Parallel adaptive Bayesian quadrature for rare event estimation

Chao Dang[a,*], Pengfei Wei[b], Matthias G.R. Faes[c], Marcos A. Valdebenito[d], Michael Beer[a,e,f]

[a]*Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany*
[b]*School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, PR China*
[c]*Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler-Str. 5, Dortmund 44227, Germany*
[d]*Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Av. Padre Hurtado 750, 2562340 Viña del Mar, Chile*
[e]*Institute for Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZF, United Kingdom*
[f]*International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, PR China*

Abstract

Various numerical methods have been extensively studied and used for reliability analysis over the past several decades. However, how to understand the effect of numerical uncertainty (i.e., numerical errors due to the discretization of the performance function) on the failure probability is still a challenging issue. The active learning probabilistic integration (ALPI) method offers a principled approach to quantify, propagate and reduce the numerical uncertainty via computation within a Bayesian framework, which has not been fully investigated in context of probabilistic reliability analysis. In this study, a novel method termed 'Parallel Adaptive Bayesian Quadrature' (PABQ) is proposed on the theoretical basis of ALPI, and is aimed at broadening its scope of application. First, the Monte Carlo method used in ALPI is replaced with an importance ball sampling technique so as to reduce the sample size that is needed for rare failure event estimation. Second, a multi-point selection criterion is proposed to enable parallel distributed processing. Four numerical examples are studied to demonstrate the effectiveness and efficiency of the proposed method. It is shown that PABQ can effectively assess small failure probabilities (e.g., as low as 10^{-7}) with a minimum number of iterations by taking advantage of parallel computing.

Keywords: Reliability analysis, Gaussian process, Numerical uncertainty, Bayesian quadrature, Parallel computing

1. Introduction

In many fields, reliability analysis has manifested itself as an essential tool to study the performance of a physical or an engineering system in the presence of uncertainties. A fundamental task in reliability analysis is to compute the probability of a predefined failure event, which is referred as failure probability. Let $\boldsymbol{X} = [X_1, X_2, \cdots, X_d] \in \mathcal{X} \subseteq \mathbb{R}^d$ denote a vector of d random variables with known joint probability

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

density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x})$. The performance function (also known as limit state function) is given as $y = g(\boldsymbol{x}) : \mathcal{X} \mapsto \mathcal{Y}$, by which the failure event $F = \{\boldsymbol{x} \in \mathcal{X} : g(\boldsymbol{x}) \leq 0\}$ is defined. The associated failure probability P_f is defined by the following multi-dimensional integral:

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

where $I(\boldsymbol{x})$ is the failure indicator function, which is defined as:

$$I(\boldsymbol{x}) = \begin{cases} 1, \text{if } g(\boldsymbol{x}) \leq 0 \\ 0, \text{otherwise} \end{cases}. \tag{2}$$

To assess the failure probability defined in Eq. (1), a variety of numerical methods have been extensively studied and applied by researchers and engineers over the past several decades. In general, the existing methods can be roughly classified into five categories:

1. Stochastic simulation methods, e.g., Monte Carlo simulation (MCS) and its variants (e.g., Subset Simulation (SS) [1] and Importance Sampling (IS) [2, 3]). Despite of their relative robustness to the dimension and complexity of the problem at hand, most of the stochastic simulation methods involve a considerable number of deterministic simulations, and hence are still very computationally demanding, especially for an expensive computational model with a small failure probability;
2. Asymptotic approximation methods, such as first-order reliability method (FORM) [4, 5] and second-order reliability method (SORM) [6, 7]. This kind of methods relies on the first- or second-order Taylor expansion of the limit state surface at the most probable point (MPP). Hence, its application is challenging whenever one must deal with multiple MPPs and highly nonlinear problems. Besides, FORM and SORM only yield approximate results in general cases, and provide no measure of the error introduced by the expansion;
3. Moment methods, for instance, integer moment based methods [8, 9, 10], fractional moment based methods [11, 12, 13], moment-generating function (or Laplace transform) based methods [14, 15, 16]. The basic idea of these methods is to fit a proper probability distribution to the output variable of a performance function based on the knowledge of its estimated moments of certain type, which, however, typically leads to an ill-posed inverse problem (i.e., the so-called classical moment problem). Moreover, the estimation errors arising from both the estimated moments and assumed probability distribution model could be intractable to assess and handle;
4. Probability-conservation based methods, including, e.g., probability density evolution method [17, 18, 19, 20] and direct probability integral method [21, 22, 23, 24]. These methods are established on rigorous theoretical fundamentals, but may still suffer from numerical difficulty especially for problems with high-dimensional inputs and/or rare failure events;

5. Surrogate assisted methods. This type of methods is of special interest in the present paper since the proposed method also falls in this category in some sense.

Surrogate assisted methods aim at constructing an inexpensive-to-evaluate surrogate model in place of the original expensive-to-evaluate performance function based on a limited number of its observations. Then, for example, stochastic simulation methods can be directly applied in conjunction with the surrogate model to produce a failure probability estimate. Typical surrogate models for reliability analysis include response surface methods [25, 26, 27], support vector machines [28, 29, 30], polynomial chaos expansions [31, 32], Gaussian process regression (GPR, also known as Kriging) [33, 34, 35], etc. In addition to developing new surrogate models, there has been growing attention paid to adaptive (optimal) design of experiments for training these surrogates. In this line, the GPR model is of particular interest for constructing an adaptive meta-model due to its attractive features, especially for active learning sampling strategies. Representative learning functions consist of the expected feasibility [36], U [37], expected risk [38], H [39], least improvement [40], reliability-based expected improvement [41], folded normal based expected improvement [42], upper-bound posterior variance contribution (UPVC) [43] and so forth. Besides, the following three aspects have also been paid special attention to in recent publications:

1. assessing small failure probabilities. In addition to MCS, other stochastic simulation methods requiring less samples are combined with active learning Kriging (AK) to evaluate small failure probabilities. Representative works include, e.g., AK-IS [44], meta-IS [34], AK-SS [45] and AK-MCMC (Markov chain Monte Carlo) [46], etc;
2. addressing high-dimensional problems. This aspect is mainly tackled by using some dimension-reduction techniques, e.g., active subspace methods [47, 48], principal component analysis [49, 50, 51], sufficient dimension reduction [52] and sliced inverse regression [53], etc;
3. enabling parallel computing. Most existing learning functions can only identify one point at each iteration, hindering the use of ever-increasing parallel-computing facilities. To overcome this obstacle, tailored strategies have been proposed, which are mainly based on applying clustering algorithms, such as k-means clustering [34], density clustering [54], spectral clustering [55] and k-medoids clustering [56].

The interested reader can refer to [57] for a comprehensive review. Despite great efforts, most existing Kriging assisted methods still possess respective limitations, and leave room for further improvement in terms of applicability, efficiency and accuracy.

In fact, Gaussian process model can be used in a different way, instead of a pure surrogate model. The first author and his co-workers proposed an active learning probabilistic integration (ALPI) method in a recent paper [43]. In this method, a Bayesian perspective is advocated to reinterpret failure probability integral estimation. By placing a prior distribution (i.e., Gaussian process) over the performance function, we finally arrive at a posterior distribution over the failure probability conditional on some observations of

the performance function. The induced posterior distribution of the failure probability reflects the fact that the performance function has been discretised, and hence numerical uncertainty arises due to discretization error. A novel feature of ALPI is that the numerical uncertainty can be properly quantified, propagated and reduced via computation, which distinguishes it from other existing methods. Unfortunately, the idea is only investigated in the context of imprecise probabilities, and lacks of comprehensive studies for probabilistic reliability analysis.

In this paper, the ALPI method is specially studied under the framework of precise probabilities. The basic idea is explained in a detailed way, and some limitations existing in the previous numerical algorithm are identified. Most importantly, we propose a new method called 'Parallel Adaptive Bayesian Quadrature' (PABQ) on the theoretical basis of the original ALPI method, while alleviating its main limitations. Compared to ALPI, PABQ has two significant advantages. First, PABQ can select multiple points at each iteration, and as such supports parallel distributed processing. Second, PABQ can assess very small failure probabilities without generating a prohibitively large number of candidate samples. Additionally, the Matlab code of the developed method is freely available to the public [1].

The outline of the remaining paper is as follows. The original ALPI method is revisited in Section 2 and the theoretical foundations are deepened. Section 3 gives the newly developed PABQ method. Four numerical examples are investigated in Section 4 to illustrate the performance of the PABQ method. Section 5 gives some concluding remarks of the present study.

2. Active learning probabilistic integration

This section gives a review of the ALPI method. In comparison to [43], we will explain the basic idea of ALPI in a more detailed and rigorous way, and provide its numerical algorithm that was omitted in [43]. Besides, the advantages and disadvantages of the method will be discussed.

2.1. Theoretical background

The ALPI method offers a Bayesian approach to approximating the intractable failure probability integral, which is defined in Eq. (1). The method is strongly motivated by Bayesian (probabilistic) integration (also well known as Bayesian quadrature or cubature) [58, 59, 60]. To be specific, the ALPI method turns the task of failure probability estimation into a Bayesian inference problem from limited data, as opposed to classical frequentist inference. To do so, we think of the g-function as being random. This is understandable in the Bayesian sense that the numerical value of $g(\boldsymbol{x})$ is always unknown until we actually evaluate $g(\cdot)$ at some point $\boldsymbol{x}$, though the g-function is said to be deterministic. Such interpretation is justified since we can not afford to compute $g(\cdot)$ at every possible location. In this regard, epistemic uncertainty due to

[1] to be released upon acceptance of the paper

discretisation error arises where the g-function is not evaluated. This kind of uncertainty will propagate into the failure indicator function $I(\boldsymbol{x})$ and will therefore affect the failure probability estimate. Consequently, the epistemic uncertainty should be properly treated within our computational framework, because it is not always negligible, especially when the available observations are scarce. Following a standard Bayesian approach, the ALPI method is intended to quantify, propagate and reduce the epistemic uncertainty. Specifically, ALPI first assigns a prior distribution over the g-function. Then, conditioning on some observations $\mathscr{D} = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$ ($\boldsymbol{\mathcal{X}} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{n}$ with $\boldsymbol{x}^{(i)}$ being the i-th row of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}} = \left\{y^{(i)}\right\}_{i=1}^{n}$ with $y^{(i)} = g(\boldsymbol{x}^{(i)})$ being the i-th row of $\boldsymbol{\mathcal{Y}}$), gives arise to a posterior distribution of g according to Bayes' rule. This will in turn imply a posterior distribution over $I(\boldsymbol{x})$, and so does over P_f. Technical details of ALPI will be discussed below.

ALPI starts by placing a Gaussian process (GP) prior over the g-function, which is written as:

$$\hat{g}_0 \sim \mathcal{GP}(m_{\hat{g}_0}(\boldsymbol{x}), k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}')), \tag{3}$$

where $\hat{g}_0$ denotes the prior distribution of g before seeing any observations; $m_{\hat{g}_0}(\boldsymbol{x})$ and $k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}')$ are the prior mean and covariance functions respectively, by which the GP model can be completely characterized. Among many options for $m_{\hat{g}_0}(\boldsymbol{x})$ and $k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}')$ in the literature, without loss of generality the constant prior mean is adopted (i.e., $m_{\hat{g}_0}(\boldsymbol{x}) = \beta$), and the prior covariance function takes the squared exponential kernel:

$$k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')\,\boldsymbol{\Sigma}^{-1}\,(\boldsymbol{x} - \boldsymbol{x}')^{\top}\right), \tag{4}$$

where σ^2 with $\sigma > 0$ denotes the process variance; $\boldsymbol{\Sigma} = \mathrm{diag}(l_1^2, l_2^2, \cdots, l_d^2)$ with $l_i > 0$ being the length scale in the i-th dimension, and $\mathrm{diag}(\cdot)$ forms a diagonal matrix whose diagonal elements are its arguments. The $d+2$ parameters collected in $\boldsymbol{\vartheta} = \{\beta, \sigma, l_1, l_2, \cdots, l_d\}$ are referred to hyper-parameters to be determined. In a fully Bayesian fashion, those hyper-parameters should also be specified by Bayesian inference (see, e.g., [61]). However, this will render the posterior distribution of g analytically intractable. For this reason, it was not explored in ALPI.

Alternatively, given the data $\mathscr{D}$, the hyper-parameters are fitted by minimizing the negative log marginal likelihood (NLML) $\mathcal{L}(\boldsymbol{\vartheta})$:

$$\hat{\boldsymbol{\vartheta}} = \arg\min_{\boldsymbol{\vartheta}} \mathcal{L}(\boldsymbol{\vartheta}), \tag{5}$$

with the NLML $\mathcal{L}(\boldsymbol{\vartheta})$ being:

$$\mathcal{L}(\boldsymbol{\vartheta}) = -\log\left[p(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}, \boldsymbol{\vartheta})\right] = \frac{1}{2}(\boldsymbol{\mathcal{Y}} - \beta)^{\top}\boldsymbol{K}_{\hat{g}_0}^{-1}(\boldsymbol{\mathcal{Y}} - \beta) + \frac{1}{2}\log\left[|\boldsymbol{K}_{\hat{g}_0}|\right] + \frac{n}{2}\log[2\pi], \tag{6}$$

where $p(\boldsymbol{\mathcal{Y}}|\boldsymbol{\mathcal{X}}, \boldsymbol{\vartheta})$ is the marginal likelihood following a normal distribution; $\boldsymbol{K}_{\hat{g}_0}$ is the covariance matrix with (i, j)-th entry $[\boldsymbol{K}_{\hat{g}_0}]_{i,j} = k_{\hat{g}_0}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$.

Once the point estimate hyper-parameters $\hat{\boldsymbol{\vartheta}}$ are obtained, it turns out that the posterior distribution of g can be derived in closed form, i.e., another GP:

$$\hat{g}_n \sim \mathcal{GP}(m_{\hat{g}_n}(\boldsymbol{x}), k_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')), \tag{7}$$

where $\hat{g}_n$ denotes the posterior distribution of g conditional on $\mathscr{D}$; $m_{\hat{g}_n}(\boldsymbol{x})$ and $k_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')$ are the posterior mean and covariance functions respectively, which are analytically available:

$$m_{\hat{g}_n}(\boldsymbol{x}) = m_{\hat{g}_0}(\boldsymbol{x}) + \boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}, \mathscr{X})^\top \boldsymbol{K}_{\hat{g}_0}^{-1} \left(\mathscr{Y} - \boldsymbol{m}_{\hat{g}_0}(\mathscr{X})\right), \tag{8}$$

$$k_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') = k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}, \mathscr{X})^\top \boldsymbol{K}_{\hat{g}_0}^{-1} \boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}', \mathscr{X}), \tag{9}$$

where $\boldsymbol{m}_{\hat{g}_0}(\mathscr{X})$ is an $n \times 1$ mean vector with i-th element being $m_{\hat{g}_0}(\boldsymbol{x}^{(i)})$; $\boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}, \mathscr{X})$ is an $n \times 1$ covariance vector with i-th entry being $k_{\hat{g}_0}\left(\boldsymbol{x}, \boldsymbol{x}^{(i)}\right)$; $\boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}', \mathscr{X})$ is defined in a way similar to $\boldsymbol{k}_{\hat{g}_0}(\boldsymbol{x}, \mathscr{X})$. Note that in Eqs. (8) and (9) $\boldsymbol{\vartheta}$ should be updated with $\hat{\boldsymbol{\vartheta}}$.

It can be deduced that the posterior distribution of failure indicator function I follows a generalized Bernoulli process [2] (GBP):

$$\hat{I}_n \sim \mathcal{GBP}(m_{\hat{I}_n}(\boldsymbol{x}), k_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}')), \tag{10}$$

where $\hat{I}_n$ denotes the posterior distribution of I conditional on $\mathscr{D}$; $m_{\hat{I}_n}(\boldsymbol{x})$ and $k_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}')$ are the posterior mean and covariance functions respectively. The posterior mean of I can be derived in closed form [43]:

$$m_{\hat{I}_n}(\boldsymbol{x}) = \Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right), \tag{11}$$

where Φ is the cumulative distribution function of the standard normal distribution; $\sigma_{\hat{g}_n}(\boldsymbol{x})$ is the posterior standard derivation (STD) function of g, i.e., $\sigma_{\hat{g}_n}(\boldsymbol{x}) = \sqrt{k_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x})}$. The posterior covariance function of I, however, is not analytically tractable. Only closed-form expression for its posterior variance function $\sigma^2_{\hat{I}_n}(\boldsymbol{x})$ is available [43]:

$$\sigma^2_{\hat{I}_n}(\boldsymbol{x}) = \Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right). \tag{12}$$

The posterior distribution $\hat{P}_{f,n}$ of failure probability P_f conditional on the data $\mathscr{D}$ should thus follow a random variable, which reflects our epistemic uncertainty about P_f, due to the limited number of observations. Note that the exact posterior distribution of P_f, however, is not known. Instead, the posterior mean and variance of P_f should be more of interest, where the posterior mean corresponds to the failure probability predictor and the posterior variance measures the prediction uncertainty. By applying Fubini's

[2] 'generalized' indicates that the Bernoulli process considered here is location-dependent, in contrast to not considering the dependence in conventional definition of a Bernoulli process.

theorem, the posterior mean and variance of P_f can be derived as [43]:

$$\begin{aligned} m_{\hat{P}_{f,n}} &= \mathbb{E}_{\hat{I}_n}\left[\hat{P}_{f,n}\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\int_{\mathcal{X}} \hat{I}_n(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right] \\ &= \int_{\mathcal{X}} \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right] f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int_{\mathcal{X}} m_{\hat{I}_n}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int_{\mathcal{X}} \Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{13}$$

$$\begin{aligned} \sigma^2_{\hat{P}_{f,n}} &= \mathbb{V}_{\hat{I}_n}\left[\hat{P}_{f,n}\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\left(\hat{P}_{f,n} - \mathbb{E}_{\hat{I}_n}\left[\hat{P}_{f,n}\right]\right)^2\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\left(\int_{\mathcal{X}} \hat{I}_n(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \int_{\mathcal{X}} \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right] f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right)^2\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\left(\int_{\mathcal{X}} \left(\hat{I}_n(\boldsymbol{x}) - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right)^2\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\left(\int_{\mathcal{X}} \left(\hat{I}_n(\boldsymbol{x}) - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right) \times \left(\int_{\mathcal{X}} \left(\hat{I}_n(\boldsymbol{x}') - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x}')\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}'\right)\right] \\ &= \mathbb{E}_{\hat{I}_n}\left[\int_{\mathcal{X}}\int_{\mathcal{X}} \left(\hat{I}_n(\boldsymbol{x}) - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right]\right)\left(\hat{I}_n(\boldsymbol{x}') - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x}')\right]\right) f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'\right] \\ &= \int_{\mathcal{X}}\int_{\mathcal{X}} \mathbb{E}_{\hat{I}_n}\left[\left(\hat{I}_n(\boldsymbol{x}) - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x})\right]\right)\left(\hat{I}_n(\boldsymbol{x}') - \mathbb{E}_{\hat{I}_n}\left[\hat{I}_n(\boldsymbol{x}')\right]\right)\right] f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\ &= \int_{\mathcal{X}}\int_{\mathcal{X}} k_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}') f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}', \end{aligned} \tag{14}$$

where $\mathbb{E}_{\hat{I}_n}[\cdot]$ and $\mathbb{V}_{\hat{I}_n}[\cdot]$ denote expectation and variance operators taken over $\hat{I}_n$ respectively. For computational purposes, Eq. (14) is further simplified by considering its upper bound. According to the Cauchy-Schwarz inequality ($k_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}') \leq \sigma_{\hat{I}_n}(\boldsymbol{x})\,\sigma_{\hat{I}_n}(\boldsymbol{x}')$), an upper-bound of the posterior variance (UPV) $\sigma^2_{\hat{P}_{f,n}}$ is given as [43]:

$$\sigma^2_{\hat{P}_{f,n}} \leq \overline{\sigma}^2_{\hat{P}_{f,n}} = \left(\int_{\mathcal{X}} \sqrt{\Phi\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)\Phi\left(\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)} f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right)^2, \tag{15}$$

where the equality holds when the correlation of $\hat{I}_n$ between any two locations $(\boldsymbol{x}, \boldsymbol{x}')$ is always equal to 1, and $\overline{\sigma}^2_{\hat{P}_{f,n}}$ denotes the upper-bound of the posterior variance.

At the theoretical level, ALPI provides two important benefits. First, it offers a principled approach to the quantification and propagation of numerical uncertainty via computation within the Bayesian framework. Second, it gives the possibility to reduce the numerical uncertainty by using an active learning strategy (see next subsection).

2.2. Numerical algorithm

For practical reliability analysis, the failure probability estimate should be inferred using as few observations as possible, with the premise of limiting its numerical uncertainty within a pre-specified tolerance. Besides, as the posterior mean and UPV of failure probability (Eqs. (13) and (15)) lack of closed-form solutions, a numerical integrator is necessary to make the method practically feasible. The numerical algorithm of the ALPI method for failure probability estimation is summarized in Appendix A.

When it comes to numerical implementation, ALPI shows two main limitations. First, it is not applicable to problems with extremely small failure probabilities (typically, less than 10^{-4}) as a large number of Monte Carlo (MC) samples (typically, more than 10^6) are required, making each iteration computationally cumbersome and even infeasible. Second, it is not suitable for parallel computing since only one point is identified at each iteration, resulting in a waste of useful information and computational resources for engineering applications.

3. Parallel adaptive Bayesian quadrature

The major limitations of ALPI at implementation level will be addressed in this section. Further, a novel method, called 'Parallel Adaptive Bayesian Quadrature' (PABQ), is presented on the theoretical basis of ALPI. As its name indicates, the proposed PABQ method can support parallel distributed processing. Most importantly, PABQ is able to estimate very small failure probabilities (e.g., 10^{-7}).

3.1. General remarks

As we did not imply any distribution types for $\boldsymbol{X}$ when making Bayesian inference about the failure probability in the last section, it means that the ALPI framework is naturally applicable in the standard normal space. In view of this, let us transform $g(\boldsymbol{x})$ from the physical space $\mathcal{X}$ to the standard normal space $\mathcal{U}$, i.e., $g(\boldsymbol{x}) = g(T^{-1}(\boldsymbol{u})) = \mathcal{G}(\boldsymbol{u})$, where $\boldsymbol{u}$ is a realization of the standard normal vector $\boldsymbol{U} = [U_1, U_2, \cdots, U_d] \in \mathcal{U} \subseteq \mathbb{R}^d$ and T^{-1} is the inverse transformation (e.g., iso-probabilistic, Nataf, and Rosenblatt transformation, etc.). For clarification, the transformed performance function is denoted as $Z = \mathcal{G}(\boldsymbol{U})$. Different from ALPI, the proposed PABQ method will be implemented with the $\mathcal{G} = g \circ T^{-1}$-function.

3.2. Importance ball sampling

In this subsection, we propose an importance ball sampling (IBS) technique to replace the MC method used in the conventional ALPI method. Let us first introduce a ball, a region enclosed by a sphere or hypersphere. The d-ball of radius $r > 0$ in the standard normal space $\mathcal{U}$ can be defined as $B^d(r) = \left\{\boldsymbol{u} \in \mathbb{R}^d : ||\boldsymbol{u}||_2 \leq r\right\}$, where $||\cdot||_2$ denotes the 2-norm. The ball is said to be 'important' when it can cover

the standard normal space with relatively large probability content (in case that r is appropriately chosen). The uniform PDF over $B^d(r)$ takes the form:

$$f_B(\boldsymbol{u}) = \begin{cases} \frac{1}{V_d(r)}, & \text{if } ||\boldsymbol{u}||_2 \leq r; \\ 0, & \text{otherwise} \end{cases}, \tag{16}$$

where $V_d(r) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} r^d$ is the volume of $B^d(r)$, $\Gamma(\cdot)$ is Euler's gamma function. To generate random points uniformly distributed within the d-ball, there are many methods available in the literature. In this study, one algorithm reported in [62] is adopted, as summarized in Algorithm 1.

Algorithm 1 Generate uniform samples within the d-ball [62]

1: **Input:** dimension d, radius r and sample size N_{ibs}
2: **for** $i = 1, 2, \cdots, N_{ibs}$ **do**
3: Generate d normally distributed samples, $\boldsymbol{w} = [w^{(1)}, w^{(2)}, \cdots, w^{(d)}], w^{(i)} \sim \mathcal{N}(0,1)$
4: Generate a uniformly distributed sample v from the interval $[0,1]$
5: Return the i-th vector $\overline{\boldsymbol{u}}^{(i)} = \frac{r v^{1/d} \boldsymbol{w}}{||\boldsymbol{w}||_2}$
6: **end for**
7: **Output:** $\overline{\mathscr{U}} = \left\{\overline{\boldsymbol{u}}^{(i)}\right\}_{i=1}^{N_{ibs}}$: N_{ibs} uniform samples in $B^d(r)$

Then, consider an auxiliary PDF constructed as follows:

$$f_0(\boldsymbol{u}) = \begin{cases} (1-\Delta) f_B(\boldsymbol{u}), ||\boldsymbol{u}||_2 \leqslant r \\ f_{\boldsymbol{U}}(\boldsymbol{u}), \text{otherwise} \end{cases}, \tag{17}$$

where $f_{\boldsymbol{U}}(\boldsymbol{u})$ is the joint PDF of $\boldsymbol{U}$; Δ is a normalizing constant that ensures that the PDF $f_0(\boldsymbol{u})$ integrates to one, which is actually equal to the probability of $f_{\boldsymbol{U}}(\boldsymbol{u})$ outside $B^d(r)$, i.e., $\Delta = \int_{\mathcal{U}\setminus\mathcal{B}} f_{\boldsymbol{U}}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$. The posterior mean $m_{\hat{P}_{f,n}}$ and upper-bound of posterior standard deviation (UPSTD) $\overline{\sigma}_{\hat{P}_{f,n}}$ with respect to the $\mathcal{G}$-function can be reformulated respectively as:

$$\begin{aligned} m_{\hat{P}_{f,n}} &= \int_{\mathcal{U}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) f_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\ &= \int_{\mathcal{U}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{f_0(\boldsymbol{u})} f_0(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\ &= \int_{\mathcal{B}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{(1-\Delta) f_B(\boldsymbol{u})} (1-\Delta) f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\ &\quad + \int_{\mathcal{U}\setminus\mathcal{B}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{f_{\boldsymbol{U}}(\boldsymbol{u})} f_{\boldsymbol{U}}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\ &= V_d(r) \int_{\mathcal{B}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) f_{\boldsymbol{U}}(\boldsymbol{u}) f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\ &\quad + \int_{\mathcal{U}\setminus\mathcal{B}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) f_{\boldsymbol{U}}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}, \end{aligned} \tag{18}$$

$$
\begin{aligned}
\overline{\sigma}_{\hat{P}_{f,n}} &= \int_{\mathcal{U}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} f_{\boldsymbol{U}}(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \\
&= \int_{\mathcal{U}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{f_0(\boldsymbol{u})} f_0(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\
&= \int_{\mathcal{B}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{(1-\Delta)f_B(\boldsymbol{u})}(1-\Delta)f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\
&\quad + \int_{\mathcal{U}\backslash\mathcal{B}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} \frac{f_{\boldsymbol{U}}(\boldsymbol{u})}{f_{\boldsymbol{U}}(\boldsymbol{u})} f_{\boldsymbol{U}}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\
&= V_d(r) \int_{\mathcal{B}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} f_{\boldsymbol{U}}(\boldsymbol{u}) f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} \\
&\quad + \int_{\mathcal{U}\backslash\mathcal{B}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} f_{\boldsymbol{U}}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u},
\end{aligned}
\tag{19}
$$

where $m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})$ and $\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})$ are the posterior mean and STD functions of $\mathcal{G}$ conditional on n observations. Note that if one chooses a sufficiently small Δ (i.e., r is sufficiently large), $f_{\boldsymbol{U}}(\boldsymbol{u})$ over $\mathcal{U}\backslash\mathcal{B}$ will approach to zero. In this case, the last terms in both Eqs. (18) and (19) can be neglected, and hence $m_{\hat{P}_{f,n}}$ and $\overline{\sigma}_{\hat{P}_{f,n}}$ are approximately equal to:

$$
m_{\hat{P}_{f,n}} \approx V_d(r) \int_{\mathcal{B}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right) f_{\boldsymbol{U}}(\boldsymbol{u}) f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}, \tag{20}
$$

$$
\overline{\sigma}_{\hat{P}_{f,n}} \approx V_d(r) \int_{\mathcal{B}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} f_{\boldsymbol{U}}(\boldsymbol{u}) f_B(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}. \tag{21}
$$

The above two equations are the basic of the proposed IBS method, and $f_B(\boldsymbol{u})$ is regarded as the importance sampling density. The IBS estimators of Eqs. (20) and (21) are given as:

$$
\tilde{m}_{\hat{P}_{f,n}} = \frac{V_d(r)}{N_{ibs}} \sum_{i=1}^{N_{ibs}} \Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right) f_{\boldsymbol{U}}(\overline{\boldsymbol{u}}^{(i)}), \tag{22}
$$

$$
\tilde{\overline{\sigma}}_{\hat{P}_{f,n}} = \frac{V_d(r)}{N_{ibs}} \sum_{i=1}^{N_{ibs}} \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right)} f_{\boldsymbol{U}}(\overline{\boldsymbol{u}}^{(i)}), \tag{23}
$$

where $\overline{\boldsymbol{u}}^{(i)} \sim f_{\mathcal{B}}(\boldsymbol{u})$. The variances of the estimators are formulated as follows:

$$
\mathbb{V}\left[\tilde{m}_{\hat{P}_{f,n}}\right] = \frac{V_d^2(r)}{(N_{ibs}-1)N_{ibs}} \sum_{i=1}^{N_{ibs}} \left[\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right) f_{\boldsymbol{U}}(\overline{\boldsymbol{u}}^{(i)}) - \tilde{m}_{\hat{P}_{f,n}}\right]^2, \tag{24}
$$

$$
\mathbb{V}\left[\tilde{\overline{\sigma}}_{\hat{P}_{f,n}}\right] = \frac{V_d^2(r)}{(N_{ibs}-1)N_{ibs}} \sum_{i=1}^{N_{ibs}} \left[\sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\overline{\boldsymbol{u}}^{(i)})}{\sigma_{\hat{\mathcal{G}}_n}\left(\overline{\boldsymbol{u}}^{(i)}\right)}\right)} f_{\boldsymbol{U}}(\overline{\boldsymbol{u}}^{(i)}) - \tilde{\overline{\sigma}}_{\hat{P}_{f,n}}\right]^2. \tag{25}
$$

Similar to the MC population in ALPI, the population generated from $f_{\mathcal{B}}(\boldsymbol{u})$ also plays two roles. First, the posterior mean and UPSTD of the failure probability should be evaluated numerically based on those samples at each iteration, as shown in Eqs. (22) and (23). Second, it will be used as a candidate sample pool by which multiple promising points can be identified at each iteration (see next subsection).

Given dimension d, the IBS method has two parameters to be specified appropriately, i.e., radius r and simple size N_{ibs}. As we have mentioned, r should be large enough to ensure that Δ is small enough. By doing so, (1) the bias between Eqs. (20) and (18), and also Eqs. (21) and (19) can be neglected; (2) candidate samples can reach the failure domain characterized by a small probability. The probability of $f_{\boldsymbol{U}}(\boldsymbol{u})$ within the d-ball can be given as (see Appendix E of [63]):

$$F(d,r) = \frac{1}{\Gamma(d/2)} \int_0^{r^2/2} x^{d/2-1} \exp(-x) \mathrm{d}x. \tag{26}$$

Based on this, r can be determined as the solution to:

$$F(d,r) = 1 - \Delta, \tag{27}$$

It should be noted that given a fixed Δ, r increases with d. For example, if we set Δ as 10^{-8}, $r \approx 6.07$ for $d = 2$, and $r \approx 6.77$ for $d = 5$. As for N_{ibs}, it cannot be too small otherwise the estimators of $m_{\hat{P}_{f,n}}$ and $\overline{\sigma}_{\hat{P}_{f,n}}$ will process relatively large variances, and also cannot fill the d-ball well. On the contrary, a too large N_{ibs} can lead to numerical difficulty and memory problems.

As an illustration, Fig. 1 shows two populations generated respectively by MC and IBS in two dimensions with the same sample size 10^5. Obviously, the IBS method can produce a better space-filling population and cover a larger area than that of MC method. If one would like the MC population to cover as large space as the IBS population, the sample size should be increased many times ($> 10^3$).

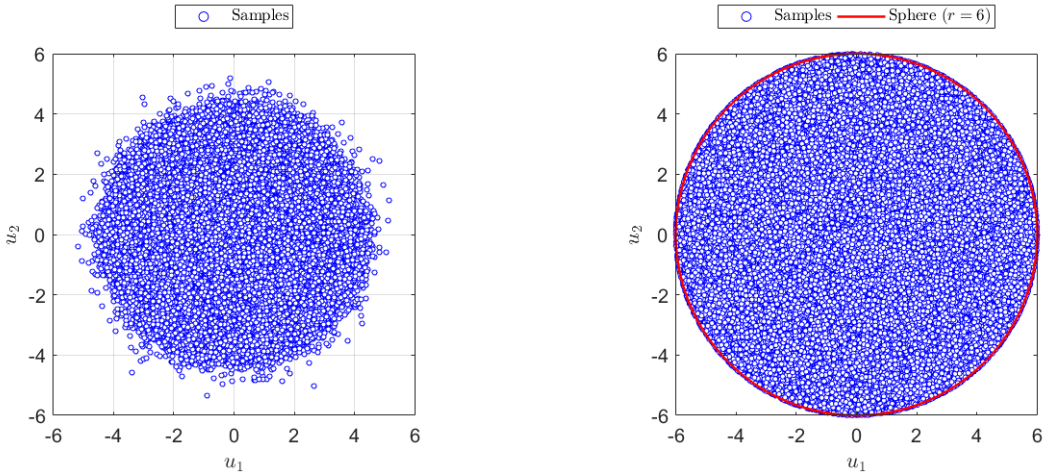
3.3. Multi-point UPVC criterion

In order to enable parallel processing, a batch of informative points should be identified to evaluate on the $\mathcal{G}$-function at each iteration, rather than only one single point. For this purpose, we propose a multi-point UPVC criterion, which leverages the advantages of both the UPVC function [43] and k-means clustering [64].

Suppose that we have inferred a GP posterior $\hat{\mathcal{G}}_n \sim \mathcal{GP}(m_{\hat{\mathcal{G}}_n}(\boldsymbol{u}), k_{\hat{\mathcal{G}}_n}(\boldsymbol{u}, \boldsymbol{u}'))$ of $\mathcal{G}$ at a certain step of the proposed PABQ method. Analogous to Eq. (A.3), the corresponding UPVC function can be defined as:

$$\mathrm{UPVC}(\boldsymbol{u}) = \sqrt{\Phi\left(-\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)\Phi\left(\frac{m_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}{\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u})}\right)} \times f_{\boldsymbol{U}}(\boldsymbol{u}), \tag{28}$$

where $\sigma_{\hat{\mathcal{G}}_n}(\boldsymbol{u}) = \sqrt{k_{\hat{\mathcal{G}}_n}(\boldsymbol{u}, \boldsymbol{u})}$ is the posterior STD function of $\mathcal{G}$. Note that $\overline{\sigma}^2_{\hat{P}_{f,n}} = \left[\int_{\mathcal{U}} \mathrm{UPVC}(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}\right]^2$ holds, and hence the UPVC function is a measure of the contribution of numerical uncertainty at the point



(a) MC method ($N_{mc} = 10^5$) (b) IBS method ($N_{ibs} = 10^5$)

Figure 1: Comparison between MC and IBS methods in two dimensions.

$\boldsymbol{u}$ to the UPV of the failure probability. The traditional UPVC criterion, however, only selects the point among a MC population that has the maximum UPVC value as the best next point to evaluate on the $\mathcal{G}$-function. As such, other information provided by the UPVC function that might be still useful is discarded at each iteration. This drawback can be alleviated by identifying multiple points. The conventional k-means clustering technique can be used to partition $\overline{\mathscr{U}}$ into k clusters, but it cannot take the UPVC measure into account. In this study, a weighted clustering algorithm is proposed by combining the UPVC function with k-means clustering, which is referred as 'UPVC-weighted k-means clustering'. Suppose that we wish to select q points among $\overline{\mathscr{U}}$ at each iteration, and hence evaluation of the $\mathcal{G}$-function at these q points can be distributed on q processors simultaneously. The number of points q also corresponds to the number of clusters. A compact pseudocode of the proposed algorithm is given in Algorithm 2. The selected q points correspond to the q centroids produced by the proposed UPVC-weighted k-means clustering. It should be pointed out that the identified points usually do not belong to $\overline{\mathscr{U}}$ any more due to the weighted averaging operator embedded in the proposed algorithm.

A test example is considered here to illustrate the proposed multi-point UPVC criterion. The performance function is given as $Z = \mathcal{G}(\boldsymbol{U}) = U_1^2 - U_2 + 2$, where U_1 and U_2 are two independent standard normal variables. For reproducibility, we specify the initial observed locations as $\mathscr{U} = \{(-\sqrt{5}, 0), (0, 0), (\sqrt{5}, 0), (0, -\sqrt{5}), (0, \sqrt{5})\}$. Based on these five initial observations, we can obtain a posterior GP over the $\mathcal{G}$-function and also the UPVC function. Additional $q = 5$ points are then identified by the proposed UPVC-weighted k-means clustering algorithm from 10^5 uniform samples within the 2-ball of $r = 6$. As shown in Fig. 2, the newly selected points are sparsely located in areas where the UPVC values are not very small. Therefore, the total information gained from those 5 points could be more than that of the one with the maximum UPVC value.

Algorithm 2 UPVC-weighted k-means clustering algorithm

Input: the number of clusters q, UPVC($\boldsymbol{u}$) and $\overline{\mathscr{U}}$

1. **Initialization**. Randomly select q points among $\overline{\mathscr{U}}$ as the initial centroids , denoted by $\boldsymbol{E}^{(1)} = \left\{\boldsymbol{e}^{(i)}\right\}_{i=1}^{q}$;

2. **Assignment step**. Each point among $\overline{\mathscr{U}}$ is assigned to a cluster for which the squared Euclidean distance between the point and the cluster centroid is shortest. The i-th cluster is denoted as $\boldsymbol{C}^{(i)} = \left\{\boldsymbol{c}_j^{(i)}\right\}_{j=1}^{N_i}$, where $\boldsymbol{c}_j^{(i)}$ is the j-th point in the i-th cluster ($j = 1, 2, \cdots, N_i$);

3. **Update step**. Each centroid is then updated by UPVC-weighted mean of the cluster:

$$\boldsymbol{e}^{(i)} = \frac{\sum_{j=1}^{N_i} \mathrm{UPVC}(\boldsymbol{c}_j^{(i)}) \times \boldsymbol{c}_j^{(i)}}{\sum_{j=1}^{N_i} \mathrm{UPVC}(\boldsymbol{c}_j^{(i)})}$$

4. **Iteration**. Repeat the assignment step and update step until the centroids do not change or the predefined number of iterations is reached.

Output: q centroids

3.4. Summary of the proposed method

The numerical implementation procedure of the proposed PABQ method for reliability analysis, which is also shown in Fig. 3, consists of the following main steps:

Step B.1: Generate uniformly distributed samples within the d-ball

Generate N_{ibs} uniform samples within the d-ball of radius r, using Algorithm 1, denoted as $\overline{\mathscr{U}} = \left\{\overline{\boldsymbol{u}}^{(i)}\right\}_{i=1}^{N_{ibs}}$.

Step B.2: Get initial observations

Randomly select N_0 samples among $\overline{\mathscr{U}}$, denoted by $\mathscr{U} = \left\{\boldsymbol{u}^{(i)}\right\}_{i=1}^{N_0}$. These samples are evaluated on the

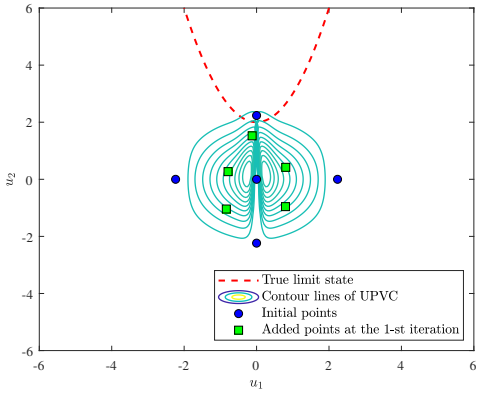


Figure 2: Illustration of the proposed multi-points UPVC criterion by a test example.

$\mathcal{G}$-function in parallel to obtain the corresponding observations $\mathscr{Z} = \left\{z^{(i)}\right\}_{i=1}^{N_0}$ ($z^{(i)} = \mathcal{G}(\boldsymbol{u}^{(i)})$). The initial dataset is constructed by $\mathscr{D} = \{\mathscr{U}, \mathscr{Z}\}$. Let $n = N_0$.

Step B.3: Make Bayesian inference about the failure probability

By assigning a GP prior for the $\mathcal{G}$-function, we finally arrive at the posterior mean and UPSTD of the failure probability conditional on $\mathscr{D}$. In this study, the prior mean and covariance of $\hat{\mathcal{G}}_0 \sim \mathcal{GP}(m_{\hat{\mathcal{G}}_0}(\boldsymbol{x}), k_{\hat{\mathcal{G}}_0}(\boldsymbol{x}, \boldsymbol{x}'))$ are assumed to be a constant and the squared exponential kernel respectively. The involved hyper-parameters are tuned by using the maximum likelihood estimation, and this stage is implemented with the *fitrgp* function in Statistics and Machine Learning Toolbox of Matlab. The posterior mean and UPSTD of the failure probability are then evaluated based on Eqs. (22) and (23).

Step B.4: Check the stopping criterion

If $\frac{\tilde{\bar{\sigma}}_{\hat{P}_{f,n}}}{\tilde{m}_{\hat{P}_{f,n}}} < \epsilon$ is satisfied, go to **Step B.6**; Else, go to **Step B.5**. Here $\frac{\tilde{\bar{\sigma}}_{\hat{P}_{f,n}}}{\tilde{m}_{\hat{P}_{f,n}}}$ denotes the estimated upper-bound of the posterior COV of the failure probability, and ϵ is a user-specified threshold.

Step B.5: Enrich the previous dataset

Identify additional q points by using the proposed multi-point UPVC criterion (see Algorithm 2), denoted by $\mathscr{U}_+ = \left\{\boldsymbol{u}_+^{(i)}\right\}_{i=1}^{q}$. Then, the corresponding observations of the $\mathcal{G}$-function at those q identified points $\mathscr{U}_+$ should be obtained using parallel computing, denoted by $\mathscr{Z}_+ = \left\{z_+^{(i)}\right\}_{i=1}^{q}$ with $z_+^{(i)} = \mathcal{G}(\boldsymbol{u}_+^{(i)})$. The previous dataset $\mathscr{D}$ is enriched with $\mathscr{D}_+ = \{\mathscr{U}_+, \mathscr{Z}_+\}$, i.e., $\mathscr{D} = \mathscr{D} \cup \mathscr{D}_+$. Let $n = n + q$, and go to **Step B.3**.

Step B.6: End the algorithm

Return $\tilde{m}_{\hat{P}_{f,n}}$ as the estimated failure probability and end the algorithm.

For practical implementation, it is necessary to set proper values for constants N_{ibs}, r, N_0, ϵ and q. The selection of these parameters is problem-dependent. However, according to our experience some general guidelines for selecting them are the following: $N_{ibs} = 5 \times 10^5 \sim 1 \times 10^6$, $r = 6$, $N_0 = 10$, $\epsilon = 5\% \sim 10\%$ and $q =$ the number of available processors for parallel computing.

4. Numerical examples

The performance of the proposed PABQ method is investigated by means of four numerical examples with varying complexity in this section. Several different parameter settings of PABQ are experimented in each example to study their effect on the results. For comparison, several state-of-the-art methods, i.e., FORM [65], SORM [65], AK-MCS [37], ALPI [43], AK-MCMC [46] and Polynomial-Chaos Kriging (PC-Kriging) [66], are also implemented when applicable.

4.1. Example 1: A series system with four branches

The first numerical example consists of a series system with four branches, which has been a classical benchmark example in structural reliability analysis (see, e.g., [67, 37, 68]). The performance function is

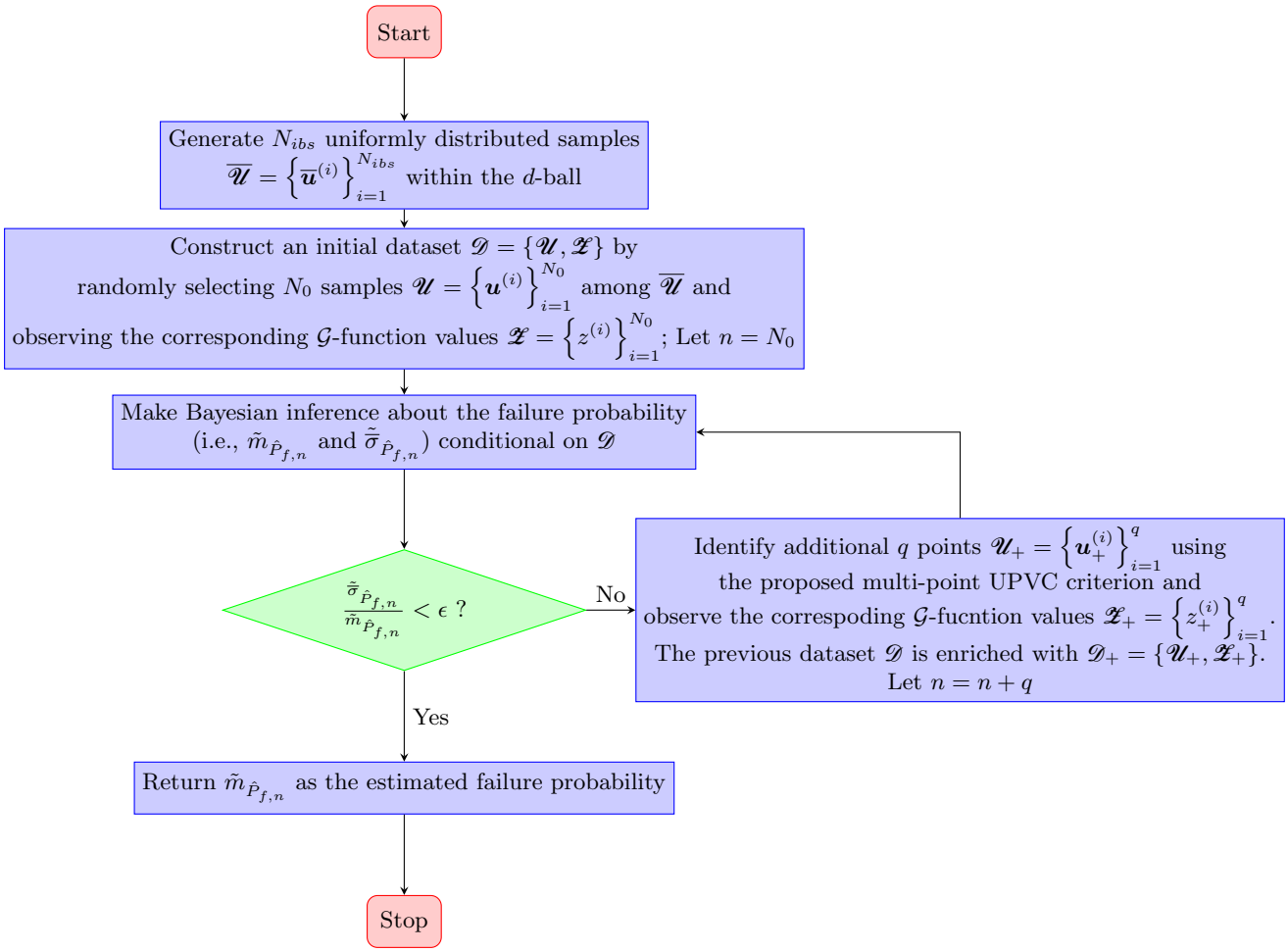


Figure 3: Flowchart of the proposed PABQ method.

given by:

$$Y = g\left(X_1, X_2\right) = \min \begin{cases} a + \frac{(X_1 - X_2)^2}{10} - \frac{(X_1 + X_2)}{\sqrt{2}}; \\ a + \frac{(X_1 - X_2)^2}{10} + \frac{(X_1 + X_2)}{\sqrt{2}}; \\ (X_1 - X_2) + \frac{b}{\sqrt{2}}; \\ (X_2 - X_1) + \frac{b}{\sqrt{2}} \end{cases}, \tag{29}$$

where X_1 and X_2 are two i.i.d. standard normal variables; a and b are two constant parameters. In this example, two cases by varying those two constant parameters are considered, where $a = 3$, $b = 6$ for the first case, and $a = 5$, $b = 10$ for the second case.

4.1.1. Results of Case I

In this case, the proposed PABQ method is compared to several other methods, i.e., AK-MCS+U [37], ALPI [43] and PC-Kriging [66]. Table 1 summarizes the results given as the number of iterations N_{iter}, the

total number of performance function calls N_{call}, the estimated failure probability $\hat{P}_f$, and the COV of $\hat{P}_f$ (i.e., COV$[\hat{P}_f]$). As seen, the proposed method with different q only takes a very few iterations in average to converge, which are less than that of PC-Kriging, and far less than that of AK-MCS+U and ALPI. This indicates that the proposed method could offer significant time savings when parallel computing is available. Furthermore, the computational advantage may still exist even in case of non-parallel computing since the average number of performance function calls is also reduced a lot by using the proposed method, especially when q is small (e.g., $q = 6$). The results of $\hat{P}_f$ and COV$[\hat{P}_f]$ also imply that the proposed method has an accuracy similar to other methods being compared. By increasing N_{ibs} from 5×10^5 to 1×10^6 and decreasing ϵ from 10% to 8%, the PABQ method can slightly reduce the COVs of failure probability estimates, at the cost of marginally increased computation in an average sense.

To illustrate the proposed method visually, Fig. 4(a) depicts the points selected at two stages of an exemplary run, as well as the true limit state curve. It is shown that most of the added points are sparsely located, and some of them are close to the four important parts of the limit state curve that are crucial for accurate failure probability estimation. These results indicate the effectiveness of the proposed multi-point selection strategy.

Table 1: Reliability results for Example 1 (Case I).

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^8	4.46×10^{-3}	0.15
AK-MCS+U		$1+96.55=97.55$	$12+96.55=108.55$	4.44×10^{-3}	1.54
ALPI		$1+72.95=73.95$	$12+72.95=84.95$	4.44×10^{-3}	1.79
PC-Kriging [66]	$q=6$	$1+14.40=15.40$	$12+86.40=98.40$	4.46×10^{-3}	1.50
Proposed PABQ ($N_{ibs}=5\times10^5, \epsilon=10\%$)	$q=6$	$1+5.60=6.60$	$10+33.60=43.60$	4.44×10^{-3}	2.53
	$q=10$	$1+4.20=5.20$	$10+42.00=52.00$	4.40×10^{-3}	2.22
	$q=15$	$1+3.65=4.65$	$10+54.75=64.75$	4.44×10^{-3}	1.35
	$q=20$	$1+3.05=4.05$	$10+61.00=71.00$	4.44×10^{-3}	1.29
Proposed PABQ ($N_{ibs}=1\times10^6, \epsilon=8\%$)	$q=6$	$1+8.64=9.64$	$10+43.20=53.20$	4.43×10^{-3}	2.17
	$q=10$	$1+4.55=5.55$	$10+45.50=55.50$	4.40×10^{-3}	1.25
	$q=15$	$1+3.70=4.70$	$10+55.50=65.50$	4.43×10^{-3}	1.02
	$q=20$	$1+3.45=4.45$	$10+69.00=79.00$	4.45×10^{-3}	0.91

Note: For AK-MCS+U and ALPI, the MC population size is set as 10^6. AK-MCS+U, ALPI and PABQ are performed 20 independent runs. PC-Kriging was performed 50 independent runs. Thus, for those methods, average results are reported for N_{iter}, N_{call}, and $\hat{P}_f$. Besides, COV$[\hat{P}_f]$ is also approximated accordingly.

4.1.2. Results of Case II

The failure probability is quite small (in the order of 10^{-7}) in Case II, and hence some methods, like AK-MCS and ALPI, are not applicable anymore. For this reason, the proposed method is mainly compared with AK-MCMC [46], which is capable of assessing extremely small failure probabilities. As can be seen from Table 2, the proposed method can not only reduce the average number of iterations greatly (especially when q is large, e.g., $q = 20$), but also the total number of calls to the performance function (especially when q is small, e.g., $q = 5$), in comparison to AK-MCMC. Besides, the proposed PABQ method is also able to yield fairly good average results for the failure probability. It is noted that the COVs of the failure probability estimates can be reduced by a more strict parameter setting (i.e., $N_{ibs} = 1 \times 10^6, \epsilon = 8\%$). This case study demonstrates the efficiency and accuracy of the proposed method for such a case with an extremely rare failure event.

Table 2: Reliability results for Example 1 (Case II).

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^9	8.84×10^{-7}	3.36
AK-MCMC		$1 + 134.00 = 135.00$	$12 + 134.00 = 146.00$	8.85×10^{-7}	1.62
Proposed PABQ ($N_{ibs} = 5 \times 10^5, \epsilon = 10\%$)	$q = 5$	$1 + 8.80 = 9.80$	$10 + 44.00 = 54.00$	8.82×10^{-7}	2.14
	$q = 10$	$1 + 5.45 = 6.45$	$10 + 54.50 = 64.50$	8.84×10^{-7}	2.06
	$q = 15$	$1 + 4.75 = 5.75$	$10 + 71.25 = 81.25$	8.83×10^{-7}	1.24
	$q = 20$	$1 + 4.40 = 5.40$	$10 + 88.00 = 98.00$	8.88×10^{-7}	1.24
Proposed PABQ ($N_{ibs} = 1 \times 10^6, \epsilon = 8\%$)	$q = 5$	$1 + 8.80 = 9.80$	$10 + 44.00 = 54.00$	8.80×10^{-7}	1.63
	$q = 10$	$1 + 5.95 = 6.95$	$10 + 59.50 = 69.50$	8.83×10^{-7}	0.89
	$q = 15$	$1 + 4.95 = 5.95$	$10 + 74.25 = 84.25$	8.86×10^{-7}	0.89
	$q = 20$	$1 + 4.80 = 5.80$	$10 + 96.00 = 106.00$	8.86×10^{-7}	0.66

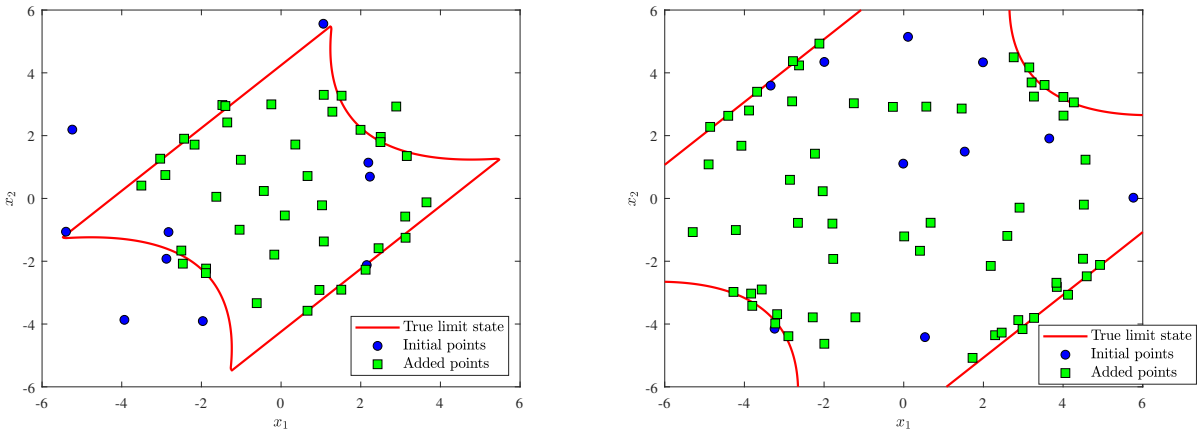
Note: AK-MCMC and PABQ are performed 20 independent runs. Thus, for those methods, average results are reported for N_{iter}, N_{call}, and $\hat{P}_f$. Besides, COV$[\hat{P}_f]$ is also approximated accordingly.

Fig. 4(b) depicts the points selected at two stages of the proposed method ($q = 10$) via an exemplary run, along with the real limit state curve. It is encouraging to see that the added points are relatively sparsely distributed, and most of them are located in the vicinity of true limit state curve.

4.2. Example 2: A nonlinear oscillator

A nonlinear undamped single-degree-of-freedom (SDOF) oscillator subjected to a rectangular pulse load [43] is adopted as the second example, as shown in Fig. 5. The performance function is defined as:

$$Y = g\left(m, c_1, c_2, r, F_1, t_1\right) = 3r - \left| \frac{2F_1}{c_1 + c_2} \sin\left(\frac{t_1}{2} \sqrt{\frac{c_1 + c_2}{m}} \right) \right|, \tag{30}$$



(a) Case I ($q = 10$) (b) Case II ($q = 10$)

Figure 4: Points selected at different stages by the proposed PABQ method for Example 2.

where m, c_1, c_2, r, F_1, t_1 are six random variables, as described in Table 3.

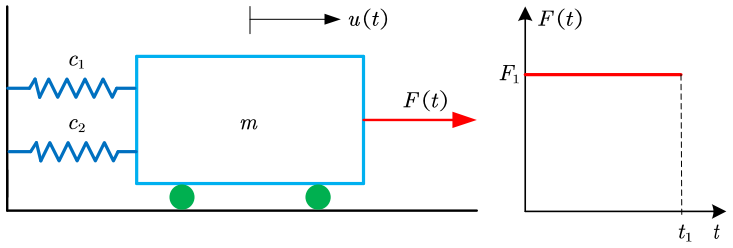


Figure 5: A nonlinear undamped SDOF oscillator subjected to pulse load.

The reference value of the failure probability is 5.17×10^{-6} (with COV being small, i.e., 1.39%), provided by MCS with 10^9 samples. As the failure probability is quite small, the proposed method is only compared to FORM [65], SORM [65] and AK-MCMC [46]. As can be seen from Table 4, the required average number of iterations by the proposed method is less than all the methods being compared, especially for AK-MCMC. This implies the parallel computing advantage of the proposed method. Besides, the proposed method is still more advantageous than FORM, SORM and AK-MCMC in computational efficiency in case of non-parallel computing, since the average number of performance function evaluations can also be reduced a lot (especially when q is small, e.g., $q = 5$). Although $\mathrm{COV}[\hat{P}_f]$ given by the proposed method ($N_{ibs} = 5 \times 10^5, \epsilon = 10\%$) is around 5%, it can still be acceptable in practical applications. If one would like to reduce $\mathrm{COV}[\hat{P}_f]$, one can increase N_{ibs} and decrease ϵ. For example, the last four rows of Table 4 give the results by of PABQ ($N_{ibs} = 1 \times 10^6, \epsilon = 5\%$). It can be seen that $\mathrm{COV}[\hat{P}_f]$ is reduced to about 3% at the expense of increased N_{iter} and N_{call} in some cases ($q = 5, 10, 20$), while still much less than those of FORM, SORM and AK-MCMC.

Table 3: Random variables for Example 2.

Variable	Distribution	Mean	STD
m	Normal	1.0	0.05
c_1	Lognormal	1.0	0.10
c_2	Lognormal	0.1	0.01
r	Normal	0.5	0.05
F_1	Lognormal	0.5	0.10
t_1	Normal	1.0	0.20

4.3. Example 3: A simple bracket model

A simple bracket model that is available in the Partial Differential Equation Toolbox of Matlab is considered as the third example. The schematic diagram of the bracket is shown in Figs. 6(a) and 6(b), and more details of the model can be found in the description in the toolbox. The bracket is fixed at the back face (face 4) and subjected to a distributed load in the negative z-direction in the front face (face 8). It is assumed that the Young's modulus E, Poisson's ratio μ, distributed load q and thickness h of the horizontal plate with hole are characterized as independent random variables, whose statistical information is summarized in Table. 5. The 10-node tetrahedral element is used to discretize the model, as shown in Figs. 6(c) and 6(d). The maximal deflection of the bracket in the z direction is of concern in this example. The limit state function is defined as:

$$Y = G(E, \mu, q, h) = \Delta - \bar{V}(E, \mu, q, h), \tag{31}$$

where Δ is the deterministic threshold, which is specified as $\Delta = 140$ μm; $\bar{V}$ denotes the maximum displacement of the bracket in the z-direction.

We implement several methods to assess the failure probability corresponding to the limit state function defined in Eq. (31). The results are reported in Table 6. FORM does not converge within 100 iterations, so its results are not included. The reference value of the failure probability is taken as the average result of AK-MCMC, i.e., 1.90×10^{-6} (with a COV of 1.15%). It can be seen from Table 6 that the proposed PABQ method can significantly reduce the number of iterations N_{iter} compared to AK-MCMC, while maintaining reasonable accuracy. This indicates that our method could greatly outperform AK-MCMC in terms of computational efficiency when parallel computing is available. One can also notice that the proposed method requires less performance function calls in average than AK-MCMC. Therefore, the proposed method could be still more efficient than AK-MCMC in case that parallel computing is unavailable. The variability of the failure probability estimate given by the proposed method can be reduced to a certain level by setting a large N_{ibs} and a small ϵ.

Table 4: Reliability results for Example 2.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^9	5.17×10^{-6}	1.39
FORM		10	80	5.45×10^{-6}	-
SORM		10	160	5.25×10^{-6}	-
AK-MCMC		$1 + 109.20 = 110.20$	$12 + 109.20 = 121.20$	5.23×10^{-6}	0.69
Proposed PABQ ($N_{ibs} = 5 \times 10^5, \epsilon = 10\%$)	$q = 5$	$1 + 3.15 = 4.15$	$10 + 15.75 = 25.75$	5.19×10^{-6}	5.68
	$q = 10$	$1 + 2.05 = 3.05$	$10 + 20.50 = 30.50$	5.21×10^{-6}	4.30
	$q = 15$	$1 + 1.65 = 2.65$	$10 + 24.75 = 34.75$	5.17×10^{-6}	4.30
	$q = 20$	$1 + 1.70 = 2.70$	$10 + 34.00 = 44.00$	5.21×10^{-6}	4.79
Proposed PABQ ($N_{ibs} = 1 \times 10^6, \epsilon = 5\%$)	$q = 5$	$1 + 4.05 = 5.05$	$10 + 20.25 = 30.25$	5.15×10^{-6}	3.08
	$q = 10$	$1 + 2.40 = 3.40$	$10 + 24.00 = 34.00$	5.15×10^{-6}	2.41
	$q = 15$	$1 + 2.00 = 3.00$	$10 + 30.00 = 40.00$	5.15×10^{-6}	3.53
	$q = 20$	$1 + 1.95 = 2.95$	$10 + 39.00 = 49.00$	5.20×10^{-6}	3.44

Note: AK-MCMC and PABQ are performed 20 independent runs. Thus, for those methods, average results are reported for N_{iter}, N_{call}, and $\hat{P}_f$. Besides, COV$[\hat{P}_f]$ is also approximated accordingly.

Table 5: Random variables for Example 3.

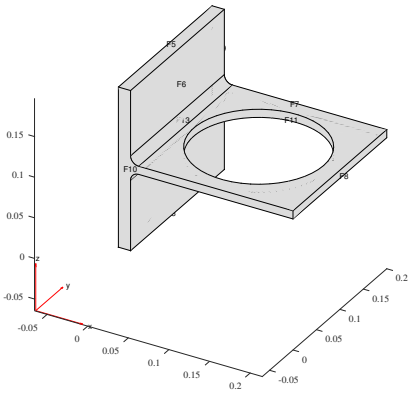
Variable	Distribution	Mean	COV
E (Gpa)	Lognormal	200	0.15
μ	Uniform	0.3	0.10
q (Pa)	Lognormal	10^4	0.20
h (mm)	Lognormal	10	0.10

4.4. Example 4: A 120-bar space truss structure

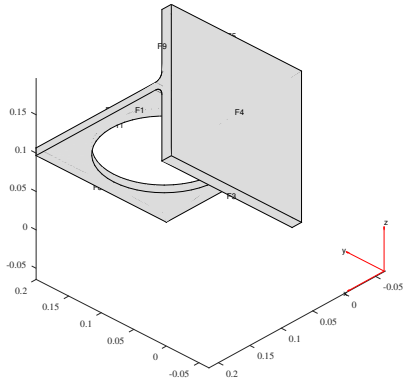
A 120-bar space truss structure [43], as shown in Fig. 7, is investigated in the last example to further demonstrate the proposed method. The structure is modelled as a three-dimensional (3D) finite-element model with 49 nodes and 120 elements in OpenSees. Nodes 0, 1, 4, 7 and 10 withstand concentrated loads along the negative z-axis, denoted as P_0, P_1, P_4, P_7 and P_{10} respectively. All elements are assumed to have the same cross-sectional area A and Young's modulus E. The structure is expected to be in a linear elastic state, so we simply employ linear finite element analysis. The performance function is defined as:

$$Y = g(P_0, P_1, P_4, P_7, A, E) = \Delta - V_{0,z}, \tag{32}$$

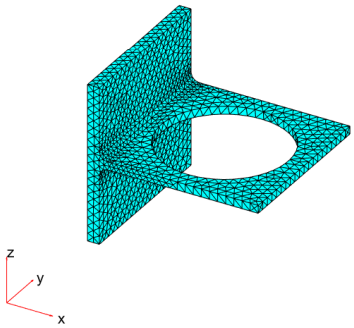
where $V_{0,z}$ denotes the vertical displacement of node 0; and Δ is the threshold, specified as 90 mm. The random variables considered in this examples are summarized in Table 7.



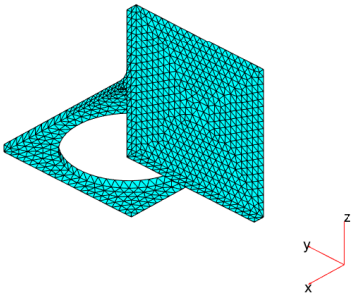
(a) Bracket with face labels (side view)



(b) Bracket with face labels (rear view)



(c) Finite-element mesh (side view)



(d) Finite-element mesh (rear view)

Figure 6: A simple bracket model: Geometry and finite-element mesh.

In this example, several methods, i.e., MCS, FORM [65], SORM [65], AK-MCS+U [37], ALPI [43], AK-MCMC [46] and PABQ, are implemented to assess the failure probability. The results are listed in Table 8. The reference value for the failure probability is 5.08×10^{-4} with COV being 4.44, provied by MCS with 10^6 samples. The results of AK-MCMC are not reported because it fails to converge in multiple trials. FORM only requires 7 iterations and a total number of 65 performance function calls, which, however, results in an inaccurate result. SORM can provide more accurate failure probability estimate than FORM at the expense of 172 calls to the performance function (hence the finite-element model). Compared to AK-MCS+U and ALPI, the proposed PABQ method performs better in terms of N_{call} (especially when q is small, e.g., $q = 5$), and much better in terms of N_{iter} (especially when q is large, e.g., $q = 20$). This implies that PABQ can be much more efficient than AK-MCS+U and ALPI in cases of both parallel and non-parallel computing. Besides, the proposed method still has a acceptable accuracy, as indicated by $\hat{P}_f$ and COV$[\hat{P}_f]$. As shown in the last four rows of Table 8, COV$[\hat{P}_f]$ can be further reduced by increasing N_{ibs} and decreasing ϵ at the

Table 6: Reliability results for Example 3.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
FORM		-	-	-	-
AK-MCMC		$1 + 44.60 = 45.60$	$12 + 44.60 = 56.60$	1.90×10^{-6}	1.15
Proposed PABQ ($N_{ibs} = 5 \times 10^5, \epsilon = 10\%$)	$q = 2$	$1 + 1.85 = 2.85$	$10 + 3.70 = 13.70$	1.93×10^{-6}	4.99
	$q = 3$	$1 + 1.45 = 2.45$	$10 + 4.35 = 14.35$	1.88×10^{-6}	6.19
	$q = 4$	$1 + 1.40 = 2.40$	$10 + 5.60 = 15.60$	1.93×10^{-6}	5.89
	$q = 5$	$1 + 1.35 = 2.35$	$10 + 6.75 = 16.75$	1.91×10^{-6}	8.99
Proposed PABQ ($N_{ibs} = 1 \times 10^6, \epsilon = 5\%$)	$q = 2$	$1 + 2.45 = 3.45$	$10 + 4.90 = 14.90$	1.91×10^{-6}	3.74
	$q = 3$	$1 + 1.95 = 2.95$	$10 + 5.85 = 15.85$	1.90×10^{-6}	2.19
	$q = 4$	$1 + 1.65 = 2.65$	$10 + 6.60 = 16.60$	1.89×10^{-6}	3.70
	$q = 5$	$1 + 1.55 = 2.55$	$10 + 7.75 = 17.75$	1.93×10^{-6}	3.16

Note: AK-MCMC and PABQ are performed 20 independent runs. Thus, for those methods, average results are reported for N_{iter}, N_{call}, and $\hat{P}_f$. Besides, COV$[\hat{P}_f]$ is also approximated accordingly.

cost of slightly increased N_{iter} and N_{call}.

Table 7: Random variables for Example 4.

Variable	Distribution	Mean	COV
P_0	Lognormal	500 kN	0.20
P_1	Lognormal	200 kN	0.20
P_4	Lognormal	200 kN	0.20
P_7	Lognormal	200 kN	0.20
P_{10}	Lognormal	200 kN	0.20
A	Normal	2000 mm^2	0.15
E	Normal	2.00×10^5 MPa	0.15

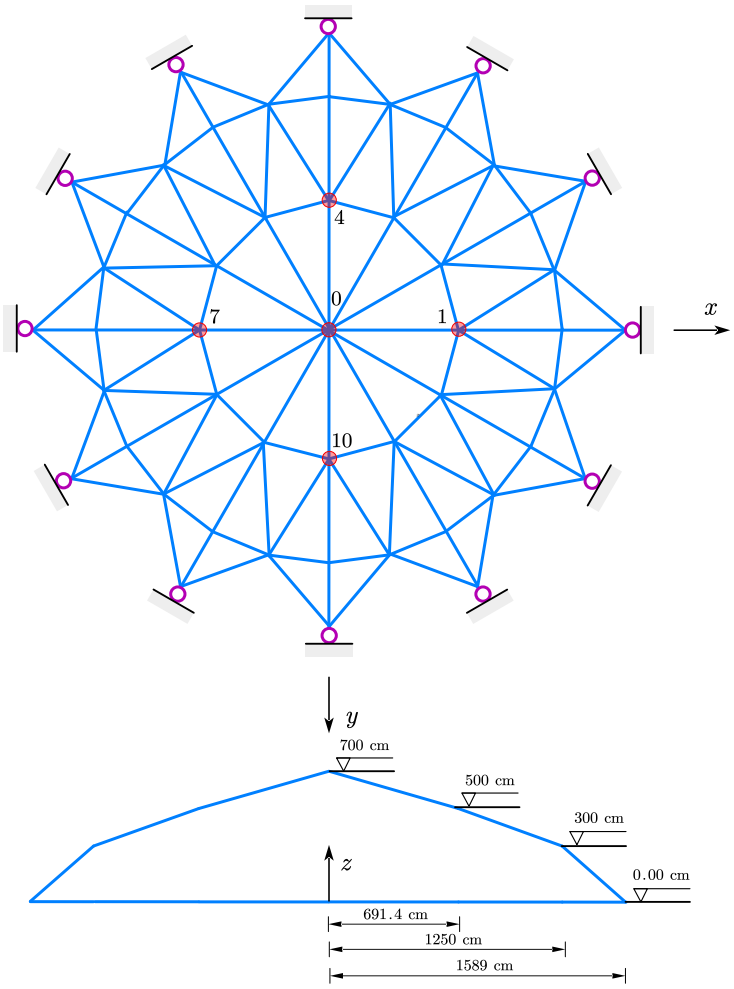


Figure 7: A 120-bar space truss structure.

Table 8: Reliability results for Example 4.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^6	5.08×10^{-4}	4.44
FORM		7	65	3.16×10^{-4}	-
SORM		7	172	5.23×10^{-4}	-
AK-MCS+U		$1 + 60.75 = 61.75$	$12 + 60.75 = 72.75$	5.16×10^{-4}	4.84
ALPI		$1 + 47.45 = 48.45$	$12 + 47.45 = 59.45$	5.10×10^{-4}	3.54
AK-MCMC		-	-	-	-
Proposed PABQ	$q = 5$	$1 + 5.90 = 6.90$	$10 + 29.50 = 39.50$	4.93×10^{-4}	4.74
	$q = 10$	$1 + 3.80 = 4.80$	$10 + 38.00 = 48.00$	4.98×10^{-4}	3.31
	$q = 15$	$1 + 2.65 = 3.65$	$10 + 39.75 = 49.75$	4.99×10^{-4}	4.68
	$q = 20$	$1 + 2.40 = 3.40$	$10 + 48.00 = 58.00$	4.98×10^{-4}	6.22
Proposed PABQ ($N_{ibs} = 1 \times 10^6, \epsilon = 5\%$)	$q = 5$	$1 + 8.65 = 9.65$	$10 + 43.25 = 53.25$	5.04×10^{-4}	3.41
	$q = 10$	$1 + 4.80 = 5.80$	$10 + 48.00 = 58.00$	5.06×10^{-4}	2.40
	$q = 15$	$1 + 3.70 = 4.70$	$10 + 55.50 = 65.50$	5.07×10^{-4}	2.15
	$q = 20$	$1 + 2.90 = 3.90$	$10 + 58.00 = 68.00$	5.02×10^{-4}	4.27

Note: For AK-MCS+U and ALPI, the MC population size is set as 10^6. AK-MCS+U, ALPI and PABQ are performed 20 independent runs. Thus, for those methods, average results are reported for N_{iter}, N_{call}, and $\hat{P}_f$. Besides, COV$[\hat{P}_f]$ is also approximated accordingly.

5. Conclusions

This paper presents a 'Parallel Adaptive Bayesian Quadrature' (PABQ) method for rare failure event estimation. As it is rooted in ALPI, PABQ offers an alternative framework to the quantification, propagation and reduction of numerical uncertainty for assessing failure probabilities. Besides, compared to ALPI, two important improvements are made in PABQ to enable the use of ever-increasing parallel computing facilities and enhance the capability of assessing small failure probabilities. The parallelism of PABQ is achieved by developing a multi-point selection strategy, while the capableness for rare failure event estimation is realized by proposing an importance ball sampling technique. The performance of the proposed method is illustrated by means of four numerical examples. In most studied cases, it is found that PABQ can not only significantly reduce the average number of iterations (especially when q is large), but also lower the average total number of performance function calls (especially when q is small) compared to several selected existing methods. This indicates the computational efficiency advantage of PABQ in both parallel and non-parallel computing. In addition, PABQ is able to produce accurate estimates for small failure probabilities (e.g., in the order of 10^{-7}).

The proposed method, in its current form, is not applicable to high-dimensional and/or strongly nonlinear problems. The former, one one hand, is due to the challenges of implementing GP models in high dimensions. On the other hand, IBS should not lead to significant improvement for a high-dimensional case. The latter is caused by the fact that the GP model is typically suitable for modelling smooth or moderately nonlinear functions. These drawbacks will be addressed in future work.

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). Pengfei Wei is grateful to the support from the National Natural Science Foundation of China (grant no. 51905430 and 72171194). Marcos Valdebenito acknowledges the support by ANID (National Agency for Research and Development, Chile) under its program FONDECYT, grant number 1180271. Chao Dang, Pengfei Wei and Michael Beer also would like to appreciate the support of Sino-German Mobility Program under grant number M-0175.

Appendix A. Numerical algorithm of the active learning probabilistic integration

The procedure for numerical implementation of the ALPI method includes the following steps:

Step A.1: Generate a Monte Carlo population

Generate a MC population comprising N_{mc} samples according to $f_{\boldsymbol{X}}(\boldsymbol{x})$, denoted by $\overline{\boldsymbol{\mathcal{X}}} = \left\{\overline{\boldsymbol{x}}^{(i)}\right\}_{i=1}^{N_{mc}}$. This population has two functions: (1) It serves as a candidate sample pool among which the next best point is identified to evaluate on the g-function; and (2) It is used to evaluate the posterior mean and UPV of the failure probability (Eqs. (13) and (15)).

Step A.2: Get initial observations

Randomly select N_0 (e.g., 12) samples among $\overline{\boldsymbol{\mathcal{X}}}$, denoted by $\boldsymbol{\mathcal{X}}$. Those points are then evaluated on the g-function to get N_0 observations, denoted by $\boldsymbol{\mathcal{Y}}$. As such, an initial dataset can be constructed, i.e., $\mathscr{D} = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$. Let $n = N_0$.

Step A.3: Infer the posterior failure probability

The prior mean and variance functions of $\hat{g}_0 \sim \mathcal{GP}(m_{\hat{g}_0}(\boldsymbol{x}), k_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}'))$ are assumed to be a constant and the squared exponential kernel in this study, respectively. Based on $\mathscr{D}$, a posterior GP $\hat{g}_n \sim \mathcal{GP}(m_{\hat{g}_n}(\boldsymbol{x}), k_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}'))$ for the g-function can be obtained. This step mainly consists of tuning the hyper-parameters via maximum likelihood estimation. For convenience, one can use the *fitrgp* function in Statistics and Machine Learning Toolbox of Matlab. Afterwards, the posterior mean of failure probability can be estimated by:

$$\tilde{m}_{\hat{P}_{f,n}} = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \boldsymbol{\Phi}\left(-\frac{m_{\hat{g}_n}(\overline{\boldsymbol{x}}^{(i)})}{\sigma_{\hat{g}_n}\left(\overline{\boldsymbol{x}}^{(i)}\right)}\right), \tag{A.1}$$

and the upper-bound of posterior standard deviation (UPSTD):

$$\tilde{\bar{\sigma}}_{\hat{P}_{f,n}} = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \sqrt{\boldsymbol{\Phi}\left(-\frac{m_{\hat{g}_n}(\overline{\boldsymbol{x}}^{(i)})}{\sigma_{\hat{g}_n}\left(\overline{\boldsymbol{x}}^{(i)}\right)}\right) \boldsymbol{\Phi}\left(\frac{m_{\hat{g}_n}(\overline{\boldsymbol{x}}^{(i)})}{\sigma_{\hat{g}_n}\left(\overline{\boldsymbol{x}}^{(i)}\right)}\right)}. \tag{A.2}$$

Step A.4: Check the stopping criterion

Only if the posterior failure probability processes a sufficiently low level of epistemic uncertainty, its mean can be used to predict the failure probability. To this end, we propose to examine the estimated upper bound of posterior COV of the failure probability as described next. If $\frac{\tilde{\bar{\sigma}}_{\hat{P}_{f,n}}}{\tilde{m}_{\hat{P}_{f,n}}} < \epsilon$ is satisfied, go to **Step A.6**; Else, go to **Step A.5**. Here ϵ is a user-specified threshold, which takes the value of 0.02 in all numerical examples.

Step A.5: Enrich the previous dataset

At this stage, the best next point to evaluate on the g-function should be identified by a learning function. By exploring the structure of UPV of failure probability (Eq. (15)), the so-called upper-bound posterior variance contribution (UPVC) function is introduced [43]:

$$\mathrm{UPVC}(\boldsymbol{x}) = \sqrt{\boldsymbol{\Phi}\left(-\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \boldsymbol{\Phi}\left(\frac{m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)} \times f_{\boldsymbol{X}}(\boldsymbol{x}), \tag{A.3}$$

where $\overline{\sigma}^2_{\hat{P}_{f,n}} = \left[\int_{\mathcal{X}} \mathrm{UPVC}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right]^2$ holds. The best next point $\boldsymbol{x}^\star$ is selected by:

$$\boldsymbol{x}^\star = \arg\max_{\overline{\boldsymbol{x}} \in \overline{\mathcal{X}}} \mathrm{UPVC}(\boldsymbol{x}). \tag{A.4}$$

The g-function is then evaluated at the point $\boldsymbol{x}^\star$, i.e., $y^\star = g(\boldsymbol{x}^\star)$. The dataset $\mathscr{D}$ is enriched by $\mathscr{D} = \mathscr{D} \cup (\boldsymbol{x}^\star, y^\star)$. Let $n = n + 1$, and go to **Step A.3**.

Step A.6: End the algorithm

Return $\tilde{m}_{\hat{P}_{f,n}}$ as the estimated failure probability and end the algorithm.

References

[1] 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.

[2] A. Harbitz, Efficient and accurate probability of failure calculation by the use of importance sampling technique, in: Proceedings of the 4-th International Conference on Applications of Statistics and Probability in Soil and Structural Engineering, Vol. 4, 1983, pp. 825–836.

[3] A. Tabandeh, G. Jia, P. Gardoni, A review and assessment of importance sampling methods for reliability analysis, Structural Safety 97 (2022) 102216.

[4] A. M. Hasofer, N. C. Lind, Exact and invariant second-moment code format, Journal of the Engineering Mechanics Division 100 (1) (1974) 111–121.

[5] R. Rackwitz, B. Flessler, Structural reliability under combined random load sequences, Computers & Structures 9 (5) (1978) 489–494.

[6] K. Breitung, Asymptotic approximations for multinormal integrals, Journal of Engineering Mechanics 110 (3) (1984) 357–366.

[7] Z. Hu, R. Mansour, M. Olsson, X. Du, Second-order reliability methods: a review and comparative study, Structural and Multidisciplinary Optimization (2021) 1–31.

[8] Y. Zhao, Z. Lu, Structural reliability: approaches from perspectives of statistical moments, John Wiley & Sons Ltd Chichester, 2021.

[9] J. Xu, C. Dang, A new bivariate dimension reduction method for efficient structural reliability analysis, Mechanical Systems and Signal Processing 115 (2019) 281–300.

[10] T. Zhou, Y. Peng, Adaptive bayesian quadrature based statistical moments estimation for structural reliability analysis, Reliability Engineering & System Safety 198 (2020) 106902.

[11] X. Zhang, M. D. Pandey, Structural reliability analysis based on the concepts of entropy, fractional moment and dimensional reduction method, Structural Safety 43 (2013) 28–40.

[12] J. Xu, F. Kong, Adaptive scaled unscented transformation for highly efficient structural reliability analysis by maximum entropy method, Structural Safety 76 (2019) 123–134.

[13] C. Dang, J. Xu, A mixture distribution with fractional moments for efficient seismic reliability analysis of nonlinear structures, Engineering Structures 208 (2020) 109912.

[14] J. Xu, C. Dang, A novel fractional moments-based maximum entropy method for high-dimensional reliability analysis, Applied Mathematical Modelling 75 (2019) 749–768.

[15] C. Dang, J. Xu, Unified reliability assessment for problems with low-to high-dimensional random inputs using the Laplace transform and a mixture distribution, Reliability Engineering & System Safety 204 (2020) 107124.

[16] C. Dang, P. Wei, M. Beer, An approach to evaluation of EVD and small failure probabilities of uncertain nonlinear structures under stochastic seismic excitations, Mechanical Systems and Signal Processing 152 (2021) 107468.

[17] J.-B. Chen, J. Li, Dynamic response and reliability analysis of non-linear stochastic structures, Probabilistic Engineering Mechanics 20 (1) (2005) 33–44.

[18] J. Li, J.-B. Chen, Dynamic response and reliability analysis of structures with uncertain parameters, International Journal for Numerical Methods in Engineering 62 (2) (2005) 289–315.

[19] J.-B. Chen, J. Li, The extreme value distribution and dynamic reliability analysis of nonlinear structures with uncertain parameters, Structural Safety 29 (2) (2007) 77–93.

[20] J. Li, J.-b. Chen, W.-l. Fan, The equivalent extreme-value event and evaluation of the structural system reliability, Structural Safety 29 (2) (2007) 112–131.

[21] X. Li, G. Chen, H. Cui, D. Yang, Direct probability integral method for static and dynamic reliability analysis of structures with complicated performance functions, Computer Methods in Applied Mechanics and Engineering 374 (2021) 113583.

[22] G. Chen, D. Yang, A unified analysis framework of static and dynamic structural reliabilities based on direct probability integral method, Mechanical Systems and Signal Processing 158 (2021) 107783.

[23] L. Li, G. Chen, M. Fang, D. Yang, Reliability analysis of structures with multimodal distributions based on direct probability integral method, Reliability Engineering & System Safety 215 (2021) 107885.

[24] G. Chen, D. Yang, Y. Liu, H. Guo, System reliability analyses of static and dynamic structures via direct probability integral method, Computer Methods in Applied Mechanics and Engineering 388 (2022) 114262.

[25] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety 7 (1) (1990) 57–66.

[26] S.-C. Kang, H.-M. Koh, J. F. Choo, An efficient response surface method using moving least squares approximation for structural reliability analysis, Probabilistic Engineering Mechanics 25 (4) (2010) 365–371.

[27] A. Hadidi, B. F. Azar, A. Rafiee, Efficient response surface method for high-dimensional structural reliability analysis, Structural Safety 68 (2017) 15–27.

[28] C. M. Rocco, J. A. Moreno, Fast monte carlo reliability evaluation using support vector machine, Reliability Engineering & System Safety 76 (3) (2002) 237–243.

[29] K. Cheng, Z. Lu, Adaptive bayesian support vector regression model for structural reliability analysis, Reliability Engineering & System Safety 206 (2021) 107286.

[30] J. Wang, C. Li, G. Xu, Y. Li, A. Kareem, Efficient structural reliability analysis based on adaptive bayesian support vector regression, Computer Methods in Applied Mechanics and Engineering 387 (2021) 114172.

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

[32] B. Bhattacharyya, Structural reliability analysis by a bayesian sparse polynomial chaos expansion, Structural Safety 90 (2021) 102074.

[33] I. Kaymaz, Application of kriging method to structural reliability problems, Structural Safety 27 (2) (2005) 133–151.

[34] V. Dubourg, B. Sudret, F. Deheeger, Metamodel-based importance sampling for structural reliability analysis, Probabilistic Engineering Mechanics 33 (2013) 47–57.

[35] I. Depina, T. M. H. Le, G. Fenton, G. Eiksund, Reliability analysis with metamodel line sampling, Structural Safety 60 (2016) 1–15.

[36] 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.

[37] 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.

[38] X. Yang, Y. Liu, Y. Gao, Y. Zhang, Z. Gao, An active learning kriging model for hybrid reliability analysis with both random and interval variables, Structural and Multidisciplinary Optimization 51 (5) (2015) 1003–1016.

[39] Z. Lv, Z. Lu, P. Wang, A new learning function for kriging and its applications to solve reliability problems in engineering, Computers & Mathematics with Applications 70 (5) (2015) 1182–1197.

[40] Z. Sun, J. Wang, R. Li, C. Tong, Lif: A new kriging based learning function and its application to structural reliability analysis, Reliability Engineering & System Safety 157 (2017) 152–165.

[41] X. Zhang, L. Wang, J. D. Sørensen, REIF: a novel active-learning function toward adaptive kriging surrogate models for structural reliability analysis, Reliability Engineering & System Safety 185 (2019) 440–454.

[42] Y. Shi, Z. Lu, R. He, Y. Zhou, S. Chen, A novel learning function based on kriging for reliability analysis, Reliability Engineering & System Safety 198 (2020) 106857.

[43] 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.

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

[45] X. Huang, J. Chen, H. Zhu, Assessing small failure probabilities by AK–SS: An active learning method combining kriging and subset simulation, Structural Safety 59 (2016) 86–95.

[46] P. Wei, C. Tang, Y. Yang, Structural reliability and reliability sensitivity analysis of extremely rare failure events by combining sampling and surrogate model methods, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 233 (6) (2019) 943–957.

[47] T. Zhou, Y. Peng, Structural reliability analysis via dimension reduction, adaptive sampling, and monte carlo simulation, Structural and Multidisciplinary Optimization 62 (5) (2020) 2629–2651.

[48] T. Zhou, Y. Peng, Active learning and active subspace enhancement for pdem-based high-dimensional reliability analysis, Structural Safety 88 (2021) 102026.

[49] Y. Liu, L. Li, S. Zhao, S. Song, A global surrogate model technique based on principal component analysis and kriging for uncertainty propagation of dynamic systems, Reliability Engineering & System Safety 207 (2021) 107365.

[50] Y. Peng, T. Zhou, J. Li, Surrogate modeling immersed probability density evolution method for structural reliability analysis in high dimensions, Mechanical Systems and Signal Processing 152 (2021) 107366.

[51] T. Zhou, Y. Peng, Kernel principal component analysis-based gaussian process regression modelling for high-dimensional reliability analysis, Computers & Structures 241 (2020) 106358.

[52] M. M. Zuniga, A. Murangira, T. Perdrizet, Structural reliability assessment through surrogate based importance sampling with dimension reduction, Reliability Engineering & System Safety 207 (2021) 107289.

[53] J. Yin, X. Du, Active learning with generalized sliced inverse regression for high-dimensional reliability analysis, Structural Safety 94 (2022) 102151.

[54] R. Teixeira, M. Nogal, A. O'Connor, B. Martinez-Pastor, Reliability assessment with density scanned adaptive kriging, Reliability Engineering & System Safety 199 (2020) 106908.

[55] R. Li, X. Liang, Q. Peng, A selection strategy for kriging based design of experiments by spectral clustering and learning function, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering 7 (2) (2021) 020902.

[56] 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.

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

[58] A. O'Hagan, Bayes–hermite quadrature, Journal of Statistical Planning and Inference 29 (3) (1991) 245–260.

[59] C. E. Rasmussen, Z. Ghahramani, Bayesian monte carlo, Advances in Neural Information Pprocessing Systems (2003) 505–512.

[60] P. Wei, X. Zhang, M. Beer, Adaptive experiment design for probabilistic integration, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113035.

[61] V. Lalchand, C. E. Rasmussen, Approximate inference for fully bayesian gaussian process regression, in: Symposium on Advances in Approximate Bayesian Inference, PMLR, 2020, pp. 1–12.

[62] A. R. Voelker, J. Gosmann, T. C. Stewart, Efficiently sampling vectors and coordinates from the n-sphere and n-ball, Centre for Theoretical Neuroscience-Technical Report (2017).

[63] J. Li, J. Chen, Stochastic dynamics of structures, John Wiley & Sons, 2009.

[64] D. J. MacKay, D. J. Mac Kay, Information theory, inference and learning algorithms, Cambridge university press, 2003.

[65] 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).

[66] R. Schöbi, B. Sudret, S. Marelli, Rare event estimation using polynomial-chaos kriging, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 3 (2) (2017) D4016002.

[67] A. Borri, E. Speranzini, Structural reliability analysis using a standard deterministic finite element code, Structural Safety 19 (4) (1997) 361–382.

[68] I. Papaioannou, S. Geyer, D. Straub, Improved cross entropy-based importance sampling with a flexible mixture model, Reliability Engineering & System Safety 191 (2019) 106564.