathscr{U}) = [k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}^{(1)}), k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}^{(2)}), \cdots, k_{\mathcal{G}_0}(\boldsymbol{u},\boldsymbol{u}^{(n)})]^\top$; $\boldsymbol{k}_{\mathcal{G}_0}(\mathscr{U},\boldsymbol{u}') = [k_{\mathcal{G}_0}(\boldsymbol{u}^{(1)},\boldsymbol{u}'), k_{\mathcal{G}_0}(\boldsymbol{u}^{(2)},\boldsymbol{u}'), \cdots, k_{\mathcal{G}_0}(\boldsymbol{u}^{(n)},\boldsymbol{u}')]^\top$.

The posterior mean and variance of the failure probability $\mathcal{P}_f$ are expressed as:

$$m_{\mathcal{P}_{f,n}} = \int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}, \tag{9}$$

$$\begin{aligned}\sigma^2_{\mathcal{P}_{f,n}} = &\int_{\mathcal{U}}\int_{\mathcal{U}} \Phi_2\left(\begin{bmatrix}0\\0\end{bmatrix}; \begin{bmatrix} m_{\mathcal{G}_n}(\boldsymbol{u}) \\ m_{\mathcal{G}_n}(\boldsymbol{u}')\end{bmatrix}, \begin{bmatrix}\sigma^2_{\mathcal{G}_n}(\boldsymbol{u}) & k_{\mathcal{G}_n}(\boldsymbol{u},\boldsymbol{u}') \\ k_{\mathcal{G}_n}(\boldsymbol{u}',\boldsymbol{u}) & \sigma^2_{\mathcal{G}_n}(\boldsymbol{u}')\end{bmatrix}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\phi_{\boldsymbol{U}}(\boldsymbol{u}')\mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{u}' \\ &- \left[\int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right]^2,\end{aligned} \tag{10}$$

where Φ and Φ_2 denote the cumulative distribution function (CDF) of the standard normal variable and bivariate normal CDF, respectively; $\phi_{\boldsymbol{U}}(\cdot)$ is the joint PDF of $\boldsymbol{U}$; $\sigma^2_{\mathcal{G}_n}(\cdot)$ represents the posterior variance function of $\mathcal{G}$, i.e., $\sigma^2_{\mathcal{G}_n}(\cdot) = k_{\mathcal{G}_n}(\cdot,\cdot)$.

The BFPI framework actually provides a probabilistic prediction for the failure probability $\mathcal{P}_f$, though the exact distribution is unknown. The posterior mean $m_{\mathcal{P}_{f,n}}$ is a natural point estimate for $\mathcal{P}_f$, while the posterior variance $\sigma^2_{\mathcal{P}_{f,n}}$ can measure our uncertainty of the estimate. Note that both $m_{\mathcal{P}_{f,n}}$ and $\sigma^2_{\mathcal{P}_{f,n}}$ are analytically intractable. Compared to $m_{\mathcal{P}_{f,n}}$, $\sigma^2_{\mathcal{P}_{f,n}}$ is harder to approximate because it involves a more complex integral.

2.2. Partially Bayesian active learning cubature

The three PBALC methods (denoted as PBALC1, PBALC2, and PBALC3) further embed the BFPI framework in an active learning circle. The underlying idea is to use only the posterior mean $m_{\mathcal{P}_{f,n}}$ to design crucial components for Bayesian active learning (i.e., stopping criterion and learning function), thus avoiding the need to deal with the posterior variance $\sigma^2_{\mathcal{P}_{f,n}}$. The resulting three sets of stopping criteria and learning functions are summarized in Table 1. Note that the key to achieving these results is to examine the numerator of the fractional term inherent in the posterior mean of the failure probability $m_{\mathcal{P}_{f,n}}$.

Table 1: Stopping criteria and learning functions developed in PBALC1, PBALC2, and PBALC3.

Method	Stopping criterion	Learning function
PBALC1	$\frac{\int_{\mathcal{U}}\left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)-\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}-b\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}{\int_{\mathcal{U}}\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}} < \epsilon_1$	$\mathrm{LSC}(\boldsymbol{u}) = \left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})} - b\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})$
PBALC2	$\frac{\int_{\mathcal{U}}\left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}+b\right)-\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}{\int_{\mathcal{U}}\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}} < \epsilon_2$	$\mathrm{RSC}(\boldsymbol{u}) = \left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})} + b\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})$
PBALC3	$\frac{\int_{\mathcal{U}}\left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}+b\right)-\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}-b\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}}{\int_{\mathcal{U}}\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}} < \epsilon_3$	$\mathrm{LSRSC}(\boldsymbol{u}) = \left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})} + b\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})} - b\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})$

Note that: b is a critical value that determines the desired confidence level; ϵ_1, ϵ_2 and ϵ_3 are three user-defined thresholds.

3. Semi-Bayesian active learning quadrature

In this section, we present another novel Bayesian active learning method, called SBALQ, which is based on the BFPI framework. This method is tailored for conducting structural reliability analysis with extremely low failure probabilities. The core idea of the proposed method aligns with that of the three previously developed PBALC methods. Specifically, we sorely use the posterior mean $m_{\mathcal{P}_{f,n}}$ to formulate both the stopping criterion and learning function. However, this study focuses on the denominator rather than the numerator in the fractional term involved in the posterior mean of the failure probability. Besides, the proposed method allows for parallel computing while the PBALC methods do not.

3.1. Stopping criterion and its numerical solution

The stopping criterion is of critical importance within a Bayesian active learning reliability analysis method, as it determines when to stop. In this context, we are looking for a stopping criterion that can judge whether the posterior mean $m_{\mathcal{P}_{f,n}}$, which serves as a failure probability estimate, reaches a desired level of accuracy. However, it is difficult to establish such a stopping criterion based on the posterior mean $m_{\mathcal{P}_{f,n}}$ without invoking the posterior variance $\sigma^2_{\mathcal{P}_{f,n}}$. To overcome this dilemma, a possible means is to explore the structure of $m_{\mathcal{P}_{f,n}}$.

As seen from Eq. (9), the integrand of $m_{\mathcal{P}_{f,n}}$ contains the ratio of the posterior mean function $m_{\mathcal{G}_n}(\boldsymbol{u})$ to the posterior standard deviation function $\sigma_{\mathcal{G}_n}(\boldsymbol{u})$. Building upon this insight, we introduce a novel quantity by imposing a penalty on $\sigma_{\mathcal{G}n}(\boldsymbol{u})$. This quantity, denoted as $Q_n(p)$, is defined as follows:

$$Q_n(p) = \int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}, \tag{11}$$

where $0 < p < 1$ acts as a penalty factor. Consequently, we can define the absolute difference of $m_{\mathcal{P}_{f,n}}$ and $Q_n(p)$, denoted as $\Delta_n(p)$:

$$\begin{aligned}\Delta_n(p) =& |m_{\mathcal{P}_{f,n}} - Q_n(p)| \\ =& \left|\int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} - \int_{\mathcal{U}} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right| \\ =& \left|\int_{\mathcal{U}} \left[\Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\right]\phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}\right|.\end{aligned} \tag{12}$$

Further, we can establish an upper bound for $\Delta_n(p)$, denoted as $\overline{\Delta}_n(p)$:

$$\overline{\Delta}_n(p) = \int_{\mathcal{U}} \left| \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\right| \phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u}. \tag{13}$$

To proceed, we examine the limit of $\overline{\Delta}_n(p)$:

$$\begin{aligned}
\mathcal{L}_n &= \lim_{p\to 0} \overline{\Delta}_n(p) \\
&= \lim_{p\to 0} \int_{\mathcal{U}} \left| \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) - \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{p\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\right| \phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\
&= \int_{\mathcal{U}} \left| \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) - I(m_{\mathcal{G}_n}(\boldsymbol{u}))\right| \phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\
&= \int_{\mathcal{U}} \Phi\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u},
\end{aligned} \tag{14}$$

where I is a indicator function: $I(m_{\mathcal{G}_n}(\boldsymbol{u})) = 1$ if $m_{\mathcal{G}_n}(\boldsymbol{u}) < 0$ and $I(m_{\mathcal{G}_n}(\boldsymbol{u})) = 0$ otherwise. Note that it is easy to show that as $\sigma_{\mathcal{G}_n}(\boldsymbol{u}) \to 0^+$ and $m_{\mathcal{G}_n}(\boldsymbol{u}) \to \mathcal{G}(\boldsymbol{u})$, there exist $\mathcal{L}_n \to 0^+$ and $m_{\mathcal{P}_{f,n}} \to \mathcal{P}_f$.

Based on these findings, we can formulate the stopping criterion as:

$$\frac{\mathcal{L}_n}{m_{\mathcal{P}_{f,n}}} < \epsilon, \tag{15}$$

where ϵ is a user-defined threshold, which should be a very small positive real number. This stopping criterion implies that the proposed method stops when the value of $\mathcal{L}_n$ becomes significantly smaller relative to $m_{\mathcal{P}_{f,n}}$. The choice of the stopping criterion is rather natural, since it actually constrains the upper bound of the relative error between $m_{\mathcal{P}_{f,n}}$ and $\lim_{p\to 0} Q_n(p)$. In the context of active learning reliability methods, similar stopping criteria have been reported in [27, 28]. The use of the proposed stopping criterion, however, is not straightforward because it involves two analytically intractable integrals.

In this study, we present a novel numerical integration method, called 'hyper-shell simulation' (HSS), to numerically approximate both $m_{\mathcal{P}_{f,n}}$ and $\mathcal{L}_n$. Our method draws inspiration from and builds upon some established methods, especially incorporating some key principles derived from the "importance ball sampling" (IBS) method [20] and "spherical decomposition - Monte Carlo simulation" (SD-MCS) method [29].

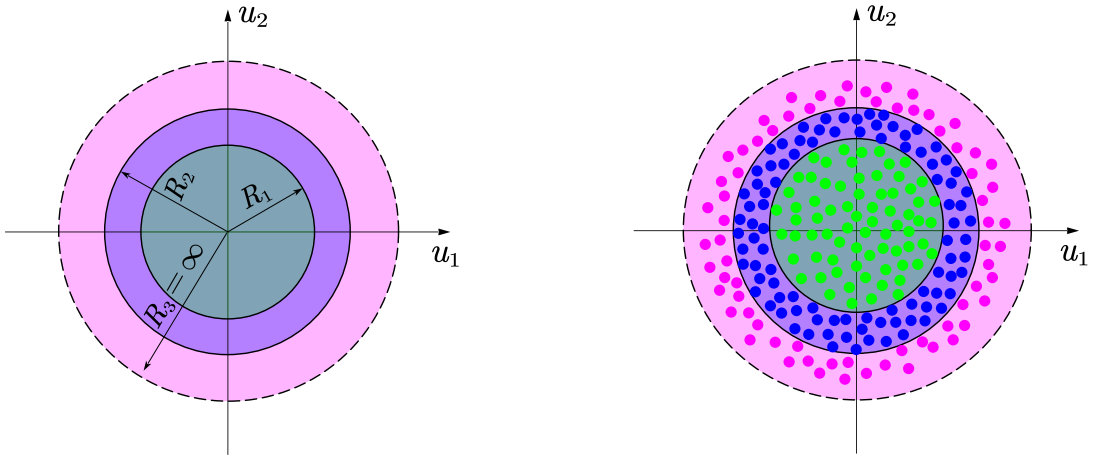
The HSS method begins by partitioning the standard normal space $\mathcal{U}$ into h concentric hyper-spherical

shells, following the SD-MCS method [29]:

$$\bigcup_{i=1}^{h} \mathcal{U}_i = \mathcal{U}, \tag{16}$$

$$\mathcal{U}_i \bigcap \mathcal{U}_j = \emptyset, i \neq j, \tag{17}$$

where $\mathcal{U}_i = \{\boldsymbol{u} | R_{i-1} \leq ||\boldsymbol{u}|| < R_i\}$ denotes the i-th hyper-shell, $i = 1, 2, \cdots, h$; R_{i-1} and R_i are the inner and outer radius of $\mathcal{U}_i$, respectively. The radii, $\{R_j\}_{i=0}^{h}$, forms an ascending sequence, i.e., $R_0 < R_1 < \cdots < R_h$. Apart from $R_0 = 0$ and $R_h = +\infty$, one has to choose R_i $(i = 1, 2, \cdots, m-1)$ properly, which can be specified as $R_i = \sqrt{\chi_d^{-2}(1 - 10^{-i})}$, where $\chi_d^{-2}(\cdot)$ denotes the inverse CDF of the chi-squared distribution with d degrees of freedom. A schematic representation of the space decomposition in two dimensions can be found in Fig. 1(a).



(a) Space decomposition (b) Subregion sampling

Figure 1: Illustration of the HSS method ($h = 3$) in two dimensions.

Building upon the space decomposition, we can rewrite $m_{\mathcal{P}_{f,n}}$ and $\mathcal{L}_n$ as follows:

$$\begin{aligned} m_{\mathcal{P}_{f,n}} &= \sum_{i=1}^{h} m_{\mathcal{P}_{f,n}}^{(i)} \\ &= \sum_{i=1}^{h} \int_{\mathcal{U}_i} \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}, \end{aligned} \tag{18}$$

$$\begin{aligned} \mathcal{L}_n &= \sum_{i=1}^{h} \mathcal{L}_n^{(i)} \\ &= \sum_{i=1}^{h} \int_{\mathcal{U}_i} \Phi\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}. \end{aligned} \tag{19}$$

Then, we introduce uniform sampling PDFs (denoted as $p^{(i)}(\boldsymbol{u}), i = 1, 2, \cdots, h-1$) for the first $h-1$ sub-regions and a truncated normal sampling PDF (denoted as $\psi^{(h)}(\boldsymbol{u})$) for the last sub-region such that:

$$p^{(i)}(\boldsymbol{u}) = \begin{cases} \frac{1}{v_i}, \boldsymbol{u} \in \mathcal{U}_i \\ 0, \text{otherwise} \end{cases}, \tag{20}$$

$$\psi^{(h)}(\boldsymbol{u}) = \begin{cases} \frac{\phi_{\boldsymbol{U}}(\boldsymbol{u})}{\delta_h}, \boldsymbol{u} \in \mathcal{U}_h \\ 0, \text{otherwise} \end{cases}, \tag{21}$$

where $v_i = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \left(R_i^d - R_{i-1}^d\right)$ is the volume of the i-th hyper-shell; $\delta_h = \int_{\mathcal{U}_h} \phi_{\boldsymbol{U}}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}$ is the probability content of the outermost hyper-shell $\mathcal{U}_h$. Accordingly, $m_{\mathcal{P}_{f,n}}$ and $\mathcal{L}_n$ can be further reformulated as:

$$\begin{aligned} m_{\mathcal{P}_{f,n}} &= \sum_{i=1}^{h} m_{\mathcal{P}_{f,n}}^{(i)} \\ &= \sum_{i=1}^{h-1} v_i \int_{\mathcal{U}_i} \boldsymbol{\Phi}\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}) p^{(i)}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} + \delta_h \int_{\mathcal{U}_h} \boldsymbol{\Phi}\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \psi^{(h)}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}, \end{aligned} \tag{22}$$

$$\begin{aligned} \mathcal{L}_n &= \sum_{i=1}^{h} \mathcal{L}_n^{(i)} \\ &= \sum_{i=1}^{h-1} v_i \int_{\mathcal{U}_i} \boldsymbol{\Phi}\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}) p^{(i)}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} + \delta_h \int_{\mathcal{U}_h} \boldsymbol{\Phi}\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right) \psi^{(h)}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}. \end{aligned} \tag{23}$$

The HSS estimators for $m_{\mathcal{P}_{f,n}}$ and $\mathcal{L}_n$ can be given by:

$$\begin{aligned} \hat{m}_{\mathcal{P}_{f,n}} &= \sum_{i=1}^{h} \hat{m}_{\mathcal{P}_{f,n}}^{(i)} \\ &= \sum_{i=1}^{h-1} v_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \boldsymbol{\Phi}\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i,j)}) \right) + \delta_h \frac{1}{N_h} \sum_{j=1}^{N_h} \boldsymbol{\Phi}\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}\right), \end{aligned} \tag{24}$$

$$\begin{aligned} \hat{\mathcal{L}}_n &= \sum_{i=1}^{h} \hat{\mathcal{L}}_n^{(i)} \\ &= \sum_{i=1}^{h-1} v_i \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \boldsymbol{\Phi}\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i,j)}) \right) + \delta_h \frac{1}{N_h} \sum_{j=1}^{N_h} \boldsymbol{\Phi}\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}\right), \end{aligned} \tag{25}$$

where $\left\{\boldsymbol{u}^{(i,j)}\right\}_{j=1}^{N_i}, i = 1, 2, \cdots, h-1$ is a set of N_i random samples generated according to $p^{(i)}(\boldsymbol{u})$; $\left\{\boldsymbol{u}^{(h,j)}\right\}_{j=1}^{N_h}$ is a set of N_h random samples drawn from $\psi^{(h)}(\boldsymbol{u})$. For how to generate these random numbers, please refer to Appendix A and Appendix B. In addition, one can examine Fig. 1(b) for a visual depiction of the sub-region sampling results.

The variances of the estimators can be expressed as:

$$\begin{aligned}\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right] =& \sum_{i=1}^{h} \mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}^{(i)}\right] \\ =& \sum_{i=1}^{h-1} \frac{1}{(N_i-1)\,N_i} \sum_{j=1}^{N_i} \left[v_i \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i,j)}) - \hat{m}_{\mathcal{P}_{f,n}}^{(i)} \right]^2 \\ &+ \frac{1}{(N_h-1)\,N_h} \sum_{j=1}^{N_h} \left[\delta_h \Phi\left(-\frac{m_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}\right) - \hat{m}_{\mathcal{P}_{f,n}}^{(h)} \right]^2,\end{aligned} \tag{26}$$

$$\begin{aligned}\mathrm{Var}\left[\hat{\mathcal{L}}_n\right] =& \sum_{i=1}^{h} \mathrm{Var}\left[\hat{\mathcal{L}}_n^{(i)}\right] \\ =& \sum_{i=1}^{h-1} \frac{1}{(N_i-1)\,N_i} \sum_{j=1}^{N_i} \left[v_i \Phi\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(i,j)})}\right) \phi_{\boldsymbol{U}}(\boldsymbol{u}^{(i,j)}) - \hat{\mathcal{L}}_n^{(i)} \right]^2 \\ &+ \frac{1}{(N_h-1)\,N_h} \sum_{j=1}^{N_h} \left[\delta_h \Phi\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u}^{(h,j)})}\right) - \hat{\mathcal{L}}_n^{(h)} \right]^2,\end{aligned} \tag{27}$$

where $\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}^{(i)}\right]$ and $\mathrm{Var}\left[\hat{\mathcal{L}}_n^{(i)}\right]$ are called partial variances and $\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]$ and $\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]$ are called total variances.

One of the salient features of the proposed HSS method is that it allows us to generate more samples in the hyper-shell with the largest partial variance, thus reducing the total variance more effectively. This can be achieved by the following procedure:

Step A1: Generate ΔN initial samples in each hyper-shell, based on which the partial and total means and variances are calculated.

Step A2: Check whether the coefficients of variation are small enough, i.e., $\sqrt{\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} < \gamma_1$ and $\sqrt{\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]}/\hat{\mathcal{L}}_n < \gamma_2$, where γ_1 and γ_2 are two user-specified thresholds.

- If $\sqrt{\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} < \gamma_1$ and $\sqrt{\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]}/\hat{\mathcal{L}}_n < \gamma_2$ are satisfied, then terminate the HSS method;
- If $\sqrt{\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} \geq \gamma_1$ and $\sqrt{\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]}/\hat{\mathcal{L}}_n < \gamma_2$, then identify the hyper-shell with the largest partial variance for $\hat{m}_{\mathcal{P}_{f,n}}$ as $i^\star = \arg\max_{1\leq i\leq h} \mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}^{(i)}\right]$ and go to **Step A3**;
- If $\sqrt{\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} < \gamma_1$ and $\sqrt{\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]}/\hat{\mathcal{L}}_n \geq \gamma_2$, then identify the hyper-shell with the largest partial variance for $\hat{\mathcal{L}}_n$ as $i^\star = \arg\max_{1\leq i\leq h} \mathrm{Var}\left[\hat{\mathcal{L}}_n^{(i)}\right]$ and go to **Step A3**;
- If $\sqrt{\mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}\right]}/\hat{m}_{\mathcal{P}_{f,n}} \geq \gamma_1$ and $\sqrt{\mathrm{Var}\left[\hat{\mathcal{L}}_n\right]}/\hat{\mathcal{L}}_n \geq \gamma_2$, then identify the two hyper-shells with the largest partial variances for $\hat{m}_{\mathcal{P}_{f,n}}$ and $\hat{\mathcal{L}}_n$ as $i_1^\star = \arg\max_{1\leq i\leq h} \mathrm{Var}\left[\hat{m}_{\mathcal{P}_{f,n}}^{(i)}\right]$ and $i_2^\star = \arg\max_{1\leq i\leq h} \mathrm{Var}\left[\hat{\mathcal{L}}_n^{(i)}\right]$,

and go to **Step A4**;

Step A3: Start by generating an additional set of ΔN samples for the $i^\star$-th hyper-shell. Afterward, update the partial and total means and variances, and proceed to **Step A2**;

Step A4: For both the $i_1^\star$-th and $i_2^\star$-th hyper-shells, generate an additional set of ΔN samples for each. Subsequently, update the partial and total means and variances before proceeding to **Step A2**;

The numerator and denominator on the left-hand side of the stopping criterion (Eq. (15)) should thus be replaced by the corresponding final estimates $\hat{m}_{\mathcal{P}_{f,n}}$ and $\hat{\mathcal{L}}_n$. To avoid false convergence, the stopping criterion needs to be satisfied twice in a row.

3.2. Learning function and multi-point selection

In addition to the stopping criterion, another pivotal component for developing a Bayesian active learning reliability analysis method is the so-called learning (or acquisition) function. This function guides the active learning process by selecting the most informative data points. These carefully chosen data points, once evaluated, are expected to yield the most significant improvement on the accuracy of the failure probability prediction. Assuming that the accuracy of the failure probability prediction can be effectively governed by the proposed stopping criterion (Eq. (15)), then the optimal learning function is one that minimizes the number of selected data points required to meet this criterion. Therefore, our underlying principle for designing the learning function is to align with the attainment of the stopping criterion. Besides, we also seek to develop a strategy that can identify multiple informative points instead of a single point in each iteration from the designated learning function, in order to facilitate parallel computing.

A new learning function, denoted as $\mathcal{J}_n$, which is derived from the proposed stopping criterion, is given as follows:

$$\mathcal{J}_n(\boldsymbol{u}) = \Phi\left(-\frac{|m_{\mathcal{G}_n}(\boldsymbol{u})|}{\sigma_{\mathcal{G}_n}(\boldsymbol{u})}\right)\phi_{\boldsymbol{U}}(\boldsymbol{u}). \tag{28}$$

Note that the equation $\int_{\mathcal{U}} \mathcal{J}_n(\boldsymbol{u})\mathrm{d}\boldsymbol{u} = \mathcal{L}_n$ holds. Consequently, we can interpret the learning function $\mathcal{J}_n$ as a measure that quantifies the contribution of the value at point $\boldsymbol{u}$ to the overall value of $\mathcal{L}_n$. Intuitively, the point with the largest $\mathcal{J}_n$-function value is likely to be the most promising candidate point to choose.

It is worth pointing out that some existing learning functions can be regarded as variants of our proposed $\mathcal{J}_n$ function, e.g., U function [15], expected misclassification probability contribution (EMPC) function [21] and learning functions in [27, 30].

Having established the learning function, we now turn our attention to the selection of multiple informative points from it. The method we are going to develop is strongly inspired by the previous work [31], where a multi-point expected improvement criterion was proposed for efficient global optimization. It is important to note that $\mathcal{J}_n$ is non-negative, and drops sharply to zero at those sampled points. This inherent property presents a promising avenue for developing a multi-point selection strategy. Suppose that in a given iteration of active learning, we wish to select an additional n_a points, in addition to the existing n points. The core of our strategy is to select the n_a points one at a time, rather than all at once. This can be achieved by sequentially adjusting the $\mathcal{J}_n$ function to account for the possible effects induced by the points that have been identified. To initiate the process, we begin by identifying the first point, denoted as $\boldsymbol{u}^{(n+1)}$, by maximizing the original $\mathcal{J}_n$ function:

$$\boldsymbol{u}^{(n+1)} = \underset{\boldsymbol{u}\in[-B,B]^d}{\arg\max}\ \mathcal{J}_n(\boldsymbol{u}), \tag{29}$$

where $[-B, B]^d$ is a d-dimensional hyperrectangle of side length B in the standard normal space. A convenient way to specify a reasonable value for B is according to $R = \sqrt{\chi_d^{-2}(1-\varepsilon)}$, where $\varepsilon = 1 \times 10^{-10}$ is adopted [26]. Then, one can choose the q-th point $\boldsymbol{u}^{(n+q)}$ by maximizing a pseudo $\mathcal{J}_{n+q}$ function (denoted as $\mathcal{SJ}_n^{(q)}$) such that:

$$\boldsymbol{u}^{(n+q)} = \underset{\boldsymbol{u}\in[-B,B]^d}{\arg\max}\ \mathcal{SJ}_n^{(q)}(\boldsymbol{u}; \boldsymbol{u}^{(n+1)}, \boldsymbol{u}^{(n+2)}, \cdots, \boldsymbol{u}^{(n+q-1)}). \tag{30}$$

Here $\mathcal{SJ}_n^{(q)}$ is used to approximate the real $\mathcal{J}_{n+q}$ function without evaluating the last $q-1$ points on the $\mathcal{G}$ function. The $\mathcal{SJ}_n^{(q)}$ function takes the following form:

$$\mathcal{SJ}_n^{(q)}(\boldsymbol{u}; \boldsymbol{u}^{(n+1)}, \boldsymbol{u}^{(n+2)}, \cdots, \boldsymbol{u}^{(n+q-1)}) = \mathcal{J}_n(\boldsymbol{u}) \times \prod_{j=1}^{q-1} IF(\boldsymbol{u}, \boldsymbol{u}^{(n+j)}), \tag{31}$$

where $IF(\boldsymbol{u}, \boldsymbol{u}^{(n+j)})$ is the influence function, defined by [31]:

$$\begin{aligned} IF(\boldsymbol{u}, \boldsymbol{u}^{(n+j)}) =& 1 - \rho(\boldsymbol{u}, \boldsymbol{u}^{(n+j)}) \\ =& 1 - \exp\left(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}^{(n+j)})\boldsymbol{\Sigma}^{-1}(\boldsymbol{u} - \boldsymbol{u}^{(n+j)})^{\top}\right), \end{aligned} \tag{32}$$

in which $\rho(\boldsymbol{u}, \boldsymbol{u}^{(n+j)})$ denotes the correlation coefficient function. By introducing the influence function, the $\mathcal{SJ}_n^{(q)}$ function takes a zero value at the $q-1$ already selected points $\boldsymbol{u}^{(n+1)}, \boldsymbol{u}^{(n+2)}, \cdots, \boldsymbol{u}^{(n+q-1)}$, and approaches to the original $\mathcal{J}_{n+q}$ function when far away from those points. This function is referred to as 'pseudo' because its primary purpose is to serve as an approximation for the true $\mathcal{J}_{n+q}$ function. To produce n_a points in a given iteration using the proposed strategy, it is necessary to perform n_a optimizations on the corresponding learning functions. Fortunately, this computational burden is typically much smaller than that of a single evaluation of the $\mathcal{G}$-function in practical scenarios.

3.3. Implementation procedure of the proposed method

The main steps for implementing the proposed SBALQ method are summarized below, alongside a flowchart shown in Fig. 2.

Step B1: Generate an initial observation dataset

To begin the proposed method, it is necessary to create an initial observation dataset by evaluating the $\mathcal{G}$-function. This can be achieved by first generating a small number (denoted as n_0) of uniformly distributed samples $\mathscr{U} = \left\{\boldsymbol{u}^{(j)}\right\}_{j=1}^{n_0}$ within a d-ball of radius B using a low-discrepancy sequence. In this study, the radius is specified as $B = \sqrt{\chi_d^{-2}(1-\upsilon)}$ with $\upsilon = 10^{-8}$, and the Hammersley sequence is employed. Next, the corresponding output values $\mathscr{Y} = [y^{(1)}, y^{(2)}, \cdots, y^{(n_0)}]^{\top}$ of the $\mathcal{G}$-function at $\mathscr{U}$ can be obtained through parallel computation. Finally, the initial observation dataset is formed by $\mathscr{D} = \{\mathscr{U}, \mathscr{Y}\}$. Let $n = n_0$.

Step B2: Obtain the GP posterior of the $\mathcal{G}$-function

This step entails obtaining the posterior distribution of the $\mathcal{G}$-function (i.e., $\mathcal{GP}(m_{\mathcal{G}_n}(\boldsymbol{u}), k_{\mathcal{G}_n}(\boldsymbol{u}, \boldsymbol{u}'))$), given the data $\mathscr{D}$. Such task can be accomplished using some well-established GP regression toolkits. In this study, we utilize the *fitrgp* function from the Statistics and Machine Learning Toolbox in Matlab.

Step B3: Compute the two integrals in the stopping criterion

In this stage, one needs to compute the two estimates $\hat{m}_{\mathcal{P}_{f,n}}$ and $\hat{\mathcal{L}}_n$ by using the proposed HSS method outlined in Section 3.1.

Step B4: Check the stopping criterion

If $\frac{\hat{\mathcal{L}}_n}{\hat{m}_{\mathcal{P}_{f,n}}} < \epsilon$ is met twice in a row, go to **Step B6**; Otherwise, go to **Step B5**.

Step B5: Enrich the observation dataset

In this phase, the current observation dataset needs to be enriched with new data. First, identify the next n_a point(s) $\boldsymbol{\mathcal{U}}^+ = \left\{\boldsymbol{u}^{(n+j)}\right\}_{j=1}^{n_a}$ by optimizing the proposed learning function(s), where the genetic algorithm is used in this study. Then, the evaluation of the $\mathcal{G}$-function at $\boldsymbol{\mathcal{U}}^+$ can be performed in parallel, which yields the output value(s) $\boldsymbol{\mathcal{Y}}^+$. Finally, the enriched observation dateset can be formulated as $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}} \cup \{\boldsymbol{\mathcal{U}}^+, \boldsymbol{\mathcal{Y}}^+\}$. Let $n = n + n_a$ and proceed to **Step B2**.

Step B6: End the method

Return the current estimate $\hat{m}_{\mathcal{P}_{f,n}}$ as the failure probability estimate and end the entire procedure.

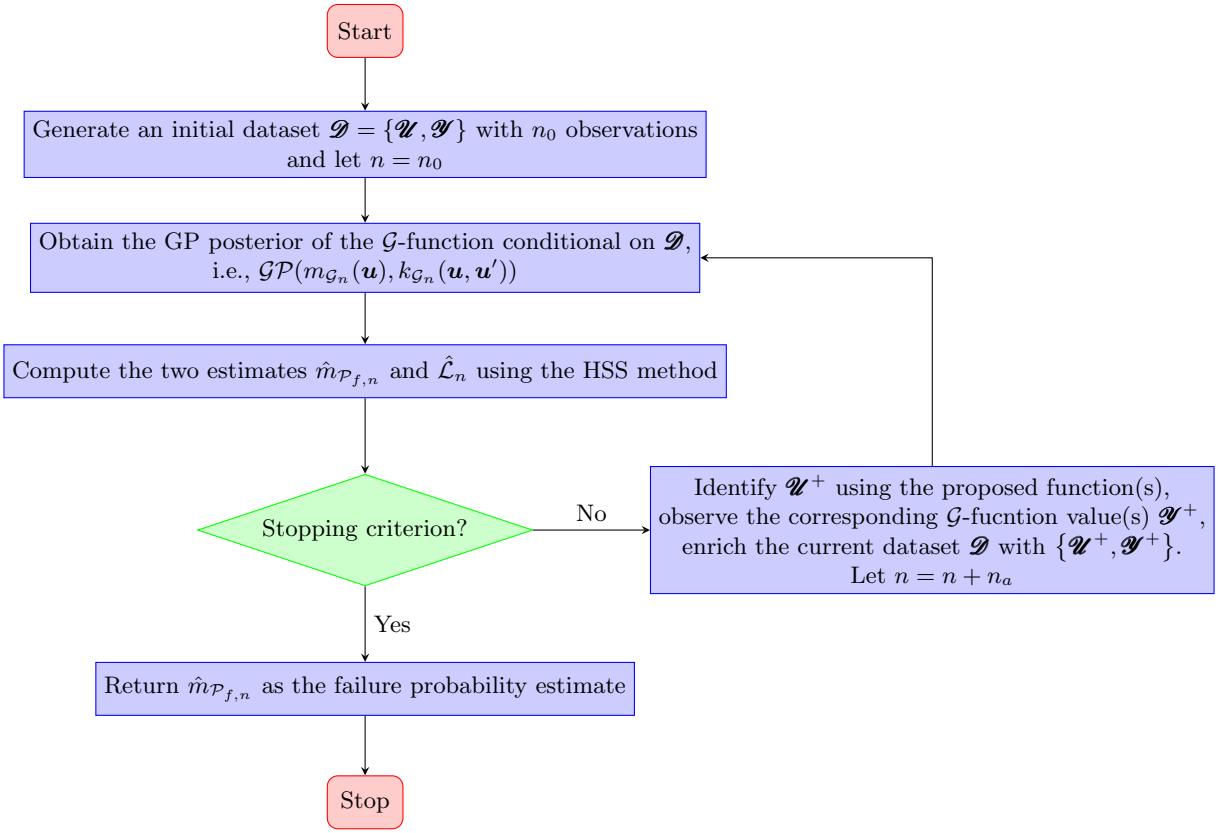


Figure 2: Flowchart of the proposed SBALQ method.

One must specify at least the parameters n_0, h, ΔN, γ_1, γ_2 and ϵ before running the SBALQ algorithm. Although they all have a clear physical meaning, some experience may be required to set the proper values. For example, the size of the initial observed dataset n_0 should be neither too small nor too large. On the one hand, too small n_0 can result in a very coarse initial GP posterior model, which is less informative for subsequent active learning. On the other hand, too large n_a is neither necessary nor unadvisable.

4. Numerical examples

Five numerical examples are analyzed in this section to demonstrate the effectiveness of the SBALQ method in estimating extremely small failure probabilities. In each of these examples, the parameters not yet specified in the proposed method are set to be: $n_0 = 10$, $h = 10$, $\Delta N = 2 \times 10^5$, $\gamma_1 = \gamma_2 = 2\%$, $\epsilon = 2\%(4\%)$. Additionally, we vary the value of n_a to systematically investigate its influence on the obtained results. For comparison, several representative existing methods, i.e., PBALC1 [26], PBALC2 [26], PBALC3 [26], PABQ [20] and extreme AK-MCS (eAK-MCS) [32] are also conducted. These methods and the proposed method are run independently 20 times to evaluate their robustness. Where it is feasible to do so, the reference failure probabilities are generated using the MCS method with a significantly large number of simulations.

4.1. Example 1: A series system with four branches

The first example involves a series system comprising four branches, which has been extensively studied as a benchmark (see for example [15, 29, 33]). The performance function is given as follows:

$$g\left(X_1, X_2\right) = \min \begin{cases} a + \frac{(X_1 - X_2)^2}{10} - \frac{(X_1 + X_2)}{\sqrt{2}} \\ a + \frac{(X_1 - X_2)^2}{10} + \frac{(X + X_2)}{\sqrt{2}} \\ (X_1 - X_2) + \frac{b}{\sqrt{2}} \\ (X_2 - X_1) + \frac{b}{\sqrt{2}} \end{cases}, \tag{33}$$

where X_1 and X_2 are two independent standard normal variables; a and b are two constant parameters, which are specified as 6 and 12, respectively.

Table 2 summarizes the results of several structural reliability analysis methods. The reference failure probability, obtained thorough the MCS method with 10^{13} samples, is 3.04×10^{-9} with a COV of 0.57%. When $n_a = 1$ (indicating that parallel computing is not permitted during the active learning phase), the proposed SBALQ method requires an average number of 32.15 iterations, which stands as the lowest among PBACLC1, PBACLC2 and PBACLC3, while producing a fairly good average failure probability with a small COV. When considering the case $n_a = 4$, the PABQ method demands more iterations on average compared to the proposed method. The results of eAK-MCS are missing because it is unable to converge in this example. The proposed method ($n_a = 4$) can give a fairly good failure probability mean with a small COV at the expense of only 12.75 iterations (on average). As for the proposed method itself, it can always produce nearly unbiased results with COVs less than 4% when n_a varies from 1 to 8. Besides, it is evident that the average number of $\mathcal{G}$-function calls increases with n_a, while the average number of iterations first decreases, stays the same, and then finally increases. This observation implies that selecting an excessive number of points at each iteration may not necessarily lead to a reduction in the overall number of iterations.

For a more intuitive illustration, Fig. 3 shows the selected points and convergence curve obtained from a typical run of the proposed method ($\epsilon = 2\%$ and $n_a = 2$). As evident in Figure 3(a), a majority of the points chosen during the active learning phase are clustered around the four critical regions of the actual limit state curve. As we continue to identify more informative points, Figure 3(b) illustrates how the posterior failure probability estimate gradually approaches the reference value before the stopping criterion is reached. These observations indicate the effectiveness of the proposed learning functions and stopping criterion.

4.2. Example 2: A nonlinear oscillator

As a second example, we consider a nonlinear oscillator subject to a rectangular pulse load [34], as shown in Fig. 4. The performance function reads:

$$g\left(m, k_1, k_2, r, F_1, t_1\right) = 3r - \left| \frac{2F_1}{k_1 + k_2} \sin\left(\frac{t_1}{2} \sqrt{\frac{k_1 + k_2}{m}} \right) \right|, \tag{34}$$

where m, k_1, k_2, r, F_1 and t_1 are six random variables, as listed in Table 3.

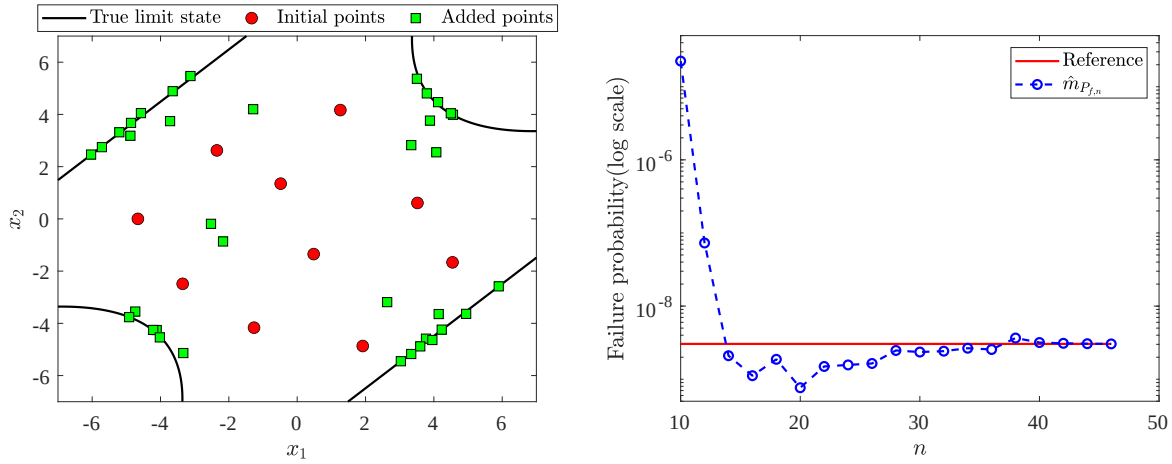
The results of several structural reliability analysis methods are reported in Table 4. The reference value

Table 2: Reliability analysis results of Example 1 by several methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS		-	10^{13}	3.04×10^{-9}	0.57%
PBALC1 ($\epsilon_1 = 2.5\%$) [26]	$n_a = 1$	35.75	44.75	3.04×10^{-9}	3.82%
PBALC2 ($\epsilon_2 = 2.5\%$) [26]	$n_a = 1$	41.10	50.10	3.04×10^{-9}	1.39%
PBALC3 ($\epsilon_3 = 5.0\%$) [26]	$n_a = 1$	40.50	49.50	3.03×10^{-9}	1.99%
PABQ	$n_a = 4$	18.00	78.00	2.95×10^{-9}	1.06%
eAK-MCS	$n_a = 4$	-	-	-	-
Proposed SBALQ ($\epsilon = 2\%$)	$n_a = 1$	32.15	41.15	3.03×10^{-9}	1.31%
	$n_a = 2$	19.35	46.70	3.03×10^{-9}	0.58%
	$n_a = 3$	15.55	53.65	3.01×10^{-9}	0.75%
	$n_a = 4$	12.75	57.00	3.02×10^{-9}	0.80%
	$n_a = 5$	11.10	60.50	3.02×10^{-9}	0.68%
	$n_a = 6$	9.95	63.70	3.03×10^{-9}	0.48%
	$n_a = 7$	9.95	72.65	3.03×10^{-9}	0.44%
	$n_a = 8$	10.65	87.20	3.00×10^{-9}	3.98%

Note: N_{iter} = the total number of iterations; N_{call} = the total number of performance function calls; $\hat{P}_f$ = the failure probability estimate; $\delta_{\hat{P}_f}$ = the COV of $\hat{P}_f$.

of the failure probability is taken as 4.04×10^{-8} (with a COV of 0.50%), which is produced by the MCS method with 10^{12} samples. In the non-parallel case (i.e., when $n_a = 1$), the proposed SBALQ method performs similarly to PBALC1, PBALC2, and PBALC3, with a slightly smaller COV. In the parallel case (i.e., when $n_a = 4$), the proposed method requires fewer iterations on average compared to the PABQ and eAK-MCS methods. However, the proposed method exhibits a smaller COV of 1.40% compared to the PABQ and eAK-MCS methods, which have COVs of 7.48% and 3.92%, respectively. In all eight cases, the



(a) Selected points (b) Convergence curve

Figure 3: Illustration of the proposed SBALQ method ($\epsilon = 2\%$ and $n_a = 2$) for Example 1.

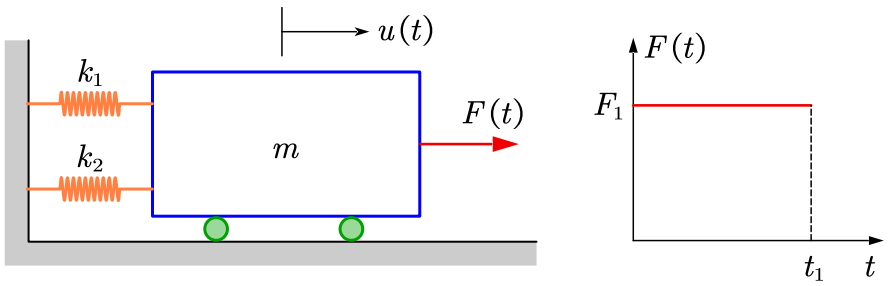


Figure 4: A nonlinear oscillator subject to a rectangular pulse load.

proposed method can give a nearly unbiased average failure probability with a small COV. Notably, the average number of iterations can be reduced from 22.45 to 6.20 as n_a increases from 1 to 8.

4.3. Example 3: A reinforced concrete section

The third example consists of a reinforced concrete section under a bending moment [35], as depicted in Fig. 5. The performance function can be expressed as:

$$g(\boldsymbol{X}) = X_1 X_2 X_3 - \frac{X_1^2 X_2^2 X_4}{X_5 X_6} - X_7, \tag{35}$$

where X_1 to X_7 are seven random variables, as shown in Table 5.

Table 6 lists the results of several different methods. The reference failure probability generated by the MCS method (with 5×10^{11} samples) is 1.59×10^{-8} (with a COV of 0.79%). In the non-parallel case (i.e.,

Table 3: Random variables for Example 2.

Variable	Description	Distribution	Mean	COV
m	Mass	Lognormal	1.0	0.05
k_1	Stiffness	Lognormal	1.0	0.10
k_2	Stiffness	Lognormal	0.2	0.10
r	Yield displacement	Lognormal	0.5	0.10
F_1	Load amplitude	Lognormal	0.4	0.20
t_1	Load duration	Lognormal	1.0	0.20

Table 4: Reliability analysis results of Example 2 by several methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS	-	-	10^{12}	4.04×10^{-8}	0.50%
PBALC1 ($\epsilon_1 = 5\%$) [26]	$n_a = 1$	20.10	29.10	4.03×10^{-8}	4.29%
PBALC2 ($\epsilon_2 = 5\%$) [26]	$n_a = 1$	22.90	31.90	4.07×10^{-8}	2.61%
PBALC3 ($\epsilon_3 = 10\%$) [26]	$n_a = 1$	21.95	30.95	4.05×10^{-8}	3.66%
PABQ	$n_a = 4$	9.40	43.60	3.94×10^{-8}	7.48%
eAK-MCS	$n_a = 4$	11.35	51.40	4.05×10^{-8}	3.92%
Proposed SBALQ ($\epsilon = 4\%$)	$n_a = 1$	22.45	31.45	4.03×10^{-8}	1.72%
	$n_a = 2$	13.35	34.70	4.09×10^{-8}	2.08%
	$n_a = 3$	10.00	37.00	4.03×10^{-8}	1.95%
	$n_a = 4$	8.50	40.00	4.05×10^{-8}	1.40%
	$n_a = 5$	7.95	44.75	4.04×10^{-8}	1.12%
	$n_a = 6$	7.20	47.20	4.02×10^{-8}	1.95%
	$n_a = 7$	6.45	48.15	4.06×10^{-8}	1.74%
	$n_a = 8$	6.20	51.60	4.04×10^{-8}	1.48%

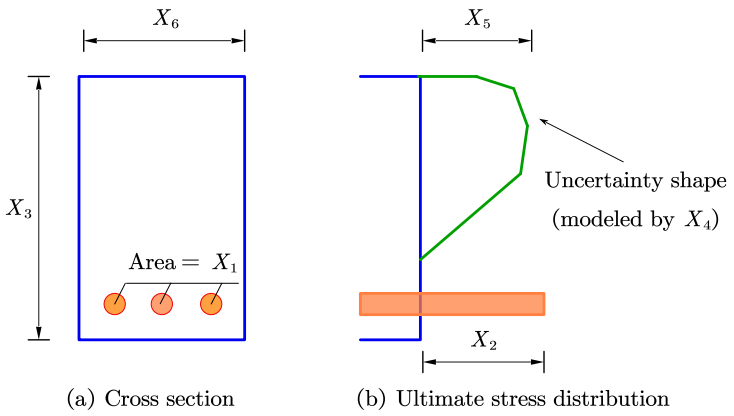


Figure 5: Ultimate stress state of a reinforced concrete section subject to a bending moment.

Table 5: Basic random variables for Example 3.

Variable	Description	Distribution	Mean	COV
X_1	Area of reinforcement	Normal	1260 mm^2	0.05
X_2	Yield stress of reinforcement	Lognormal	300 N/mm^2	0.10
X_3	Effective depth of reinforcement	Normal	770 mm	0.05
X_4	Stress–strain factor of concrete	Lognormal	0.35	0.10
X_5	Compressive strength of concrete	Lognormal	30 N/mm^2	0.15
X_6	Width of section	Normal	400 mm	0.05
X_7	Applied bending moment	Lognormal	80 kN·m	0.20

$n_a = 1$), the proposed method performs slightly better than PBALC1, PBALC2 and PBALC3 . When $N_a = 4$, the PABQ and eAK-MCS methods require fewer iterations than the proposed method. However, they exhibit COVs of 12.90% and 7.17%, respectively, which are much larger than that of the proposed method (1.50%). As n_a increases from 1 to 8, the proposed method can produce fairly good results with almost no bias and a small COV, while significantly reducing the average number of iterations.

Table 6: Reliability analysis results of Example 3 by several methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
MCS		-	10^{12}	1.59×10^{-8}	0.79%
PBALC1 ($\epsilon_1 = 5\%$)	$n_a = 1$	14.70	23.70	1.60×10^{-8}	3.16%
PBALC2 ($\epsilon_2 = 5\%$)	$n_a = 1$	16.45	25.45	1.57×10^{-8}	2.63%
PBALC3 ($\epsilon_3 = 10\%$)	$n_a = 1$	16.10	25.10	1.59×10^{-8}	3.37%
PABQ	$n_a = 4$	6.75	33.00	1.58×10^{-8}	12.90%
eAK-MCS	$n_a = 4$	5.40	27.60	1.60×10^{-8}	7.17%
Proposed SBALQ ($\epsilon = 4\%$)	$n_a = 1$	14.60	23.60	1.58×10^{-8}	2.35%
	$n_a = 2$	9.95	27.90	1.58×10^{-8}	2.51%
	$n_a = 3$	8.90	33.70	1.56×10^{-8}	1.59%
	$n_a = 4$	8.15	38.60	1.57×10^{-8}	1.50%
	$n_a = 5$	7.25	41.25	1.57×10^{-8}	1.53%
	$n_a = 6$	6.60	43.60	1.56×10^{-8}	1.26%
	$n_a = 7$	6.20	46.40	1.57×10^{-8}	1.64%
	$n_a = 8$	6.20	51.60	1.57×10^{-8}	1.05%

4.4. Example 4: A spatial truss structure

The fourth example is a 120-bar spatial truss structure under vertical loads [19, 20], as depicted in Fig. 6. The finite element of the structure is built using OpenSees and it contains 49 nodes and 120 truss members. The cross-sectional area and Young's modulus, denoted A and E, are assumed to be the same for all members. Thirteen vertical loads, denoted P_0 - P_{12}, are applied to nodes 0 - 12, respectively. The performance function is formulated as follows:

$$g(\boldsymbol{X}) = \Delta - V_0(A, E, P_0\text{-}P_{12}), \tag{36}$$

where V_0 denotes the vertical displacement of node 0; Δ is the associated threshold, which is specified as 100 mm; A, E, P_0 - P_{12} are treated as 15 random variables, as listed in Table 7.

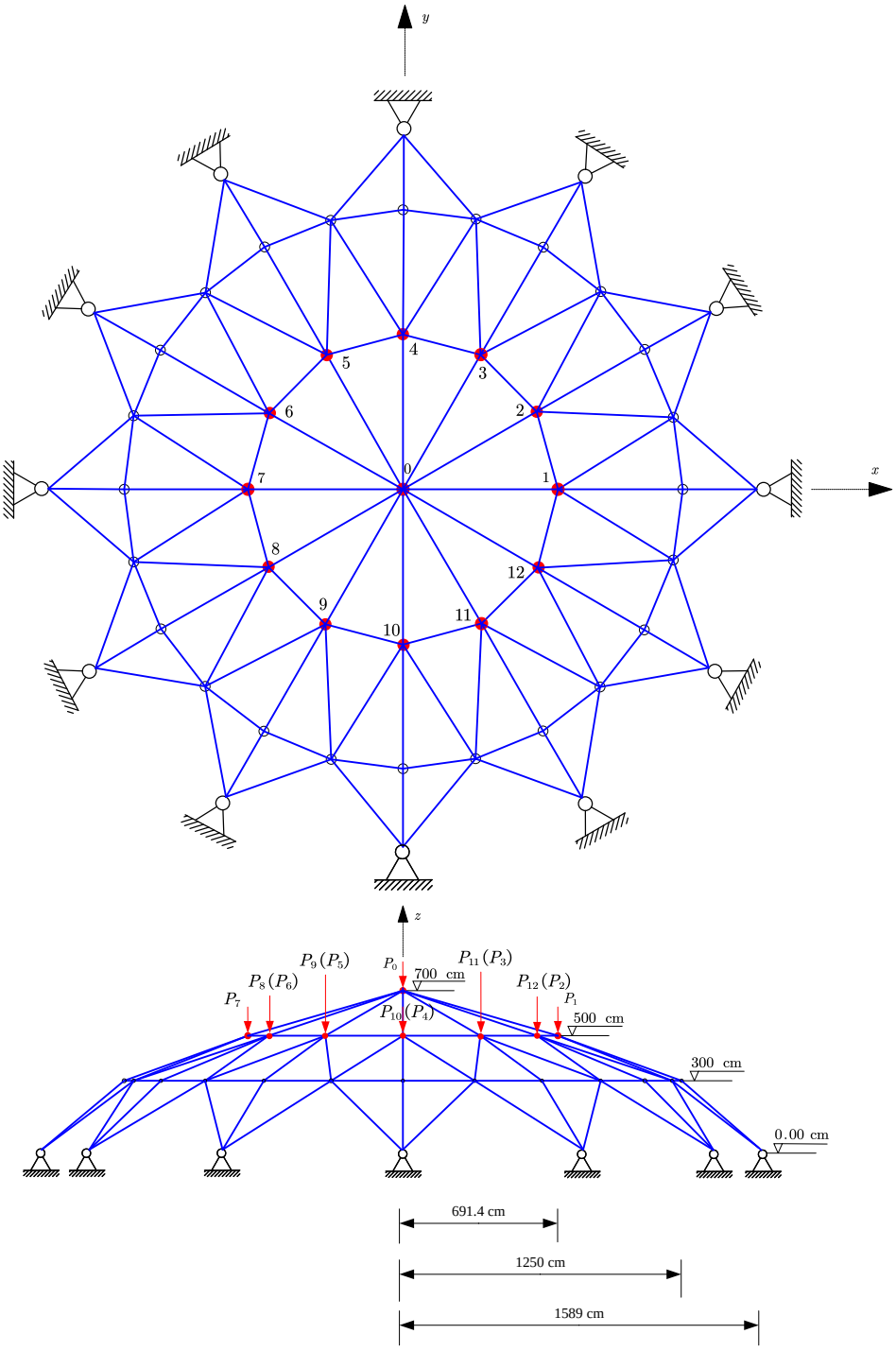


Figure 6: A 120-bar space truss structure subject to thirteen vertical loads.

Table 7: Random variables for Example 4.

Variable	Distribution	Mean	COV
A	Normal	2,000 mm^2	0.10
E	Normal	200 GPa	0.10
P_0	Lognormal	500 kN	0.20
$P_1 \sim P_{12}$	Lognormal	60 kN	0.15

The reliability analysis results of several methods are summarized in Table 8. To provide a reference solution, the important sampling (IS) method available in UQLab [36] is used instead of MCS. The reference failure probability obtained is 1.87×10^{-7} with a COV of 1.98%, at the expense of 30,142 $\mathcal{G}$-function evaluations. When $n_a = 1$, the three PBALC methods, on average, demand slightly fewer $\mathcal{G}$ function calls compared to the proposed SBALQ method, but they exhibit higher variability. When $n_a = 4$, although the PABQ method outperforms the proposed SBALQ method in terms of the average number of iterations, the former yields a rather large COV of 162.04%. The eAK-MCS method fails to produce results due to computer memory issues. As for the proposed SBALQ method itself, the average number of iterations can be reduced from 37.05 to 12.55 (although the average total number of performance function calls increases) when n_a is increased from 1 to 8. In addition, the proposed method can produce a nearly unbiased failure probability mean with a COV less than 8% in each case.

4.5. Example 5: A jet engine turbine blade

The final example concerns a turbine blade from a jet engine (as depicted in Fig. 7), which is available in the Partial Differential Equation Toolbox of Matlab R2022b. The turbine blade is made of nickel-based alloy (NIMONIC 90) with Young's modulus E, Poisson's ratio ν and the coefficient of thermal expansion CTE. The root face (face 3 in Fig. 7(a)) in contact with other metal is fixed. Pressure load p_1 is applied to the pressure sides, and pressure loads p_2 are applied to the suction sides. The finite-element model is discretized by linear tetrahedral elements with the maximum element size 0.01 m, which is shown in Fig. 7(b). One typical cause of the blade failure is mechanical stress, and hence we define the following

Table 8: Reliability analysis results of Example 4 by several methods.

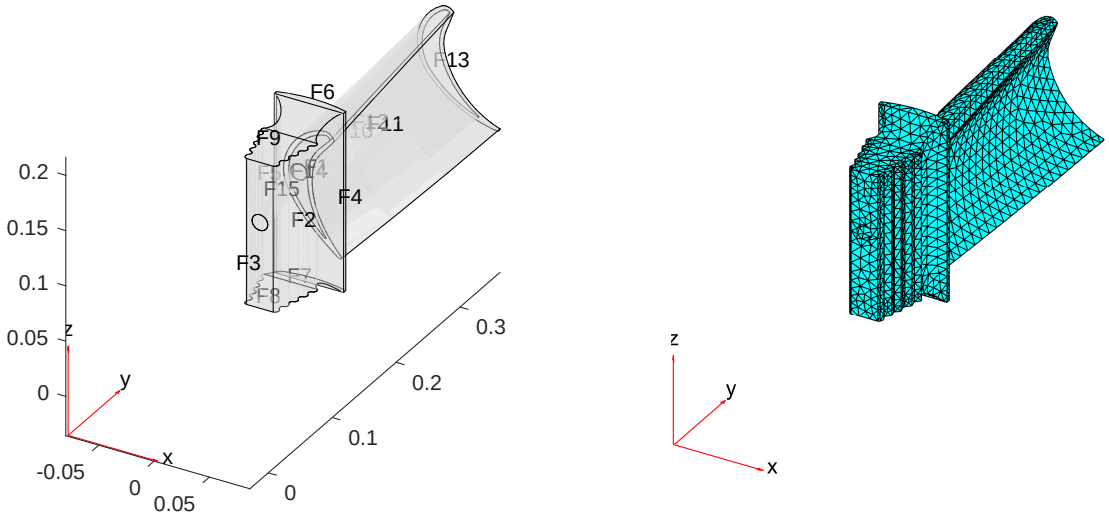
Method		N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
IS		-	$30,142$	1.87×10^{-7}	1.98%
PBALC1 ($\epsilon_1=5\%$)	$n_a=1$	36.25	45.25	1.85×10^{-7}	7.39%
PBALC2 ($\epsilon_2=5\%$)	$n_a=1$	30.45	39.45	1.84×10^{-7}	6.36%
PBALC3 ($\epsilon_3=10\%$)	$n_a=1$	32.75	41.75	1.84×10^{-7}	7.44%
PABQ	$n_a=4$	9.70	44.80	1.95×10^{-7}	162.04%
eAK-MCS	$n_a=4$	-	-	-	-
Proposed SBALQ ($\epsilon=4\%$)	$n_a=1$	37.05	46.05	1.82×10^{-7}	3.39%
	$n_a=2$	22.70	53.40	1.80×10^{-7}	6.87%
	$n_a=3$	20.20	67.60	1.82×10^{-7}	6.48%
	$n_a=4$	17.20	74.80	1.85×10^{-7}	5.09%
	$n_a=5$	14.95	79.75	1.82×10^{-7}	7.93%
	$n_a=6$	14.30	89.80	1.81×10^{-7}	6.54%
	$n_a=7$	13.15	95.05	1.83×10^{-7}	6.64%
	$n_a=8$	12.55	102.40	1.82×10^{-7}	6.42%

performance function:

$$g(\boldsymbol{X}) = \sigma_{th} - \sigma_{max}(E, \nu, CET, p_1, p_2), \tag{37}$$

where $\sigma_{th}=0.8$ GPa is the threshold for the maximum von Mises stress σ_{max} of the blade; E, ν, CET, p_1 and p_2 are five random variables, as listed in Table 9.

Table 10 reports the reliability analysis results of several methods, i.e., IS [36], PBALC1, PBALC2, PBALC3, PABQ, eAK-MCS and the proposed SBALQ method. IS [36] was implemented to provide a reference value for the failure probability. However, its results were not available because something went wrong during the analysis. As an alternative, the reference failure probability is taken as 1.25×10^{-8} (with a COV of 0.77%), which is the mean value given by PBALC1 ($\epsilon_1=5\%$) with 20 runs. For $n_a=1$, PBALC1,



(a) Face labels

(b) Finite-element mesh

Figure 7: A jet engine turbine blade.

Table 9: Random variables for Example 5.

Variable	Distribution	Mean	COV
E	Normal	220 GPa	0.10
ν	Normal	0.30	0.05
CET	Uniform	1.25×10^{-7} 1/K	0.05
p_1	Gumbel	500 kPa	0.15
p_2	Gumbel	450 kPa	0.15

PBALC2 and PBALC3 methods and the proposed SBALQ method can produce quite similar mean values for the failure probability with rather small COVs. Among them, the proposed method requires the fewest $\mathcal{G}$-function calls. When $n_a = 4$, PABQ gives a biased mean for the failure probability, while processing a large COV. Like IS, eAK-MCS encountered an error when running the finite element analysis, so no results can be given. On the contrary, the proposed method ($n_a = 4$) performs well, as do other cases (i.e., $n_a = 1, 2, 3, 5, 6, 7, 8$). Moreover, the number of iterations required by the proposed method decreases as n_a increases from 1 to 5, but increases as n_a increases from 5 to 8.

Table 10: Reliability analysis results of Example 5 by several methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	$\delta_{\hat{P}_f}$
IS		-	-	-	-
PBALC1 ($\epsilon_1 = 5\%$)	$n_a = 1$	28.85	37.85	1.25×10^{-8}	0.77%
PBALC2 ($\epsilon_2 = 5\%$)	$n_a = 1$	41.00	50.00	1.25×10^{-8}	1.45%
PBALC3 ($\epsilon_3 = 10\%$)	$n_a = 1$	38.30	47.30	1.24×10^{-8}	2.04%
PABQ	$n_a = 4$	3.95	21.80	1.01×10^{-8}	34.05%
eAK-MCS	$n_a = 4$	-	-	-	-
Proposed SBALQ ($\epsilon = 4\%$)	$n_a = 1$	18.65	27.65	1.25×10^{-8}	1.63%
	$n_a = 2$	12.00	32.00	1.26×10^{-8}	2.92%
	$n_a = 3$	10.50	38.50	1.25×10^{-8}	1.42%
	$n_a = 4$	8.35	39.40	1.26×10^{-8}	1.10%
	$n_a = 5$	8.05	45.25	1.25×10^{-8}	1.02%
	$n_a = 6$	8.55	55.30	1.25×10^{-8}	0.84%
	$n_a = 7$	8.15	60.05	1.25×10^{-8}	0.59%
	$n_a = 8$	8.95	73.60	1.25×10^{-8}	1.28%

Remark. As observed in the five numerical examples above, the average number of iterations required by the proposed method does not always decrease as n_a increases. This means that if n_a cores are used for an expensive $\mathcal{G}$ function, a too large n_a may not lead to a reduction in the overall computation time. According to our computational experience, $n_a = 4$ - 6 should be sufficient.

5. Concluding remarks

This study presents an innovative method termed 'semi-Bayesian active learning quadrature' (SBALQ) for structural reliability analysis, particularly for evaluating extremely small failure probabilities. The main contributions lie in the development of two key components for active learning (i.e., stopping criterion and

learning function) based on the well-established Bayesian failure probability inference framework, while avoiding the use of the posterior variance of the failure probability, which is expensive to evaluate. First, we introduce a new stopping criterion by exploring the structure of the posterior mean of the failure probability only. This criterion involves two analytically intractable integrals. Second, a numerical integration technique called 'hyper-shell simulation' is devised to approximate the integrals. Third, we propose a new learning function based on the proposed stopping criterion, and by maximizing it a single point can be identified at each iteration of the active learning phase. Fourth, the proposed learning function is further modified by multiplying an influence function so as to enable multi-point selection and hence parallel distributed processing. It is empirically shown from five numerical examples that the proposed SBALQ method is capable of estimating very low failure probabilities in the order of 10^{-7} - 10^{-9}, while maintaining desired efficiency and accuracy. It is worth noting that the computational efficiency can be further improved by leveraging the parallelizability inherent in the proposed approach.

The proposed method, in its current form, performs poorly in high dimensions due to the limitations of GP and HSS. Consequently, a promising avenue for future research lies in the integration of effective dimension reduction techniques. Furthermore, it is still challenging to apply the proposed method to highly nonlinear problems when using the squared exponential kernel, as it implies a smooth assumption.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Chao Dang is mainly supported by China Scholarship Council (CSC). Michael Beer would like to thank the support of the National Natural Science Foundation of China under grant number 72271025.

Data availability

Data will be made available on request.

Appendix A. Generation of uniform random samples in the $h-1$ inner hyper-shells

The procedure for generating N_i uniform random samples in the $h-1$ inner hyper-shells is as follows:

1. Draw N_i random samples that are uniformly distributed on $\left[R_{i-1}^d, R_i^d\right]$, denoted as $\left\{v^{(j)} : j = 1, 2, \cdots, N_i\right\}$;
2. Generate N_i random samples from $\phi_{\boldsymbol{U}}(\boldsymbol{u})$, denoted as $\left\{\boldsymbol{u}^{(j)} : j = 1, 2, \cdots, N_i\right\}$;
3. Obtain the j-th sample in the i-th inner hyper-shell by $\boldsymbol{u}^{(i,j)} = \frac{\sqrt[d]{v^{(j)}}\boldsymbol{u}^{(j)}}{||\boldsymbol{u}^{(j)}||}$.

Appendix B. Generation of random samples in the outermost hyper-shell

The procedure for generating N_i random samples from $\psi^{(h)}(\boldsymbol{u})$ in the outermost hyper-shell is as follows:

1. Draw N_i random samples that are uniformly distributed on $\left[1 - 10^{-(h-1)}, 1\right]$, which are denoted as $\left\{p^j : j = 1, 2, \cdots, N_i\right\}$;
2. Generate N_i random samples from $\phi_{\boldsymbol{U}}(\boldsymbol{u})$, denoted as $\left\{\boldsymbol{u}^{(j)} : j = 1, 2, \cdots, N_i\right\}$;
3. Obtain the j-th sample in the outermost hyper-shell by $\boldsymbol{u}^{(i,j)} = \frac{\sqrt{\chi_d^{-2}(p^j)}\boldsymbol{u}^{(j)}}{||\boldsymbol{u}^{(j)}||}$.

References

[1] G. Rubino, B. Tuffin, et al., Rare event simulation using Monte Carlo methods, Vol. 73, Wiley Online Library, 2009.

[2] S.-K. Au, J. L. Beck, A new adaptive importance sampling scheme for reliability calculations, Structural Safety 21 (2) (1999) 135–158. doi:https://doi.org/10.1016/S0167-4730(99)00014-4.

[3] I. Papaioannou, S. Geyer, D. Straub, Improved cross entropy-based importance sampling with a flexible mixture model, Reliability Engineering & System Safety 191 (2019) 106564. doi:https://doi.org/10.1016/j.ress.2019.106564.

[4] J. Xian, Z. Wang, Relaxation-based importance sampling for structural reliability analysis, Structural Safety 106 (2024) 102393. doi:https://doi.org/10.1016/j.strusafe.2023.102393.

[5] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277. doi:https://doi.org/10.1016/S0266-8920(01)00019-4.

[6] J. Nie, B. R. Ellingwood, Directional methods for structural reliability analysis, Structural Safety 22 (3) (2000) 233–249. doi:https://doi.org/10.1016/S0167-4730(00)00014-X.

[7] P.-S. Koutsourelakis, H. J. Pradlwarter, G. I. Schuëller, Reliability of structures in high dimensions, part I: algorithms and applications, Probabilistic Engineering Mechanics 19 (4) (2004) 409–417. doi:https://doi.org/10.1016/j.probengmech.2004.05.001.

[8] A. M. Hasofer, N. C. Lind, Exact and invariant second-moment code format, Journal of the Engineering Mechanics Division 100 (1) (1974) 111–121. doi:https://doi.org/10.1061/JMCEA3.0001848.

[9] K. W. Breitung, Asymptotic approximations for probability integrals, Springer, 2006.

[10] L. Tvedt, Distribution of quadratic forms in normal space—application to structural reliability, Journal of Engineering Mechanics 116 (6) (1990) 1183–1197. doi:https://doi.org/10.1061/(ASCE)0733-9399(1990)116:6(1183).

[11] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2) (2010) 183–197. doi:https://doi.org/10.1016/j.probengmech.2009.10.003.

[12] I. Kaymaz, Application of kriging method to structural reliability problems, Structural Safety 27 (2) (2005) 133–151. doi:https://doi.org/10.1016/j.strusafe.2004.09.001.

[13] C. Song, R. Kawai, Monte Carlo and variance reduction methods for structural reliability analysis: A comprehensive review, Probabilistic Engineering Mechanics (2023) 103479doi:https://doi.org/10.1016/j.probengmech.2023.103479.

[14] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, J. M. McFarland, Efficient global reliability analysis for nonlinear implicit performance functions, AIAA Journal 46 (10) (2008) 2459–2468. doi:https://doi.org/10.2514/1.34321.

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

[16] R. Teixeira, M. Nogal, A. O'Connor, Adaptive approaches in metamodel-based reliability analysis: A review, Structural Safety 89 (2021) 102019. doi:https://doi.org/10.1016/j.strusafe.2020.102019.

[17] S. S. Afshari, F. Enayatollahi, X. Xu, X. Liang, Machine learning-based methods in structural reliability analysis: A review, Reliability Engineering & System Safety 219 (2022) 108223. doi:https://doi.org/10.1016/j.ress.2021.108223.

[18] M. Moustapha, S. Marelli, B. Sudret, Active learning for structural reliability: Survey, general framework and benchmark, Structural Safety 96 (2022) 102174. doi:https://doi.org/10.1016/j.strusafe.2021.102174.

[19] C. Dang, P. Wei, J. Song, M. Beer, Estimation of failure probability function under imprecise probabilities by active learning–augmented probabilistic integration, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 7 (4) (2021) 04021054. doi:https://doi.org/10.1061/AJRUA6.0001179.

[20] C. Dang, P. Wei, M. G. Faes, M. A. Valdebenito, M. Beer, Parallel adaptive Bayesian quadrature for rare event estimation, Reliability Engineering & System Safety 225 (2022) 108621. doi:https://doi.org/10.1016/j.ress.2022.108621.

[21] C. Dang, M. A. Valdebenito, M. G. Faes, P. Wei, M. Beer, Structural reliability analysis: A Bayesian perspective, Structural Safety 99 (2022) 102259. doi:https://doi.org/10.1016/j.strusafe.2022.102259.

[22] C. Dang, M. A. Valdebenito, J. Song, P. Wei, M. Beer, Estimation of small failure probabilities by partially Bayesian

active learning line sampling: Theory and algorithm, Computer Methods in Applied Mechanics and Engineering 412 (2023) 116068. doi:https://doi.org/10.1016/j.cma.2023.116068.

[23] C. Dang, M. A. Valdebenito, M. G. Faes, J. Song, P. Wei, M. Beer, Structural reliability analysis by line sampling: A Bayesian active learning treatment, Structural Safety 104 (2023) 102351. doi:https://doi.org/10.1016/j.strusafe.2023.102351.

[24] C. Dang, M. Beer, Bayesian active learning line sampling with log-normal process for rare-event probability estimation, Reliability Engineering & System Safety (2024) In Press.

[25] Z. Hu, C. Dang, L. Wang, M. Beer, Parallel Bayesian probabilistic integration for structural reliability analysis with small failure probabilities, Structural Safety 106 (2024) 102409. doi:https://doi.org/10.1016/j.strusafe.2023.102409.

[26] C. Dang, M. G. Faes, M. A. Valdebenito, P. Wei, M. Beer, Partially Bayesian active learning cubature for structural reliability analysis with extremely small failure probabilities, Computer Methods in Applied Mechanics and Engineering 422 (2024) 116828. doi:https://doi.org/10.1016/j.cma.2024.116828.

[27] Z. Chen, G. Li, J. He, Z. Yang, J. Wang, A new parallel adaptive structural reliability analysis method based on importance sampling and K-medoids clustering, Reliability Engineering & System Safety 218 (2022) 108124. doi:https://doi.org/10.1016/j.ress.2021.108124.

[28] G. Li, T. Wang, Z. Chen, J. He, X. Wang, X. Du, RBIK-SS: A parallel adaptive structural reliability analysis method for rare failure events, Reliability Engineering & System Safety 239 (2023) 109513. doi:https://doi.org/10.1016/j.ress.2023.109513.

[29] M. Su, G. Xue, D. Wang, Y. Zhang, Y. Zhu, A novel active learning reliability method combining adaptive Kriging and spherical decomposition-MCS (AK-SDMCS) for small failure probabilities, Structural and Multidisciplinary Optimization 62 (2020) 3165–3187. doi:https://doi.org/10.1007/s00158-020-02661-w.

[30] A. Persoons, P. Wei, M. Broggi, M. Beer, A new reliability method combining adaptive Kriging and active variance reduction using multiple importance sampling, Structural and Multidisciplinary Optimization 66 (6) (2023) 144. doi:https://doi.org/10.1007/s00158-023-03598-6.

[31] D. Zhan, J. Qian, Y. Cheng, Pseudo expected improvement criterion for parallel EGO algorithm, Journal of Global Optimization 68 (2017) 641–662. doi:https://doi.org/10.1007/s10898-016-0484-7.

[32] N. Razaaly, P. M. Congedo, Extension of AK-MCS for the efficient computation of very small failure probabilities, Reliability Engineering & System Safety 203 (2020) 107084. doi:https://doi.org/10.1016/j.ress.2020.107084.

[33] J. Bect, L. Li, E. Vazquez, Bayesian subset simulation, SIAM/ASA Journal on Uncertainty Quantification 5 (1) (2017) 762–786. doi:https://doi.org/10.1137/16M1078276.

[34] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1) (1990) 57–66. doi:https://doi.org/10.1016/0167-4730(90)90012-E.

[35] J. Zhou, A. S. Nowak, Integration formulas to evaluate functions of random variables, Structural Safety 5 (4) (1988)

267–284. doi:https://doi.org/10.1016/0167-4730(88)90028-8.

[36] S. Marelli, R. Schöbi, B. Sudret, UQLab user manual – Structural reliability (Rare event estimation), Tech. rep., Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, report UQLab-V2.0-107 (2022).