Structural reliability analysis: A Bayesian perspective

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

Numerical methods play a dominant role in structural reliability analysis, and the goal has long been to produce a failure probability estimate with a desired level of accuracy using a minimum number of performance function evaluations. In the present study, we attempt to offer a Bayesian perspective on the failure probability integral estimation, as opposed to the classical frequentist perspective. For this purpose, a Bayesian Failure Probability Inference (BFPI) framework is first developed, which allows to quantify, propagate and reduce numerical uncertainty behind the failure probability due to discretization error. Especially, the posterior variance of the failure probability is derived in a semi-analytical form, and the Gaussianity of the posterior failure probability distribution is investigated numerically. Then, a Parallel Adaptive-Bayesian Failure Probability Learning (PA-BFPL) method is proposed within the Bayesian framework. In the PA-BFPL method, a variance-amplified importance sampling technique is presented to evaluate the posterior mean and variance of the failure probability, and an adaptive parallel active learning strategy is proposed to identify multiple updating points. Thus, a novel advantage of PA-BFPL is that both prior knowledge and parallel computing can be used to make inference about the failure probability. Four numerical examples are investigated, indicating the potential benefits by advocating a Bayesian approach to failure probability estimation.

Keywords: Failure probability, Bayesian inference, Gaussian process, Numerical uncertainty, Parallel computing

---

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

1. Introduction

A fundamental problem in structural reliability analysis is to assess the likelihood that a structure attains an unsatisfactory performance in the presence of uncertainties. Within a probabilistic framework, the primary objective is to compute the so-called failure probability P_f, defined by the following multifold integral:

$$P_f = \text{Prob}\left[g(\boldsymbol{X}) \leq 0\right] = \int_{\mathcal{X}} I(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\text{Prob}\left[\cdot\right]$ denotes the probability operator; $\boldsymbol{X} = [X_1, X_2, \cdots, X_d] \in \mathcal{X} \subseteq \mathbb{R}^d$ is a vector of d random variables with known joint probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x})$; $Y = g(\boldsymbol{X}) : \mathbb{R}^d \to \mathbb{R}$ is the performance function (or limit state function) with $y = g(\boldsymbol{x}) \leq 0$ indicating a failure state and a safe state otherwise; $I(\boldsymbol{x})$ is the failure indicator function such that:

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

Except for some special cases, it is impossible to derive the analytical solution to the failure probability (defined by Eq. (1)). Besides, the g-function in practical applications is typically dependent on a simulation model (e.g., a finite element model) so that each evaluation can be computationally demanding. Therefore, numerical methods that minimize the number of g-function evaluations are highly desirable to approximate the failure probability. Even though various methods following different paradigms have been developed over the past several decades (e.g., as summarized in [1]), it seems that they never reach the end of being efficient while accurate and generally applicable. The present paper is also concerned with developing a new reliability analysis method, but putting more emphasis on how to interpret the problem of failure probability estimation.

In fact, the problem of evaluating the failure probability integral (Eq. (1)) can be treated as a statistical problem, though it does not mean that all methods must follow this perspective. Specifically, the failure probability P_f is an unknown quantity of interest, about which we wish to make inference using a set of g-function observations (equivalently, I-function observations), say $g(\boldsymbol{x}^{(1)}), g(\boldsymbol{x}^{(2)}), \cdots, g(\boldsymbol{x}^{(n)})$. Further, a statistical inference rule approximates P_f as a function of those observations.

In the classical frequentist viewpoint, the sample $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}$ might be supposed to draw at random from a population distributed according to $f_{\boldsymbol{X}}(\boldsymbol{x})$. Taking the Monte Carlo simulation (MCS)

method as an example, the MCS estimator for the failure probability is given by the sample mean:

$$\hat{P}_f^{MCS} = \frac{1}{n} \sum_{i=1}^{n} I(\boldsymbol{x}^{(i)}). \tag{3}$$

The law of large numbers implies that $\hat{P}_f^{MCS}$ converges to P_f with probability 1 as $n \to \infty$. The estimator is viewed as a random variable since $\boldsymbol{x}^{(i)}$ is random. Besides, by the central limit theorem, $\hat{P}_f^{MCS}$ asymptotically follows a normal distribution for a large n. In practical applications, one can only afford a finite sample size to approximate the failure probability. Hence, the uncertainty associated with $\hat{P}_f^{MCS}$ due to the sampling variability may not be neglected. Such uncertainty can be measured by the variance of the estimator [2]:

$$\mathbb{V}\left[\hat{P}_f^{MCS}\right] = \frac{\hat{P}_f^{MCS}(1 - \hat{P}_f^{MCS})}{n}, \tag{4}$$

where $\mathbb{V}[\cdot]$ denotes the variance operator. Despite its conceptual and algorithmic simplicity, the MCS method is often criticized by many authors for its unreasonable effectiveness and theoretical unsoundness [3, 4]. In addition, some variants of the MCS method, e.g., subset simulation [5, 6], importance sampling [7, 8, 9, 10], have been developed and are able to offer improved efficiency. These methods, however, can still be regarded as more advanced frequentist approaches, and hence may be subject to the same criticism as MCS.

In contrast to the classical frequentist perspective, we seek to interpret the problem of failure probability integral estimation as a Bayesian inference problem. For this context, a central role is played by numerical integration (also known as quadrature) that is widely encountered in scientific computing. The study of numerical integration from a point of view of Bayesian dates back to at least the work of Diaconis [11] and has led to the commonly known Bayesian quadrature, Bayesian cubature or probabilistic integration [12, 13, 14, 15]. In such methods, our uncertainty about the true integral value resulted from a limited number of integrand observations (i.e., discretization error) is regarded as a kind of epistemic uncertainty, which can be modelled following a Bayesian approach. The Bayesian approach to numerical integration has demonstrated many promising advantages with respect to the classical approach (e.g., see [11, 16]). However, only a few studies have investigated the Bayesian approach to failure probability estimation, which requires a slightly different treatment compared to a common quadrature problem. Loosely speaking, the popular active learning reliability methods [17, 18], e.g., efficient global reliability analysis [19] and AK-MCS [20], have almost reached the idea of being Bayesian. That is, the surrogate models (e.g., Kriging) used in

those methods allow a Bayesian interpretation. In spite of that, the existing methods do not count as fully Bayesian in the strict sense because they provide no probabilistic uncertainty measure over the failure probability. A truly Bayesian interpretation was, to the best of our knowledge, first clearly reported in the work [21], where the Bayesian Monte Carlo method developed in [13] was applied. However, it is challenging to directly place a Gaussian process (GP) prior over the failure indicator function with a large discontinuity. The first author and his co-workers continued the idea of re-interpreting the failure probability integral estimation with Bayesian inference in a recent work [22], and then it was further improved in [1]. In [22], the posterior mean and an upper-bound of the posterior variance of the failure probability were derived, given that a GP prior was assigned to the performance function. Nevertheless, the posterior variance and posterior distribution of the failure probability are still not available, which are undoubtedly of interest and importance in a Bayesian framework.

This paper aims to present a Bayesian perspective on failure probability estimation, complementing the work in [22, 1]. The main contributions of this work are summarized as follows. First, to the best of the authors' knowledge, a complete and principled Bayesian framework for failure probability estimation is developed for the first time. The framework is termed 'Bayesian Failure Probability Inference' (BFPI), in which the posterior variance of the failure probability is derived in a semi-analytical form. Besides, the posterior distribution of the failure probability is also empirically investigated by several numerical examples. Second, we illustrate how the BFPI framework can be used to make inference about the failure probability in an adaptive scheme. The resulting method is called 'Parallel Adaptive-Bayesian Failure Probability Learning' (PA-BFPL). In the PA-BFPL method, a variance-amplified importance sampling (VAIS) method is proposed to approximate the posterior mean and variance of the failure probability and an adaptive parallel learning strategy based on the concepts of expected misclassification probability contribution (EMPC) and k-means clustering is presented to enable multipoint selection (hence parallel distributed processing). In addition, we also suggest a new stopping criterion in order to achieve a desired level of accuracy for the failure probability estimate.

The rest of this paper is organized as follows. The proposed BFPI framework is introduced in Section 2. Section 3 presents the proposed PA-BFPL method. Four numerical examples are investigated in Section 4 to demonstrate the proposed method. The Gaussianity of the posterior failure probability is numerically studied in Section 5. The paper is closed with some concluding remarks in Section 6.

4

2. Bayesian failure probability inference

In this section, the problem of failure probability estimation is interpreted as a Bayesian inference problem, leading to a framework of Bayesian failure probability inference (BFPI). As shown in Fig. 1, the proposed BFPI framework begins with a prior distribution over the g-function. Conditional on the observations that arise from evaluating the g-function at some points, we arrive at a posterior distribution over g. This in turn implies a posterior distribution over the failure indicator function I, as well as the failure probability P_f.
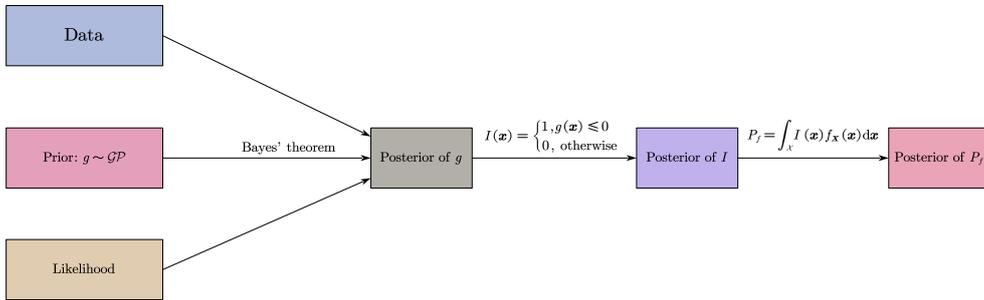
Figure 1: A schematic illustration of the proposed BFPI framework.

2.1. Prior distributions

A convenient way of putting a prior over the g-function is through GP. A GP can be viewed as a collection of random variables indexed by time or space, any finite number of which have a multivariate Gaussian distribution.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, where Ω is a sample space, $\mathcal{F}$ is a set of events equipped with σ-algebra and $\mathbb{P}$ is a probability measure. A GP can be defined as $Z(\varpi, \boldsymbol{x})$: $\Omega \times \mathcal{X} \to \mathbb{R}$. For a fixed location $\boldsymbol{x} \in \mathcal{X}$, $Z(\varpi, \cdot)$ is Gaussian. Conversely, for every fixed elementary event $\varpi \in \Omega$, $Z(\cdot, \boldsymbol{x})$ is a realization of the GP. As a GP can be completely characterized by its mean and covariance functions, our prior assumption is thus rewritten as follows:

$$\hat{g}_0(\varpi, \boldsymbol{x}) \sim \mathcal{GP}(m_{\hat{g}_0}(\boldsymbol{x}), c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}')), \tag{5}$$

where $\hat{g}_0$ denotes the prior distribution of g before any observations are obtained; the prior mean function $m_{\hat{g}_0}(\boldsymbol{x}) : \mathcal{X} \to \mathcal{Z}$ and prior covariance function $c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \to \mathcal{Z}$ are respectively defined as:

$$m_{\hat{g}_0}(\boldsymbol{x}) = \mathbb{E}_{\varpi}\left[\hat{g}_0(\varpi, \boldsymbol{x})\right], \tag{6}$$

5

$$c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\varpi}\left[(\hat{g}_0(\varpi, \boldsymbol{x}) - m(\boldsymbol{x}))\left(\hat{g}_0(\varpi, \boldsymbol{x}') - m(\boldsymbol{x}')\right)\right], \tag{7}$$

in which $\mathbb{E}_{\varpi}[\cdot]$ denotes the expectation operation taken over Ω. The prior mean function reflects the general trend of the GP prior, whereas the prior covariance function encodes our key beliefs on the similarity of the g-function between two points. Among many options available in the literature, this study adopts the commonly used constant mean and squared exponential kernel functions:

$$m_{\hat{g}_0}(\boldsymbol{x}) = \beta, \tag{8}$$

$$c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') = s^2 \exp\left[-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}'\right)\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x} - \boldsymbol{x}'\right)^{\mathrm{T}}\right], \tag{9}$$

where β is a constant; s^2 denote the process variance; $\boldsymbol{\Sigma} = \mathrm{diag}(l_1^2, l_2^2, ..., l_d^2)$ is a diagonal matrix, whose i-th diagonal entry is l_i^2 with $l_i > 0$ being the length scale in the i-th dimension. Note that the choice of the prior and covariance function does not affect the generality of our developments, and other forms can also be employed. In Eqs. (8) and (9), there exist $d + 2$ hyper-parameters to be determined (collected in $\boldsymbol{\vartheta} = [\beta, s, l_1, l_2, \cdots, l_d]$) in total.

Remark 1. Corresponding to the GP prior for g, this also implies implicitly prior distributions for the failure indicator function $I(\boldsymbol{x})$, and the failure probability P_f. They are not given here because our main concern is their posterior distributions. However, one still can easily obtain these prior distributions referring to Subsection 2.3.

2.2. Learning the hyper-parameters

Suppose that now we observe the g-function at some locations. Let $\boldsymbol{\mathscr{X}} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{n}$ denote a $n \times d$ matrix containing n design points. The corresponding g-function values at $\boldsymbol{\mathscr{X}}$ are collected in a $n \times 1$ vector $\boldsymbol{\mathscr{Y}} = \left\{y^{(i)}\right\}_{i=1}^{n}$ with $y^{(i)} = g(\boldsymbol{x}^{(i)})$. The hyper-parameters should be learned from the given data $\boldsymbol{\mathscr{D}} = \{\boldsymbol{\mathscr{X}}, \boldsymbol{\mathscr{Y}}\}$, and three approaches are typically considered [23]: (1) maximum likelihood estimation (MLE); (2) maximum a posteriori (MAP) estimation; (3) fully Bayesian approach. In this study, we use the MLE method as follows.

Under the GP prior, the marginal likelihood of $\boldsymbol{\mathscr{Y}}$ is a multivariate normal density:

$$p(\boldsymbol{\mathscr{Y}}|\boldsymbol{\mathscr{X}}, \boldsymbol{\vartheta}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{C}_{\hat{g}_0}|}} \exp\left[-\frac{1}{2}(\boldsymbol{\mathscr{Y}} - \beta)\boldsymbol{C}_{\hat{g}_0}^{-1}(\boldsymbol{\mathscr{Y}} - \beta)^{\mathrm{T}}\right], \tag{10}$$

6

where $\boldsymbol{C}_{\hat{g}_0}$ is a n-by-n covariance matrix, whose (i,j)-th entry is $[\boldsymbol{C}_{\hat{g}_0}]_{ij} = c_{\hat{g}_0}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$; $|\cdot|$ is the determinant operator. The hyper-parameters are tuned by minimizing the negative log marginal likelihood:

$$\hat{\boldsymbol{\vartheta}} = \underset{\boldsymbol{\vartheta}}{\arg\min} -\log\left[p(\boldsymbol{\mathscr{Y}}|\boldsymbol{\mathscr{X}}, \boldsymbol{\vartheta})\right], \tag{11}$$

where

$$\log\left[p(\boldsymbol{\mathscr{Y}}|\boldsymbol{\mathscr{X}}, \boldsymbol{\vartheta})\right] = -\frac{1}{2}\left[(\boldsymbol{\mathscr{Y}} - \beta)\boldsymbol{C}_{\hat{g}_0}^{-1}(\boldsymbol{\mathscr{Y}} - \beta)^{\mathrm{T}} + \log|\boldsymbol{C}_{\hat{g}_0}| + n\log 2\pi\right]. \tag{12}$$

2.3. Posterior distributions

Conditional on the data $\boldsymbol{\mathscr{D}}$, the induced posterior distribution of g is also a GP:

$$\hat{g}_n(\varpi, \boldsymbol{x}) \sim \mathcal{GP}(m_{\hat{g}_n}(\boldsymbol{x}), c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')), \tag{13}$$

where $\hat{g}_n$ denotes the posterior distribution of g given n observations; $m_{\hat{g}_n}(\boldsymbol{x})$ and $c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')$ are the posterior mean and covariance functions respectively, which can be analytically derived as:

$$m_{\hat{g}_n}(\boldsymbol{x}) = m_{\hat{g}_0}(\boldsymbol{x}) + \boldsymbol{c}_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{\mathscr{X}})\boldsymbol{C}_{\hat{g}_0}^{-1}(\boldsymbol{\mathscr{Y}} - \boldsymbol{m}_{\hat{g}_0}(\boldsymbol{\mathscr{X}})), \tag{14}$$

$$c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') = c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{c}_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{\mathscr{X}})\boldsymbol{C}_{\hat{g}_0}^{-1}\boldsymbol{c}_{\hat{g}_0}(\boldsymbol{\mathscr{X}}, \boldsymbol{x}'), \tag{15}$$

where $\boldsymbol{m}_{\hat{g}_0}(\boldsymbol{\mathscr{X}})$ is n-by-1 vector with i-th element being $m_{\hat{g}_0}(\boldsymbol{x}^{(i)})$; $\boldsymbol{c}_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{\mathscr{X}})$ is a 1-by-n covariance vector with i-th element being $c_{\hat{g}_0}(\boldsymbol{x}, \boldsymbol{x}^{(i)})$; $\boldsymbol{c}_{\hat{g}_0}(\boldsymbol{\mathscr{X}}, \boldsymbol{x}')$ is a n-by-1 covariance vector with i-th element being $\boldsymbol{c}_{\hat{g}_0}(\boldsymbol{x}^{(i)}, \boldsymbol{x}')$. It should be pointed out that: (1) For any $\boldsymbol{x}^{(i)} \in \boldsymbol{\mathscr{X}}$, the posterior GP is an exact predictor. This means that if a prediction is carried out at an observed point $\boldsymbol{x}^{(i)}$, the posterior mean is exactly equal to the corresponding observation (i.e., $m_{\hat{g}_n}(\boldsymbol{x}^{(i)}) = y^{(i)}$) and the posterior variance is equal to zero (i.e., $\sigma_{\hat{g}_n}^2(\boldsymbol{x}^{(i)}) = c_{\hat{g}_n}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(i)}) = 0$). (2) For any $\boldsymbol{x} \notin \boldsymbol{\mathscr{X}}$, the posterior GP at $\boldsymbol{x}$ is Gaussian. In this case, the posterior mean $m_{\hat{g}_n}(\boldsymbol{x})$ is a natural estimate of the g-function value, whereas the posterior variance $\sigma_{\hat{g}_n}^2(\boldsymbol{x}) = c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x})$ can measure our uncertainty of the estimate.

The posterior distribution of the failure indicator function I has the following relationship with $\hat{g}_n$:

$$\hat{I}_n(\varpi, \boldsymbol{x}) = \begin{cases} 1, \hat{g}_n(\varpi, \boldsymbol{x}) \leqslant 0 \\ 0, \text{otherwise} \end{cases}, \tag{16}$$

where $\hat{I}_n$ denote the posterior distribution of I conditional on $\mathscr{D}$. Based on Eqs. (13) and (16), the induced posterior distribution $\hat{I}_n$ should follow a generalized Bernoulli process [1] (GBP):

$$\hat{I}_n(\varpi, \boldsymbol{x}) \sim \mathcal{GBP}(m_{\hat{I}_n}(\boldsymbol{x}), c_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}')), \tag{17}$$

where $m_{\hat{I}_n}(\boldsymbol{x})$ and $c_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}')$ are the posterior mean and covariance functions, respectively. They can be derived as follows:

$$
\begin{aligned}
m_{\hat{I}_n}(\boldsymbol{x}) &= \mathbb{E}_\varpi \left[\hat{I}_n(\varpi, \boldsymbol{x}) \right] \\
&= \mathbb{P} \left[\hat{g}_n(\varpi, \boldsymbol{x}) \le 0 \right] \\
&= \Phi \left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})} \right),
\end{aligned} \tag{18}
$$

$$
\begin{aligned}
c_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x}') =& \mathbb{E}_\varpi \left[\left(\hat{I}_n(\varpi, \boldsymbol{x}) - m_{\hat{I}_n}(\boldsymbol{x}) \right) \left(\hat{I}_n(\varpi, \boldsymbol{x}') - m_{\hat{I}_n}(\boldsymbol{x}') \right) \right] \\
=& \mathbb{E}_\varpi \left[\hat{I}_n(\varpi, \boldsymbol{x}) \hat{I}_n(\varpi, \boldsymbol{x}') \right] - \mathbb{E}_\varpi \left[\hat{I}_n(\varpi, \boldsymbol{x}) \right] \mathbb{E}_\varpi \left[\hat{I}_n(\varpi, \boldsymbol{x}') \right] \\
=& \mathbb{P} \left[\hat{g}_n(\varpi, \boldsymbol{x}) \le 0, \hat{g}_n(\varpi, \boldsymbol{x}') \le 0 \right] - m_{\hat{I}_n}(\boldsymbol{x}) m_{\hat{I}_n}(\boldsymbol{x}') \\
=& F \left([0\ 0]; \boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}'), \boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') \right) - \Phi \left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})} \right) \Phi \left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}')}{\sigma_{\hat{g}_n}(\boldsymbol{x}')} \right),
\end{aligned} \tag{19}
$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution; F is the joint CDF of a bivariate normal distribution; The terms $\boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')$ and $\boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}')$ are expressed as:

$$\boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') = [m_{\hat{g}_n}(\boldsymbol{x}), m_{\hat{g}_n}(\boldsymbol{x}')] \tag{20}$$

$$\boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') = \begin{bmatrix} c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}) & c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') \\ c_{\hat{g}_n}(\boldsymbol{x}', \boldsymbol{x}) & c_{\hat{g}_n}(\boldsymbol{x}', \boldsymbol{x}') \end{bmatrix} = \begin{bmatrix} \sigma_{\hat{g}_n}^2(\boldsymbol{x}) & c_{\hat{g}_n}(\boldsymbol{x}, \boldsymbol{x}') \\ c_{\hat{g}_n}(\boldsymbol{x}', \boldsymbol{x}) & \sigma_{\hat{g}_n}^2(\boldsymbol{x}') \end{bmatrix}. \tag{21}$$

Note that though no closed form is available for F, there are a number of software packages that evaluate it numerically.

The posterior distribution of the failure probability P_f is defined as:

$$\hat{P}_{f,n}(\varpi) = \int_{\mathcal{X}} \hat{I}_n(\varpi, \boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{22}$$

where $\hat{P}_{f,n}$ denotes the posterior distribution of $P_{f,n}$ conditional on $\mathscr{D}$. Eq. (22) implies that $\hat{P}_f$ is a random variable, whose exact distribution is not known yet. To this end, we investigate empirically the posterior

---

[1]In the conventional way, a Bernoulli process is defined as a finite or infinite sequence of binary random variables that are independent and identical distributed. Here we use 'generalized' to indicate that the possible correlation among the sequence is considered. For more information on this topic, one can refer to, e.g., [24].

failure probability distribution by several numerical examples in Section 5. By applying Fubini's theorem, the posterior mean and variance of P_f can be derived as:

$$
\begin{aligned}
m_{\hat{P}_{f,n}} &= \mathbb{E}_{\varpi}\left[\hat{P}_{f,n}(\varpi)\right] \\
&= \int_{\Omega}\int_{\mathcal{X}}\hat{I}_n(\varpi,\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathbb{P}\left(\mathrm{d}\varpi\right) \\
&= \int_{\mathcal{X}}\int_{\Omega}\hat{I}_n(\varpi,\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathbb{P}\left(\mathrm{d}\varpi\right)\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}}\mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x})\right]f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}}\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x},
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
\sigma^2_{\hat{P}_{f,n}} &= \mathbb{V}_{\varpi}\left[\hat{P}_{f,n}(\varpi)\right] \\
&= \mathbb{E}_{\varpi}\left[\left(\hat{P}_f(\varpi) - \mathbb{E}_{\varpi}\left[\hat{P}_f(\varpi)\right]\right)^2\right] \\
&= \mathbb{E}_{\varpi}\left[\left(\int_{\mathcal{X}}\hat{I}_n(\varpi,\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int_{\mathcal{X}}\mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x})\right]f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2\right] \\
&= \int_{\Omega}\left(\int_{\mathcal{X}}\left(\hat{I}_n(\varpi,\boldsymbol{x}) - \mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x})\right]\right)f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^2\mathbb{P}\left(\mathrm{d}\varpi\right) \\
&= \int_{\Omega}\int_{\mathcal{X}}\int_{\mathcal{X}}\left(\hat{I}_n(\varpi,\boldsymbol{x}) - \mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x})\right]\right)\left(\hat{I}_n(\varpi,\boldsymbol{x}') - \mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x}')\right]\right)f_{\boldsymbol{X}}(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'\mathbb{P}\left(\mathrm{d}\varpi\right) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}}\int_{\Omega}\left(\hat{I}_n(\varpi,\boldsymbol{x}) - \mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x})\right]\right)\left(\hat{I}_n(\varpi,\boldsymbol{x}') - \mathbb{E}_{\varpi}\left[\hat{I}_n(\varpi,\boldsymbol{x}')\right]\right)f_{\boldsymbol{X}}(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x}')\mathbb{P}\left(\mathrm{d}\varpi\right)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}}\mathbb{E}_{\varpi}\left[\left(\hat{I}_n(\varpi,\boldsymbol{x}) - m_{\hat{I}_n}(\boldsymbol{x})\right)\left(\hat{I}_n(\varpi,\boldsymbol{x}') - m_{\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}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}}c_{\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}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}}\left[F\left([0\ 0];\boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}'),\boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}')\right) - \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)\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}}F\left([0\ 0];\boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}'),\boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}')\right)f_{\boldsymbol{X}}(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' - \left(\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}\right)^2 \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}}F\left([0\ 0];\boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}'),\boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}')\right)f_{\boldsymbol{X}}(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' - m^2_{\hat{P}_{f,n}},
\end{aligned}
\tag{24}
$$

where $\mathbb{V}_{\varpi}\left[\cdot\right]$ denotes the variance operation taken over Ω. The posterior distribution $\hat{P}_f$ reflects our uncertainty about P_f, which arises from the discretization error resulting from the fact that the g-function is only observed at a finite number of discrete locations. From this perspective, the proposed BFPI framework offers a principled approach to quantifying and propagating the numerical uncertainty behind the failure probability. Once given the data $\mathscr{D}$, the posterior mean $m_{\hat{P}_{f,n}}$ is a natural estimate for the failure probability

9

P_f, while the posterior variance $\sigma^2_{\hat{P}_{f,n}}$ can measure our uncertainty about the estimate.

Remark 2. The posterior mean function $m_{\hat{I}_n}(\boldsymbol{x})$ of the failure indicator function I (defined in Eq. (18)) is the same as that given in our recent work [22]. In that work, we also derived the closed-form expressions of the posterior variance function of I and an upper bound of the posterior covariance $c_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x'})$ (Eq. (19)) by using Cauchy-Schwarz inequality.

Remark 3. The posterior mean $m_{\hat{P}_{f,n}}$ of the failure probability P_f (defined in Eq. (23)) was previously given in [22]. In additional, an upper bound of the posterior variance $\sigma^2_{\hat{P}_{f,n}}$ (Eq. (24)) was derived based on the upper bound of $c_{\hat{I}_n}(\boldsymbol{x}, \boldsymbol{x'})$.

Remark 4. Numerical integration techniques are required to evaluate $m_{\hat{P}_{f,n}}$ and $\sigma^2_{\hat{P}_{f,n}}$ due to their analytical intractability. It is interesting that in our context the failure probability estimate $m_{\hat{P}_{f,n}}$ (Eq. (23)) is a integral over the whole domain $\mathcal{X}$ (both failure and safe domains), which is in contrast to the classical definition of failure probability (Eq. (1)) that is essentially a integral over the failure domain only. The former could be explained by the fact that Eq. (23) accounts for the numerical uncertainty at any $\boldsymbol{x} \in \mathcal{X}$ no matter where it is. Besides, if we assume that the numerical uncertainty approaches to zero (i.e., $\sigma^2_{\hat{P}_{f,n}}(\boldsymbol{x}) \to 0^+$ and $m_{\hat{P}_{f,n}}(\boldsymbol{x}) \to g(\boldsymbol{x})$), then there exits $\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \to I(\boldsymbol{x})$. In this regard, Eq. (1) can be seen as a limiting case of Eq. (23) when the numerical uncertainty disappears.

Remark 5. From the Bayesian perspective, the computation of failure probability estimate can be interpreted as a process aiming at reducing the numerical uncertainty that prevents us from inferring the true value. Therefore, an optimal inference about the failure probability requires an optimal decision on where to observe the g-function that leads to maximum reduction of the numerical uncertainty on the failure probability with as less g-function evaluations as possible.

3. Parallel adaptive-Bayesian failure probability learning

This section presents a novel method, termed 'parallel adaptive–Bayesian failure probability learning' (PA-BFPL), to make inference about the failure probability. The proposed method builds upon the BFPI framework, and aims at producing a reasonably accurate failure probability estimate using a limited number of observations from the g-function. This objective is achieved mainly by developing a variance-amplified importance sampling (VAIS) method and an adaptive parallel active learning (APAL) strategy, as described in what follows.

10

3.1. Variance-amplified importance sampling

In the BFPI framework, one open task consists of approximating the analytically intractable integrals ($m_{\hat{P}_{f,n}}$ in Eq. (23) and $\sigma^2_{\hat{P}_{f,n}}$ in Eq. (24)). The most straightforward solution would be to use the crude MCS due to its broad applicability. However, a considerably large number of samples are needed to achieve a reasonable accuracy in certain conditions, which can make each iteration of our method time-consuming and even cause the problem of computer memory loss. Taking $m_{\hat{P}_{f,n}}$ as an example, $\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)$ might be small where $f_{\boldsymbol{X}}(\boldsymbol{x})$ is large and vice versa. In these cases, directly sampling according to $f_{\boldsymbol{X}}(\boldsymbol{x})$ could be less efficient. If we turn to importance sampling, the optimal sampling density should be proportional to $\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})$, but is not practically achievable since it requires knowledge of the quantity we are trying to estimate. Similar problems also exist for $\sigma^2_{\hat{P}_{f,n}}$, and we will not repeat too much herein.

The present study proposes a VAIS technique to assess $m_{\hat{P}_{f,n}}$ and $\sigma^2_{\hat{P}_{f,n}}$. Let us reformulate $m_{\hat{P}_{f,n}}$ and $\sigma^2_{\hat{P}_{f,n}}$ as follows:

$$
\begin{aligned}
m_{\hat{P}_{f,n}} &= \int_{\mathcal{X}} \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{h_{\boldsymbol{X}}(\boldsymbol{x})} h_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x},
\end{aligned}
\tag{25}
$$

$$
\begin{aligned}
\sigma^2_{\hat{P}_{f,n}} &= \int_{\mathcal{X}}\int_{\mathcal{X}} \left[F\left([0\ 0]; \boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}'), \boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}')\right) - \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}')}{\sigma_{\hat{g}_n}(\boldsymbol{x}')}\right) \right] f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \left[F\left([0\ 0]; \boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}'), \boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x},\boldsymbol{x}')\right) - \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x})}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right)\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}')}{\sigma_{\hat{g}_n}(\boldsymbol{x}')}\right) \right] \frac{f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}')}{h_{\boldsymbol{X}}(\boldsymbol{x}) h_{\boldsymbol{X}}(\boldsymbol{x}')} h_{\boldsymbol{X}}(\boldsymbol{x}) h_{\boldsymbol{X}}(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}',
\end{aligned}
\tag{26}
$$

where $h_{\boldsymbol{X}}(\boldsymbol{x})$ is the so-called 'importance sampling density' (ISD). In this study, we do not intend to approach a nearly optimal ISD (whose formulation may be challenging), yet a simple but effective one. The concept of increasing the variances of random variables has been used in the different contexts, such as [25, 26, 27, 28]. In particular, it has been reported in [29] that such an approach was used within Importance Sampling as early as 1983. Following those ideas, the ISD $h_{\boldsymbol{X}}(\boldsymbol{x})$ is simply constructed by amplifying the standard deviations $\boldsymbol{\sigma}_{\boldsymbol{X}}$ (or equivalently variance $\boldsymbol{\sigma}_{\boldsymbol{X}}^2$) of $f_{\boldsymbol{X}}(\boldsymbol{x})$ (keep the mean $\boldsymbol{m}_{\boldsymbol{X}}$ unchanged), i.e., $h_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{m}_{\boldsymbol{X}}, \alpha\boldsymbol{\sigma}_{\boldsymbol{X}})$, where $\alpha \geq 1$ denotes the amplification factor of standard deviation. Note that for any X_i that follows a uniform distribution, its standard deviation does not need to be enlarged. Besides, one can use different amplification factors for different random variables, but for the sake of convenience we just consider a single amplification factor in this work. The unbiased VAIS estimators of $m_{\hat{P}_{f,n}}$ and $\sigma^2_{\hat{P}_{f,n}}$

are simply given as their sample means:

$$\hat{m}_{\hat{P}_{f,n}} = \frac{1}{N_1} \sum_{i=1}^{N_1} \left[\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})} \right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})} \right], \tag{27}$$

$$\hat{\sigma}^2_{\hat{P}_{f,n}} = \frac{1}{N_2} \sum_{j=1}^{N_2} \left[F\left([0\ 0]; \boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x}^{(j)}, \boldsymbol{x}'^{,(j)}), \boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x}^{(j)}, \boldsymbol{x}'^{,(j)}) \right) - \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}^{(j)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(j)})} \right) \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}'^{,(j)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}'^{,(j)})} \right) \right] \\ \times \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(j)}) f_{\boldsymbol{X}}(\boldsymbol{x}'^{,(j)})}{h_{\boldsymbol{X}}(\boldsymbol{x}^{(j)}) h_{\boldsymbol{X}}(\boldsymbol{x}'^{,(j)})}, \tag{28}$$

where $\left\{ \boldsymbol{x}^{(i)} \right\}_{i=1}^{N_1}$ is a set of N_1 random samples generated according to $h_{\boldsymbol{X}}(\boldsymbol{x})$; $\left\{ \boldsymbol{x}^{(j)} \right\}_{j=1}^{N_2}$ and $\left\{ \boldsymbol{x}'^{,(j)} \right\}_{j=1}^{N_2}$ are two sets of N_2 random samples generated according to $h_{\boldsymbol{X}}(\boldsymbol{x})$ and $h_{\boldsymbol{X}}(\boldsymbol{x}')$ respectively. The variances of the VAIS estimators $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}^2_{\hat{P}_{f,n}}$ are given by:

$$\mathrm{Var}\left[\hat{m}_{\hat{P}_{f,n}} \right] = \frac{1}{N_1(N_1-1)} \sum_{i=1}^{N_1} \left[\Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}^{(i)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(i)})} \right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})} - \hat{m}_{\hat{P}_{f,n}} \right]^2, \tag{29}$$

$$\mathrm{Var}\left[\hat{\sigma}^2_{\hat{P}_{f,n}} \right] = \frac{1}{N_2(N_2-1)} \sum_{j=1}^{N_2} \left\{ \left[F\left([0\ 0]; \boldsymbol{m}_{\hat{g}_n}(\boldsymbol{x}^{(j)}, \boldsymbol{x}'^{,(j)}), \boldsymbol{C}_{\hat{g}_n}(\boldsymbol{x}^{(j)}, \boldsymbol{x}'^{,(j)}) \right) - \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}^{(j)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}^{(j)})} \right) \Phi\left(\frac{-m_{\hat{g}_n}(\boldsymbol{x}'^{,(j)})}{\sigma_{\hat{g}_n}(\boldsymbol{x}'^{,(j)})} \right) \right] \\ \times \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(j)}) f_{\boldsymbol{X}}(\boldsymbol{x}'^{,(j)})}{h_{\boldsymbol{X}}(\boldsymbol{x}^{(j)}) h_{\boldsymbol{X}}(\boldsymbol{x}'^{,(j)})} - \hat{\sigma}^2_{\hat{P}_{f,n}} \right\}^2, \tag{30}$$

where $\mathrm{Var}[\cdot]$ means to take variance of its argument.

When $\alpha = 1$, the proposed VAIS method reduces to crude MCS. In case that $\alpha > 1$, $h_{\boldsymbol{X}}(\boldsymbol{x})$ can be viewed as an auxiliary sampling density formed by redistributing the density of $f_{\boldsymbol{X}}(\boldsymbol{x})$. Typically, $h_{\boldsymbol{X}}(\boldsymbol{x})(\alpha > 1)$ is more dispersedly distributed than $f_{\boldsymbol{X}}(\boldsymbol{x})$ over $\mathcal{X}$. As an illustration, Fig. 2 compares the density change of a standard normal density $\phi(x)$ before and after its variance is amplified, where two amplification factors are considered, i.e., $\alpha = 1.5, 2.0$. It is shown that as α increases, $h_{\boldsymbol{X}}(\boldsymbol{x})$ becomes more flatter than $\phi(x)$, and hence enlarges density where $\phi(x)$ is small, while lowers the density where $\phi(x)$ is large. Consequently, the variance amplification will have an effect on random sampling. To be specific, the random samples generated from $h_{\boldsymbol{X}}(\boldsymbol{x})(\alpha > 1)$ are more dispersedly distributed than those of $f_{\boldsymbol{X}}(\boldsymbol{x})$ over $\mathcal{X}$. If we take $f_{\boldsymbol{X}}(\boldsymbol{x}) \sim \phi(x_1)\phi(x_2)$ as an example, the random samples generated before and after variance amplification are depicted in Fig. 3, where two cases (i.e., $\alpha = 1.5, 2.0$) are also considered. As can been seen, as the amplification factor increases, the random samples will reach the area where $f_{\boldsymbol{X}}(\boldsymbol{x})$ is relatively small. Thus, sampling from $h_{\boldsymbol{X}}(\boldsymbol{x})$ instead of $f_{\boldsymbol{X}}(\boldsymbol{x})$ could alleviate some of limitations discussed at the beginning of this subsection. The effect of variance amplification on random sampling is also useful for our adaptive parallel

12

active learning strategy (see next subsection). The optimal α values for $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}^2_{\hat{P}_{f,n}}$ could be determined by minimizing their corresponding variances (Eqs. (29) and (30)), which, however, is still a tricky task. To determine the appropriate sample sizes N_1 and N_2 for $\hat{m}_{\hat{P}_{f,n}}$ and $\hat{\sigma}^2_{\hat{P}_{f,n}}$, one can first assign them two small values, and then gradually increase the sample sizes until $\mathrm{Var}\left[\hat{m}_{\hat{P}_{f,n}}\right]$ and $\mathrm{Var}\left[\hat{\sigma}^2_{\hat{P}_{f,n}}\right]$ are acceptable.
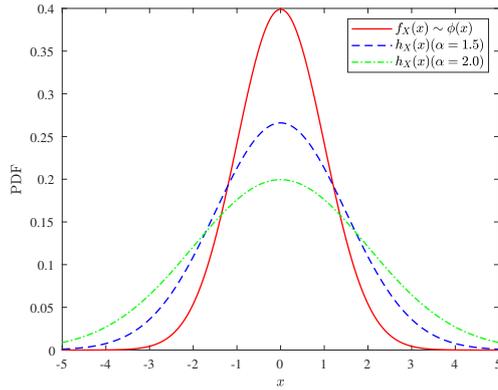
Figure 2: Illustration the effects of variance amplification from density change.
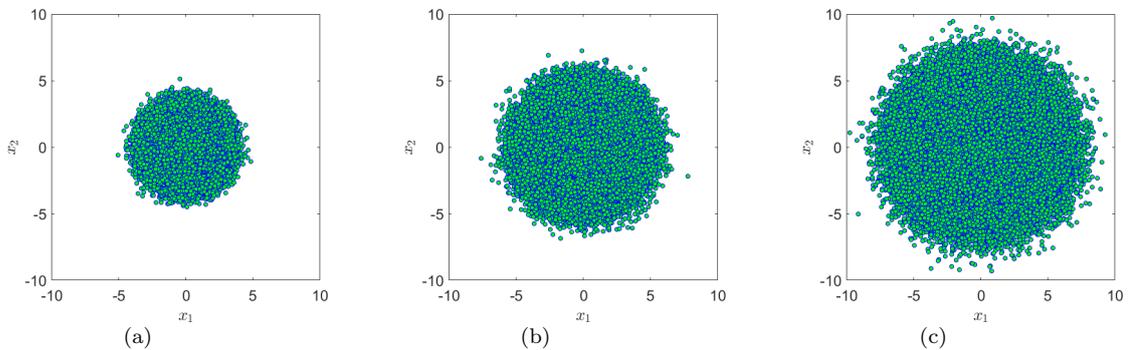
Figure 3: Illustration the effects of variance amplification from random sampling: (a) 10^6 random samples drawn from $f_{\boldsymbol{X}}(\boldsymbol{x}) \sim \phi(x_1)\phi(x_2)$; (b) 10^6 random samples drawn from $h_{\boldsymbol{X}}(\boldsymbol{x})$ ($\alpha = 1.5$); (c) 10^6 random samples drawn from $h_{\boldsymbol{X}}(\boldsymbol{x})$ ($\alpha = 2.0$).

Remark 6. As a common limitation in Importance Sampling [30], the proposed VAIS method could not be directly applied to high-dimensional problems (e.g., larger than 20). Besides, a premise of the proposed VAIS method is that all random variables process variances. In case that there exist a random variable without variance (e.g., Cauchy distribution), some pre-processing steps are needed in order to apply the VAIS method, e.g., transforming it to a random variable with variance if possible.

Remark 7. In [1], the authors developed a importance ball sampling (IBS) method to approximate

$m_{\hat{P}_{f,n}}$ and an upper bound of $\sigma^2_{\hat{P}_{f,n}}$. However, the method is biased in nature and has to work in the standard normal space.

Remark 8. It is interesting to note that VAIS itself is a purely frequentist approach. As in many Bayesian methodologies, the frequentist methods also play a significant role [31].

3.2. Adaptive parallel active learning strategy

Another issue to be solved in the BFPI framework is how to select design points $\boldsymbol{\mathcal{X}}$, which is commonly known as design of experiments (DOE). Although the BFPI framework itself does not impose any restrictions on the DOE, an optimal DOE can yield an accurate estimate for the failure probability with a minimum number of g-function evaluations. In view of this, we propose an APAL strategy to sequentially select a batch of points, which attempts to make the fullest possible use of all previous g-function evaluations and parallel computing simultaneously. The core of the APAL strategy is a weighted clustering technique.

Considering the posterior distribution $\hat{g}_n$ defined in Eq. (13), the probability of making a wrong prediction on the sign of g at $\boldsymbol{x}$ is given by [20]:

$$\pi(\boldsymbol{x}) = \Phi\left(\frac{-|m_{\hat{g}_n}(\boldsymbol{x})|}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right). \tag{31}$$

For simplicity, we refer to $\pi(\boldsymbol{x})$ as the probability of misclassification (POM). The well known U function [20] (i.e., $U(\boldsymbol{x}) = \frac{|m_{\hat{g}_n}(\boldsymbol{x})|}{\sigma_{\hat{g}_n}(\boldsymbol{x})}$) is proposed based on the concept of POM, and the best next point to evaluate on the g-function is identified by minimizing the U function (equivalently maximizing the POM). However, only the misclassification probability at a single point that minimizes the U function is considered, without taking other points and the probability distribution information of $\boldsymbol{X}$ into account. This may lead to underutilization of useful information and is not suitable for parallel distributed processing.

To overcome these limitations, a new concept, called 'expected misclassification probability' (EMP), is first introduced as follows:

$$\Pi = \mathbb{E}_{\boldsymbol{x}}\left[\pi(\boldsymbol{x})\right] = \int_{\mathcal{X}} \Phi\left(\frac{-|m_{\hat{g}_n}(\boldsymbol{x})|}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{32}$$

which is actually defined as an expectation of the POM $\pi(\boldsymbol{x})$ under the density $f_{\boldsymbol{X}}(\boldsymbol{x})$. Hence, the EMP can be interpreted as the posterior expected probability that $\hat{g}_n$ makes a mistake on the sign of g. In order to have an accurate failure probability estimate, an alternative way is to reduce Π instead of the maximum value of $\pi(\boldsymbol{x})$.

14

Let us rewrite Π with respect to $h_{\boldsymbol{X}}(\boldsymbol{x})$ as:

$$\Pi = \int_{\mathcal{X}} \Phi\left(\frac{-|m_{\hat{g}_n}(\boldsymbol{x})|}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{h_{\boldsymbol{X}}(\boldsymbol{x})} h_{\boldsymbol{X}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{33}$$

To reduce Π defined in Eq. (33), one potential solution is to find out the locations that contribute the most to Π. Here, we introduce a measure (i.e., the proposed learning function), called 'expected misclassification probability contribution' (EMPC), as follows:

$$\mathrm{EMPC}(\boldsymbol{x}) = \Phi\left(\frac{-|m_{\hat{g}_n}(\boldsymbol{x})|}{\sigma_{\hat{g}_n}(\boldsymbol{x})}\right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{h_{\boldsymbol{X}}(\boldsymbol{x})}. \tag{34}$$

It is straightforward to observe that the EMPC function provides a natural measure of the contribution of the misclassification probability at $\boldsymbol{x}$ to Π, where $\boldsymbol{x} \sim h_{\boldsymbol{X}}(\boldsymbol{x})$. Besides, one should note that the probability density $f_{\boldsymbol{X}}(\boldsymbol{x})$ is properly included in the EMPC function.

Now, we consider the problem of how to identify a batch of informative points among a set of points generated from $h_{\boldsymbol{X}}(\boldsymbol{x})$, e.g., $\left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^{N_3}$. This objective is realized by developing a weighted clustering algorithm, called 'EMPC-weighted k-means clustering'. As its name indicates, the proposed algorithm actually combines EMPC with k-means clustering [32]. As mentioned before, the EMPC function can measure the contribution of the misclassification probability at $\boldsymbol{x}^{(l)}$ to Π. On the other hand, the k-means clustering algorithm can partition a dataset into k clusters that are represented by k centroids. However, the conventional k-means clustering algorithm does not account for the weight information of data. The proposed EMPC-weighted k-means clustering enables to identify k centroids by using the data $\left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^{N_3}$ while considering their EMPC values as weights. The k centroids correspond to the batch of points we wish to select. Once the k points are obtained, computation of the corresponding g-function values can be distributed on k CPU cores simultaneously. A compact pseudocode of the proposed algorithm is given in Algorithm 1.

The reason why we introduce the ISD $h_{\boldsymbol{X}}(\boldsymbol{x})$ to Eq. (33) (and hence in Eq. (34)) is because with the same sample size $h_{\boldsymbol{X}}(\boldsymbol{x})$ can generate much more dispersed samples than $f_{\boldsymbol{X}}(\boldsymbol{x})$, making it possible to reach the failure domain characterized with a small failure probability. Besides, by doing so, the random samples generated for evaluating, e.g., $\hat{m}_{\hat{P}_{f,n}}$, can be reused in the proposed weighted clustering algorithm. To illustrate the proposed weighted clustering method, let us consider the case that: $\mathrm{EMPC}(\boldsymbol{x}) = \Phi\left(-\left(x_1^2 + x_2^4 - 4\right)^2\right) \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{h_{\boldsymbol{X}}(\boldsymbol{x})}$, $f_{\boldsymbol{X}}(\boldsymbol{x}) \sim \phi(x_1)\phi(x_2)$, $\alpha = 1.5$, $N_3 = 10^6$ and $k = 5$. As shown in Fig. 4, the identified points are sparsely located in the region with relatively large EMPC values, and hence informative in our context.

15

Algorithm 1 Proposed EMPC-weighted k-means clustering algorithm
---
Input: The EMPC function, number of clusters k and dataset $\left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^{N_3}$

1. **Initialization**. Randomly choose k points from the dataset $\left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^{N_3}$ as the initial centroids, denoted by $\boldsymbol{S} = \left\{\boldsymbol{s}^{(i)}\right\}_{i=1}^{k}$;

2. **Assignment step**. Assign each point among the dataset $\left\{\boldsymbol{x}^{(l)}\right\}_{l=1}^{N_3}$ to the nearest cluster: that with the least squared Euclidean distance. The i-th cluster is denoted as $\boldsymbol{R}^{(i)} = \left\{\boldsymbol{r}_j^{(i)}\right\}_{j=1}^{N^{(i)}}$, where $\boldsymbol{r}_j^{(i)}$ is the j-th point in the i-th cluster and $N^{(i)}$ is the number of points in the i-th cluster;

3. **Update step**. The i-th centroid is updated by the EMPC-weighted mean of the points belonging to i-th cluster:

$$\boldsymbol{s}^{(i)} = \frac{\sum_{j=1}^{N^{(i)}} \text{EMPC}(\boldsymbol{r}_j^{(i)}) \times \boldsymbol{r}_j^{(i)}}{\sum_{j=1}^{N^{(i)}} \text{EMPC}(\boldsymbol{r}_j^{(i)})}$$

4. **Iteration**. Repeat steps 2 and 3 until the centroids do not change or the pre-specified number of iterations is reached.

Output: k centroids

Remark 9. For reducing the numerical uncertainty of $\hat{P}_{f,n}(\varpi)$, one obvious way is to minimize its variance $\sigma_{\hat{P}_{f,n}}^2$. However, the variance itself is analytically intractable, in contrast to the proposed EMPC function.

Remark 10. The basic idea of proposed APAL is similar to the one in [1], while different learning functions are used.

3.3. Numerical implementation procedure

The numerical implementation procedure of the proposed PA-BFPL consists of the following main steps:

Step 1: Generate samples according to the ISD $h_{\boldsymbol{X}}(\boldsymbol{x})$

In order to approximate $m_{\hat{P}_{f,n}}$ and $\sigma_{\hat{P}_{f,n}}^2$ by the proposed VAIS method, random samples need to be generated according to the ISD $h_{\boldsymbol{X}}(\boldsymbol{x})$. First, draw a set of N_1 random samples from $h_{\boldsymbol{X}}^{(1)}(\boldsymbol{x})$ (that corresponds to α_1), denoted by $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N_1}$. Then, draw two sets of N_2 random samples from $h_{\boldsymbol{X}}^{(2)}(\boldsymbol{x})$ and $h_{\boldsymbol{X}}^{(2)}(\boldsymbol{x}')$ (that correspond to α_2) respectively, denoted by $\left\{\boldsymbol{x}^{(j)}\right\}_{j=1}^{N_2}$ and $\left\{\boldsymbol{x}'^{,(j)}\right\}_{j=1}^{N_2}$.

The reason why we introduce two amplification factors α_1 and α_2 is because $\sigma_{\hat{P}_{f,n}}^2$ is much more time consuming to evaluate than $m_{\hat{P}_{f,n}}$. By doing so, we can use a larger α_2 and hence a smaller N_2 for $\sigma_{\hat{P}_{f,n}}^2$ in
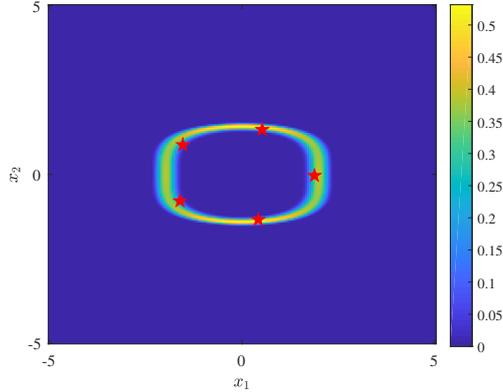
Figure 4: Five points identified by the proposed weighted clustering algorithm: the colormap represents the EMPC values and the pentagrams denote the identified centroids.

order to save computational time. Once α_1 and α_2 are properly chosen, one can specify N_1 and N_2 either adaptively or as large as possible.

Step 2: Obtain an initial dataset $\mathscr{D}$ from the g-function

To perform the proposed PA-BFPL method, an initial dataset observed from the g-function is required. First, generate a set of n_0 samples from $f_{\boldsymbol{X}}(\boldsymbol{x})$ by using Latin hypercube sampling (LHS), which is denoted by $\mathscr{X} = \{\boldsymbol{x}^{(l)}\}_{l=1}^{n_0}$. Then, these points are evaluated on the g-function in parallel, and the corresponding observations are denoted by $\mathscr{Y} = \{y^{(l)}\}_{l=1}^{n_0}$ with $y^{(l)} = g(\boldsymbol{x}^{(l)})$. At last, the initial dataset is constructed by $\mathscr{D} = \{\mathscr{X}, \mathscr{Y}\}$. Let $n = n_0$.

Step 3: Compute the posterior mean and variance of P_f

The posterior distribution of g conditional on $\mathscr{D}$ can be inferred as Eq. (13), which mainly involves learning the hyper-parameters using maximum likelihood estimation. This in turn implies posterior distributions over I and P_f. The posterior mean $m_{\hat{P}_{f,n}}$ and variance $\sigma^2_{\hat{P}_{f,n}}$ are approximated by the proposed VAIS method (Eqs. (27) and (28)) using samples $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N_1}$, $\{\boldsymbol{x}^{(j)}\}_{j=1}^{N_2}$ and $\{\boldsymbol{x}'^{(j)}\}_{j=1}^{N_2}$, respectively.

Step 4: Check the stopping criterion

A stopping criterion that determines when to stop the iteration is needed. In this study, we present a hybrid convergence measure consisting of two indices. The first index is defined as the relative error of

17

$\hat{m}_{\hat{P}_{f,n}}$ between two consecutive iterations:

$$e_1 = \frac{|\hat{m}_{\hat{P}_{f,n}}^{(q)} - \hat{m}_{\hat{P}_{f,n-k}}^{(q-1)}|}{\hat{m}_{\hat{P}_{f,n-k}}^{(q-1)}}, \tag{35}$$

where $\hat{m}_{\hat{P}_{f,n-k}}^{(q-1)}$ and $\hat{m}_{\hat{P}_{f,n}}^{(q)}$ are the estimated failure probabilities at the $(q-1)$-th and q-th iterations, respectively. The first index e_1 can measure the stability of the estimated failure probability. The estimated posterior coefficient of variation (COV) of the failure probability is considered as the second index such that:

$$e_2 = \frac{\hat{\sigma}_{\hat{P}_{f,n}}^{(q)}}{\hat{m}_{\hat{P}_{f,n}}^{(q)}}, \tag{36}$$

where $\hat{m}_{\hat{P}_{f,n}}^{(q)}$ and $\hat{\sigma}_{\hat{P}_{f,n}}^{(q)}$ represent the estimated posterior mean and standard deviation of the failure probability at the q-th iteration. The second index e_2 implies the level of epistemic uncertainty of the failure probability estimate. Based on these two indices, this step proceeds as follows:

If both $e_1 < \epsilon_1$ and $e_2 < \epsilon_2$ are satisfied twice in succession, go to **Step 6**; Else, go **Step 5**. Here ϵ_1 and ϵ_2 are two user-specified thresholds. 'Twice in succession' is adopted here to avoid possible fake convergence.

Step 5: Enrich the dataset by the proposed APAL

This stage consists of identifying k new points $\boldsymbol{\mathscr{X}}^+ = \left\{\boldsymbol{x}^{+,(i)}\right\}_{i=1}^{k}$ from $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N_1}$ using the proposed APAL (i.e., EMPC-weighted k-means clustering). Then, the g-function is evaluated in parallel at $\boldsymbol{\mathscr{X}}^+$ to produce the corresponding observations $\boldsymbol{\mathscr{Y}}^+ = \left\{y^{+,(i)}\right\}_{i=1}^{k}$ with $y^{+,(i)} = g(\boldsymbol{x}^{+,(i)})$. Finally, the previous dataset is enriched by $\boldsymbol{\mathscr{D}}^+ = \{\boldsymbol{\mathscr{X}}^+, \boldsymbol{\mathscr{Y}}^+\}$, i.e., $\boldsymbol{\mathscr{D}} = \boldsymbol{\mathscr{D}} \cup \boldsymbol{\mathscr{D}}^+$. Let $n = n + k$, and go to **Step 3**.

Step 6: End the algroithm

The proposed method stops and the last failure probability estimate is considered as the result of the method.

There remain several parameters in the proposed algroithm to be specified. In all numerical examples of this study, unless otherwise specified these parameters except for k are set to: $N_1 = 10^6$, $N_2 = 5 \times 10^4$, $\alpha_1 = 1.6$, $\alpha_2 = 2.1$, $n_0 = 10$, $\epsilon_1 = 15\%$ and $\epsilon_2 = 5\%$. The parameter k will be varied to test the performance of the proposed method.

4. Numerical examples

The performance of the proposed PA-BFPL method will be illustrated in this section by means of four numerical examples. These examples cover a variety of problems with varying dimensions, non-linearity and failure probabilities, etc. The reference results for the target failure probabilities are provided by MCS when there is no (semi-) analytical solution. For comparison, AK-MCS [20], Active Learning Probabilistic Integration (ALPI) [22], Active learning Kriging Markov Chain Monte Carlo (AK-MCMC) [33] and other methods are also considered if applicable. In particular, the active learning reliability (ALR) method in UQLab (version 2.0.0) [34], denoted as ALR in UQLab 2.0.0, is compared to the proposed method in all four numerical examples. If not further specified, the ALR method runs with its default setting [35]. The efficiency of these methods is measured by the number of iterations N_{iter}, the total number of g-function calls N_{call}, while the accuracy is measured by the failure probability estimate $\hat{P}_f$ and its COV denoted by $\text{COV}\left[\hat{P}_f\right]$. Except for MCS and the (semi-) analytical method, these performance measures are estimated from the average results over 10 independent runs unless otherwise specified.

4.1. Example 1: A test problem with four beta points

The first example considers a test problem with multiple beta points, which is modified from [36]. The performance function is given by:

$$Y = g\left(X_1, X_2\right) = \beta^2 - |X_1 \cdot X_2|, \tag{37}$$

where β is a constant parameter, specified as 3; X_1 and X_2 are two standard normal variables. It is easy to know that the limit state surface $g\left(x_1, x_2\right) = 0$ has four beta points: (β, β), $(-\beta, \beta)$, $(\beta, -\beta)$ and $(-\beta, -\beta)$. Another characteristic of the problem is that the semi-analytic formula of the failure probability can be derived as:

$$P_f = 1 - \frac{2}{\pi} \int_0^{\beta^2} K_0\left(u\right) \mathrm{d}u, \tag{38}$$

where $K_0(\cdot)$ is the modified Bessel function of the second kind of order zero. By applying some numerical integration techniques, it is trivial to obtain the result of Eq. (38) with sufficient accuracy.

The proposed method with different k is implemented to assess the failure probability, together with several other existing methods. The results are summarized in Table. 1. It can be seen that the failure probability provided by the the semi-analytic formula accords well with that of MCS, and hence we take

3.09×10^{-5} as the reference result. The proposed method is able to yield unbiased estimates with COVs less than 3%. The two AK-MCS methods are less accurate than the proposed method in terms of the average failure probabilities and their COVs. On the other hand, the proposed method greatly outperforms the non-parallel counterparts (i.e., AK-MCS+U [20], AK-MCS+EFF [20] and AK-MCMC [33]) in terms of N_{iter}, especially compared with AK-MCMC. This implies that the proposed method can be much more efficient than those non-parallel counterparts when parallel computing is available. When it comes to the parallel counterpart, i.e., ALR in UQLab 2.0.0 [35], the proposed method needs slightly more computational efforts than it, regarding both N_{iter} and N_{call}. However, the ALR method produces obvious biases for the failure probabilities for all three cases (i.e., $k = 5, 10, 15$). For $k = 10, 15$, the COVs of the ALR method even reach up to 38.32% and 34.06% respectively. The reason for why the biased results are yielded could be due to the limitation of subset simulation used in ALR, as has been revealed in [36]. In view of these, the proposed method also shows better overall performance than the ALR method in this example.

Table 1: Results for Example 1 by different methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
Semi-analytic		-	-	3.09×10^{-5}	-
MCS		-	10^9	3.09×10^{-5}	0.57
AK-MCS+U		$1 + 41.50 = 42.50$	$12 + 41.50 = 53.50$	3.13×10^{-5}	7.15
AK-MCS+EFF		$1 + 45.90 = 46.90$	$12 + 45.90 = 57.90$	3.03×10^{-5}	6.65
AK-MCMC		$1 + 100.90 = 101.90$	$12 + 100.90 = 112.90$	3.09×10^{-5}	0.72
	$k = 5$	$1 + 5.20 = 6.20$	$10 + 26.00 = 36.00$	1.64×10^{-5}	3.80
ALR in UQLab 2.0.0	$k = 10$	$1 + 4.30 = 5.30$	$10 + 43.00 = 53.00$	2.16×10^{-5}	38.32
	$k = 15$	$1 + 3.70 = 4.70$	$10 + 55.50 = 65.50$	2.19×10^{-5}	34.16
	$k = 5$	$1 + 7.40 = 8.40$	$10 + 37.00 = 47.00$	3.07×10^{-5}	2.55
Proposed PA-BFPL	$k = 10$	$1 + 5.20 = 6.20$	$10 + 52.00 = 62.00$	3.08×10^{-5}	1.41
	$k = 15$	$1 + 4.80 = 5.80$	$10 + 72.00 = 82.00$	3.07×10^{-5}	0.97

For illustration purposes, Fig. 5 depicts the identified points resulted from an exemplary run of the proposed method ($k = 10$), along with the true limit state. It is shown that as the iteration goes on, the points identified by the proposed method gradually move towards the vicinity of the true limit state.

20
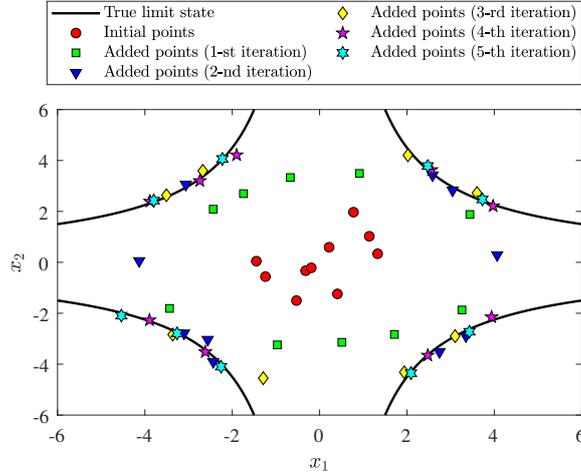
Figure 5: Selected points by the proposed method ($k = 10$) for Example 1.

Moreover, the selected points, at least most of them, are sparsely distributed in the design space, but slightly denser around the true beta points. These results imply that these points are informative for accurately inferring the failure probability.

4.2. Example 2: A series system with four branches

The second example consists of a series system with four branches, which has been extensively studied as a benchmark [20, 9, 22]. The performance function is given by:

$$
Y = g\left(X_1, X_2\right) = \min \begin{cases} a + \frac{(X_1 - X_2)^2}{10} - \frac{(X_1 + X_2)}{\sqrt{2}}, \\ a + \frac{(X_1 - X_2)^2}{10} + \frac{(X_1 + X_2)}{\sqrt{2}}, \\ (X_1 - X_2) + \frac{b}{\sqrt{2}}, \\ (X_2 - X_1) + \frac{b}{\sqrt{2}} \end{cases},
\tag{39}
$$

where a and b are two constant parameters, which can be used to adjust the failure probability; X_1 and X_2 are normally distributed with zero means and unit variances. In the following, two cases are considered.

Case I: $a = 3$ and $b = 7$

In this case, the target failure probability is in the order of magnitude 10^{-3}, as indicated by the reference result from MCS (i.e., $\hat{P}_f = 2.22 \times 10^{-3}$ with COV$\left[\hat{P}_f\right] = 0.21\%$). The proposed method is also compared with several other methods, as listed in Table 2. It is found that all methods can give close average failure probabilities to the reference result with COVs less than 5%, except for ALR that processes COVs larger

21

than 10%. The latter is due to the fact that ALR cannot always identify all failure domains. As for N_{iter}, the proposed method is significantly advantageous against these non-parallel methods (i.e., AK-MCS+U [20], AK-MCS+EFF [20] and ALPI [22]), and still fairly better than these parallel methods (i.e., ISKRA (KB) [37], ISKRA (k-means) [37] and ALR [35]). In addition, the average number of g-function calls required by the proposed method is also less than the other methods, especially when k is small, say $k = 5$. These results demonstrate the accuracy and efficiency of the proposed method in this case.

Table 2: Results of Example 2 (Case I) by different methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^8	2.22×10^{-3}	0.21
AK-MCS+U		$1 + 82.20 = 83.20$	$12 + 82.20 = 94.20$	2.22×10^{-3}	1.35
AK-MCS+EFF		$1 + 103.20 = 104.20$	$12 + 103.20 = 115.20$	2.21×10^{-3}	1.20
ALPI		$1 + 70.70 = 71.70$	$12 + 70.70 = 72.70$	2.22×10^{-3}	2.25
ISKRA (KB) [37]	$k = 12$	$1 + 6.68 = 7.68$	$12 + 80.16 = 92.16$	2.23×10^{-3}	1.50
ISKRA (k-means) [37]	$k = 12$	$1 + 8.62 = 9.62$	$12 + 103.44 = 115.44$	2.22×10^{-3}	1.50
	$k = 5$	$1 + 10.90 = 11.90$	$10 + 54.50 = 64.50$	2.05×10^{-3}	11.95
ALR in UQLab 2.0.0	$k = 12$	$1 + 6.40 = 7.40$	$10 + 76.80 = 86.80$	2.07×10^{-3}	18.55
	$k = 15$	$1 + 5.20 = 6.20$	$10 + 78.00 = 88.00$	1.94×10^{-3}	17.92
	$k = 5$	$1 + 6.50 = 7.50$	$10 + 32.50 = 42.50$	2.13×10^{-3}	3.07
Proposed PA-BFPL	$k = 12$	$1 + 4.30 = 5.30$	$10 + 51.60 = 61.60$	2.24×10^{-3}	1.59
	$k = 15$	$1 + 3.40 = 4.40$	$10 + 51.00 = 61.00$	2.22×10^{-3}	1.08

Note: The results of ISKRA (KB) and ISKRA (k-means) are directly taken from [37], and they were averaged over 50 independent runs.

Fig. 6(a) shows the points selected by the proposed method ($k = 10$) with an exemplary run. It is observed that most of the points identified from iterations 1-5 are scattered in the vicinity of true limit state, indicating the effectiveness of the proposed APAL strategy.

Case II: $a = 5$ and $b = 9$

The second case is more challenging than the first one since the target failure probability is relatively small, i.e., in the order of 10^{-6} as provided by MCS with 10^{10} samples. Table 3 compares the results from

MCS, AK-MCMC [33], ALR [35] and the proposed method. As can be seen, fairly accurate results for such a small failure probability can still be produced by the proposed method with different k. Besides, the proposed method requires far less N_{iter} and N_{call} than those of AK-MCMC, especially for N_{iter}. The ALR method still produces biased results with considerably large COVs in this case, though it requires similar N_{iter} and N_{call} than the proposed method. The results indicate that the proposed method is accurate and efficient in such a case.

Table 3: Results of Example 2 (Case II) by different methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^{10}	7.09×10^{-6}	0.38
AK-MCMC		$1 + 127.50 = 128.50$	$12 + 127.50 = 139.50$	7.10×10^{-6}	1.37
	$k = 5$	$1 + 9.90 = 10.90$	$10 + 49.50 = 59.50$	4.82×10^{-6}	79.59
ALR in UQLab 2.0.0	$k = 10$	$1 + 6.70 = 7.70$	$10 + 67.00 = 77.00$	4.42×10^{-6}	81.79
	$k = 15$	$1 + 5.30 = 6.30$	$10 + 79.50 = 89.50$	6.50×10^{-6}	49.85
	$k = 5$	$1 + 10.00 = 11.00$	$10 + 50.00 = 60.00$	7.04×10^{-6}	2.17
Proposed PA-BFPL	$k = 10$	$1 + 5.80 = 6.80$	$10 + 58.00 = 68.00$	7.13×10^{-6}	2.01
	$k = 15$	$1 + 5.10 = 6.10$	$10 + 76.50 = 86.50$	7.06×10^{-6}	1.20

Once again, we depict the points selected at different stages of the proposed method ($k = 10$) via an exemplary run in Fig. 6(b). One can see that the identified points gradually approach to the four important parts of the true limit state that are relatively important for failure probability estimation. This demonstrates the effectiveness of the proposed method.

4.3. Example 3: A slender column

The third example considers a sufficiently slender column subject to an axial compressive force [38], as shown in Fig. 7. The performance function corresponding to the buckling failure is given by:

$$Z = g(\boldsymbol{X}) = \frac{\pi^2 E}{L^2} \left\{ \frac{\pi}{64} \left[(D + T)^4 - D^4 \right] \right\} - P, \tag{40}$$

where $\boldsymbol{X} = [E, D, T, L, P]$, as listed in Table 4.

The proposed method is compared in Table 5 with several other methods, i.e., MCS, AK-MCS+U [20], ALPI [22] and ALR [35]. The MCS with 10^7 samples can produce a failure probability estimate with a very
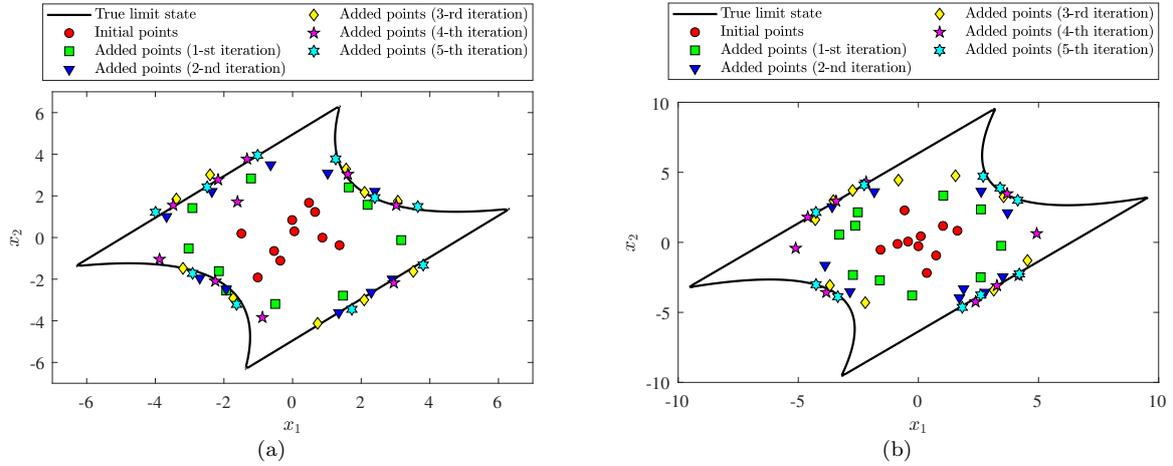
Figure 6: Selected points by the proposed method ($k = 10$) for Example 1: (a) Case I; (b) Case II.

Table 4: Random variables of Example 3.

Variable	Description	Distribution	Mean	Standard deviation
E/Gpa	Young's modulus	Normal	203	30.45
D/mm	Dimension of the section	Normal	23.5	2.0
T/mm	Dimension of the section	Normal	4	1
L/mm	Height of the column	Normal	2500	50
P/N	Axial load	Lognormal	1000	200

small COV, and hence it is taken as a reference. The results of $\hat{P}_f$ and $\mathrm{COV}[\hat{P}_f]$ show that AK-MCS+U, ALPI, ALR and the proposed method have similar accuracy. However, the proposed method is much more efficient than AK-MCS+U, ALPI and ALR in terms of N_{iter}. Besides, when $k = 5$ the proposed method also requires less calls to the g-function in average than all those methods being compared. Overall, this example demonstrates the potential high-efficiency advantage of PA-BFPL when parallel computing facilities are available.

4.4. Example 4: A transmission tower

To illustrate the practical applicability of the proposed method, a transmission tower structure subject to horizontal loads (Fig. 8) is considered as the last example, which is modified from [39]. The structure is modelled as a three-dimensional (3-D) truss using the finite element software OpenSees. The finite element

24
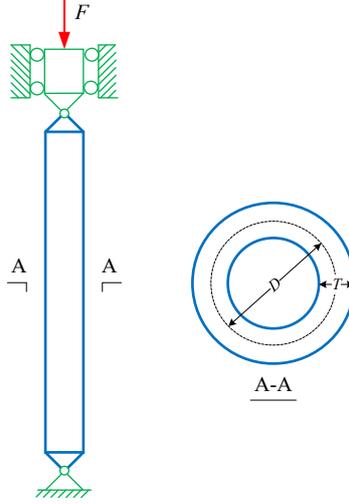
Figure 7: A slender column subject to an axial compressive force.

model consists of 24 joints and 80 truss members. As schematized in Fig. 8(c), the constitute law of the material adopts the bi-linear model. Ten horizontal forces along the x-axis are applied to the structure, which are shown in Fig. 8(a) and 8(b). The performance function is defined as:

$$Y = g(\boldsymbol{X}) = \Delta - U(P_1, P_2, P_3, P_4, P_5, E, A, b, F_y), \tag{41}$$

where $U(\cdot)$ denotes the horizontal displacement at the top of the structure along the x-axis, which is a function of nine random variables (see Table 6); Δ is the threshold of U, specified as 50 mm in this study.

In this example, the reference failure probability is 6.25×10^{-4} with COV being 1.26%, which given by MCS with 10^7 samples. As summarized in Table 7, the proposed method is compared with several other methods, i.e., AK-MCS+U [20], ALPI [22], and ALR [35]. One can see that AK-MCS+U, ALPI and the proposed method can produce fairly good average failure probability estimates with small COVs (say less than 4%). However, the ALR method produces biased results for $k = 5, 10, 15$. When it comes to the computational efficiency, the proposed method outperforms other methods in terms of the average number of iterations N_{iter}. In addition, when k is small (e.g., 5), the average total number of calls N_{call} required by PA-BFPL is also slightly less than that of ALPI, and far less than that of AK-MCS+U and ALR.

4.5. Final remarks

As can be seen from the above numerical studies, the parameter k greatly affects the performance of the

25

Table 5: Reliability results of Example 3 by different methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV[$\hat{P}_f$]/%
MCS		-	10^7	5.80×10^{-3}	0.41
AK-MCS+U		$1 + 68.00 = 69.00$	$12 + 68.00 = 80.00$	5.76×10^{-3}	1.74
ALPI		$1 + 40.50 = 41.50$	$12 + 40.50 = 42.50$	5.71×10^{-3}	1.56
	$k = 5$	$1 + 17.10 = 18.10$	$10 + 85.50 = 95.50$	5.97×10^{-3}	1.79
ALR in UQLab 2.0.0	$k = 10$	$1 + 10.20 = 11.20$	$10 + 102.00 = 112.00$	5.90×10^{-3}	3.08
	$k = 15$	$1 + 6.90 = 7.90$	$10 + 103.50 = 113.50$	5.94×10^{-3}	2.08
	$k = 5$	$1 + 5.40 = 6.40$	$10 + 27.00 = 37.00$	5.71×10^{-3}	1.05
Proposed PA-BFPL	$k = 10$	$1 + 3.70 = 4.70$	$10 + 37.00 = 47.00$	5.74×10^{-3}	1.50
	$k = 15$	$1 + 3.00 = 4.00$	$10 + 45.00 = 55.00$	5.77×10^{-3}	0.97

proposed PA-BFPL method, especially for efficiency. Typically, the average number of iterations decreases, while the average number of g-function evaluations when k increases from 5 to 15. However, this should not be regarded as a general conclusion because we only investigated three cases for k in each example. As a rule of thumb, one can choose a small k for non-parallel computing, whereas a large k when parallel computing is available.

5. Numerical investigation on the posterior distribution of failure probability

In addition to the posterior mean and variance, the posterior distribution of failure probability could be of interest for a complete Bayesian framework. For example, one can offer a confidence interval for the failure probability when the posterior distribution is available. However, it cannot be obtained analytically according to its definition (Eq. (22)). In this section, we attempt to numerically investigate the posterior distribution of failure probability through the four numerical examples given in the preceding section.

According to the proposed VAIS method, Eq. (22) can be rewritten as follows:

$$\hat{P}_{f,n}(\varpi) = \int_{\mathcal{X}} \hat{I}_n(\varpi, \boldsymbol{x}) \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{h_{\boldsymbol{X}}(\boldsymbol{x})} h_{\boldsymbol{X}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{42}$$

The reformulation actually allows us to evaluate the above integral numerically as:

$$\hat{P}_{f,n}(\varpi) \approx \frac{1}{N_4} \sum_{i=1}^{N_4} \hat{I}_n(\varpi, \boldsymbol{x}^{(i)}) \frac{f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})}{h_{\boldsymbol{X}}(\boldsymbol{x}^{(i)})} h_{\boldsymbol{X}}(\boldsymbol{x}^{(i)}), \tag{43}$$
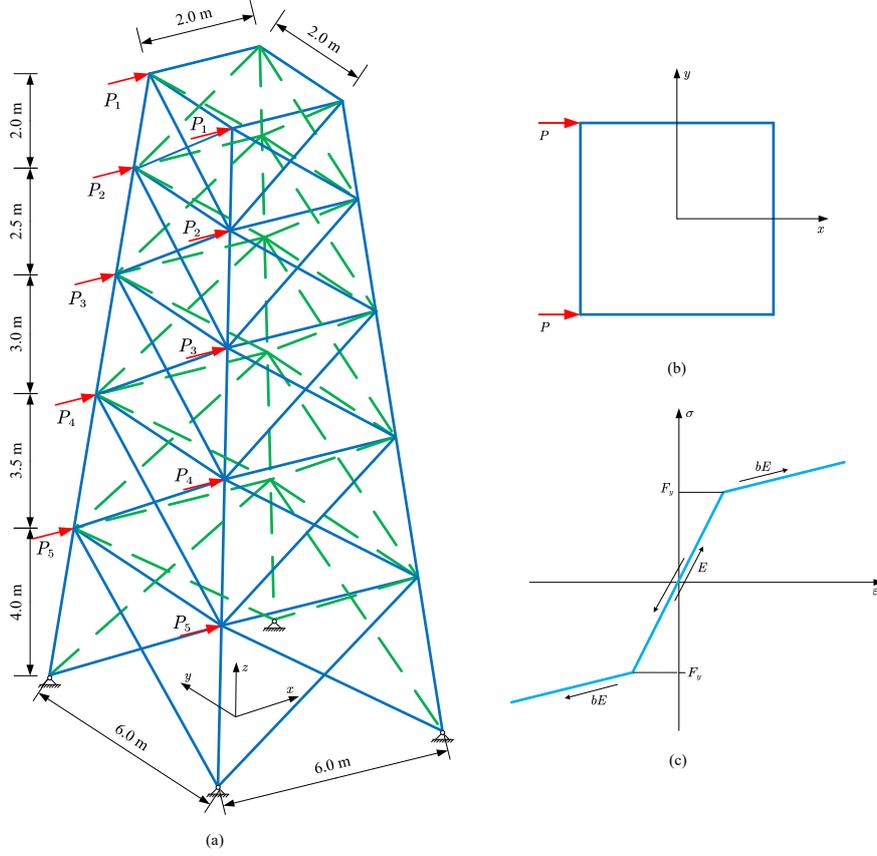
Figure 8: A transmission tower subject to horizontal loads: (a) 3-D finite element model; (b) Schematic diagram of load direction; (c) Schematic diagram of bi-linear constitute law.

where $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N_4}$ is a set of N_4 random samples generated according to $h_{\boldsymbol{X}}(\boldsymbol{x})$. Given that N_4 is sufficiently large, we can approximately generate random numbers for the posterior failure probability $\hat{P}_f(\varpi)$ via Eq. (43). The key is to sample from correlated Bernoulli random variables $\left\{\hat{I}_n(\varpi, \boldsymbol{x}^{(1)}), \hat{I}_n(\varpi, \boldsymbol{x}^{(2)}), \cdots, \hat{I}_n(\varpi, \boldsymbol{x}^{(N_4)})\right\}$ defined by Eq. (17). Nevertheless, this task is still challenging, especially when N_4 is large. For simplicity, the Bernoulli random variables are assumed to be independent and it is shown from some numerical experiments that this assumption does not affect our finial conclusion. Under these settings, posterior failure probability samples can be generated at each step of the proposed PA-BFPL method. To limit the length of the paper, only the results from the last step of the proposed method (an exemplary run, $k = 10$) are reported for those four numerical examples. The number of posterior failure probability samples is set to be 10^3. Other parameters are specified as: $\alpha = 2$ and $N_4 = 5 \times 10^5$.

The results of the normality tests for the simulated data of the posterior failure probabilities are depicted

27

Table 6: Random variables of Example 4.

Variable	Description	Distribution	Mean	COV
P_1/kN	Horizontal load	Lognormal	100	0.20
P_2/kN	Horizontal load	Lognormal	80	0.20
P_3/kN	Horizontal load	Lognormal	60	0.20
P_4/kN	Horizontal load	Lognormal	40	0.20
P_5/kN	Horizontal load	Lognormal	20	0.20
E/GPa	Young's modulus	Normal	200	0.15
A/mm^2	Cross-sectional area	Normal	5000	0.15
F_y/MPa	Yield stress	Normal	400	0.15
b	Strain-hardening ratio	Normal	0.02	0.10

502 in Fig. 9. It is shown that the posterior failure probability samples can be well-modelled by normal
503 distributions for all the cases studied. The results indicate that the posterior distribution of the failure
504 probability might be approximated by a Gaussian distribution $\mathcal{N}(m_{\hat{P}_{f,n}}, \sigma^2_{\hat{P}_{f,n}})$.

Table 7: Reliability results of Example 4 by different methods.

Method		N_{iter}	N_{call}	$\hat{P}_f$	COV$[\hat{P}_f]$/%
MCS		-	10^7	6.25×10^{-4}	1.26
AK-MCS+U		$1 + 68.00 = 69.00$	$12 + 113.80 = 125.80$	6.17×10^{-4}	1.83
ALPI		$1 + 46.60 = 47.60$	$12 + 46.60 = 58.60$	6.12×10^{-4}	4.28
ALR in UQLab 2.0.0	$k = 5$	$1 + 37.30 = 38.30$	$10 + 186.50 = 196.50$	2.27×10^{-3}	225.88
	$k = 10$	$1 + 18.80 = 19.80$	$10 + 188.00 = 198.00$	2.50×10^{-2}	215.52
	$k = 15$	$1 + 12.50 = 13.50$	$10 + 187.50 = 197.50$	7.20×10^{-4}	25.03
Proposed PA-BFPL	$k = 5$	$1 + 6.80 = 7.80$	$10 + 34.00 = 44.00$	6.32×10^{-4}	3.34
	$k = 10$	$1 + 4.90 = 5.90$	$10 + 49.00 = 59.00$	6.30×10^{-4}	2.09
	$k = 15$	$1 + 4.70 = 5.70$	$10 + 70.50 = 80.50$	6.25×10^{-4}	1.78

Note: For most runs, the ALR method cannot converge even for $N_{call} > 200$. For this reason, the maximum value of N_{call} is set to be 200 for $k = 5, 10$, while 205 for $k = 15$.
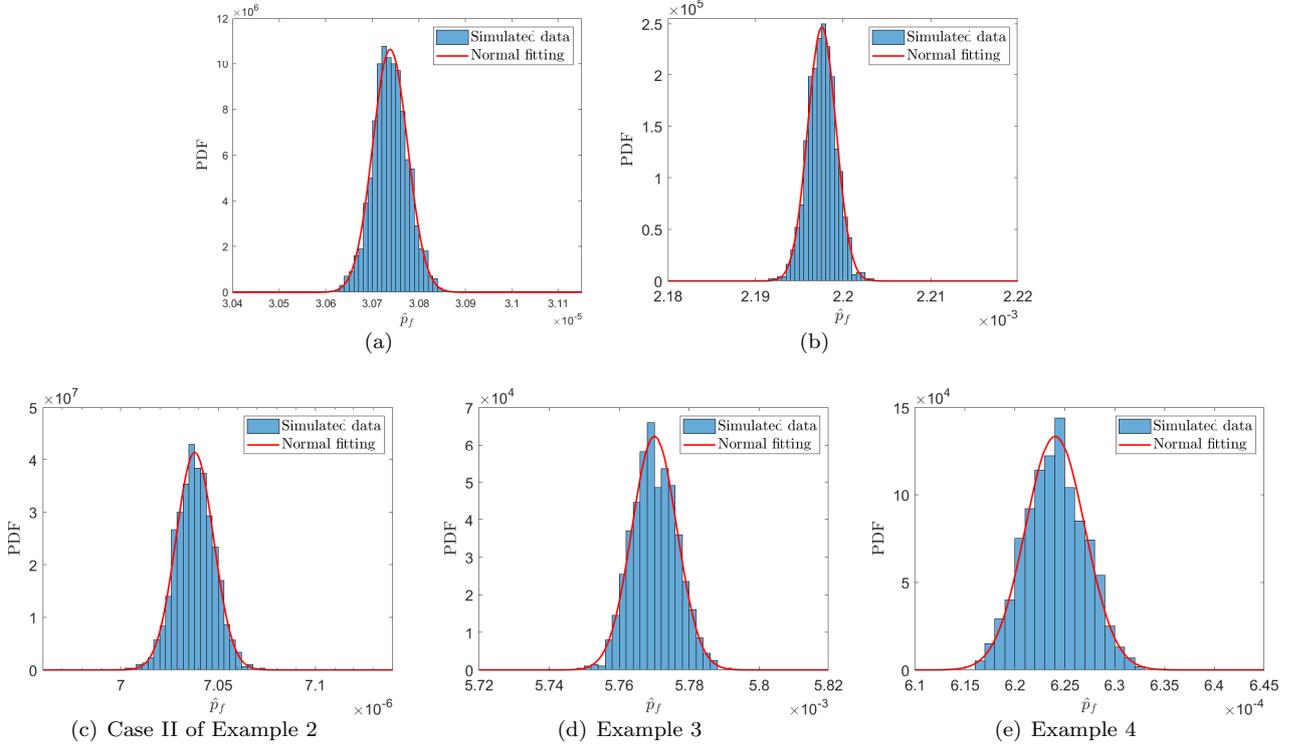
Figure 9: Normality tests of the simulated data from the posterior failure probabilities: (a) Example 1; (b) Case I of Example 2; (c) Case II of Example 2; (d) Example 3; (e) Example 4.

6. Concluding remarks

In the present paper, the task of failure probability estimation is interpreted from a perspective of Bayesian inference, in contrast to the classical frequentist inference. The proposed Bayesian failure probability inference (BFPI) framework regards the discretization error as a kind of epistemic uncertainty, and allows it to be properly modelled. To be specific, a prior Gaussian process is assumed for the performance function, a posterior distribution is then derived for the performance function, failure indicator function and failure probability conditional on observations arising from evaluating the performance function at a set of points. Numerical investigation indicates that the posterior failure probability could be approximated by a normal distribution. In addition, the parallel adaptive Bayesian failure probability learning (PA-BFPL) method is developed to make inference about the failure probability within the Bayesian framework in a parallel adaptive scheme. The proposed PA-BFPL enables to make the fullest possible use of prior evaluations on the performance function evaluation, and can take advantage of parallel computing. Compared to several existing methods, the proposed method shows improved performance for structural reliability analysis regarding robustness, accuracy and efficiency. The advantage in computational efficiency is significant especially when parallel computing facilities are available.

The proposed PA-BFPL method is supposed to work well in linear, weakly nonlinear and moderately nonlinear problems with up to medium-dimensional random variables. For highly nonlinear and/or high-dimensional problems, additional research efforts are still needed in the future.

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

References

[1] 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.

[2] R. Y. Rubinstein, D. P. Kroese, Simulation and the Monte Carlo method, John Wiley & Sons, 2016.

[3] A. B. Owen, Comment: Unreasonable effectiveness of Monte Carlo, Statistical Science 34 (1) (2019) 29–33. doi:https://doi.org/10.1214/18-STS676.

[4] A. O'Hagan, Monte carlo is fundamentally unsound, The Statistician (1987) 247–249.

[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] S.-K. Au, Y. Wang, Engineering risk assessment with subset simulation, John Wiley & Sons, 2014.

[7] 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.

[8] Z. Wang, J. Song, Cross-entropy-based adaptive importance sampling using von Mises-Fisher mixture for high dimensional reliability analysis, Structural Safety 59 (2016) 42–52. doi:https://doi.org/10.1016/j.strusafe.2015.11.002.

[9] S. Geyer, I. Papaioannou, D. Straub, Cross entropy-based importance sampling using Gaussian densities revisited, Structural Safety 76 (2019) 15–27. doi:https://doi.org/10.1016/j.strusafe.2018.07.001.

[10] 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.

[11] P. Diaconis, Bayesian Numerical Analysis, Statistical Decision Theory and Related Topics, IV, Vol. 1 (West Lafayette, Ind., 1986), 1988.

[12] A. O'Hagan, Bayes–Hermite quadrature, Journal of Statistical Planning and Inference 29 (3) (1991) 245–260. doi:https://doi.org/10.1016/0378-3758(91)90002-V.

[13] C. E. Rasmussen, Z. Ghahramani, Bayesian Monte Carlo, Advances in Neural Information Processing Systems (2003) 505–512.

[14] P. Pandita, I. Bilionis, J. Panchal, Bayesian optimal design of experiments for inferring the statistical expectation of expensive black-box functions, Journal of Mechanical Design 141 (10) (2019). doi:https://doi.org/10.1115/1.4043930.

[15] P. Wei, X. Zhang, M. Beer, Adaptive experiment design for probabilistic integration, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113035. doi:https://doi.org/10.1016/j.cma.2020.113035.

[16] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, D. Sejdinovic, Probabilistic integration: a role in statistical computation?, Statistical Science 34 (1) (2019) 1–22. doi:https://doi.org/10.1214/18-STS660.

[17] 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.

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

[19] 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.

[20] 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`.

[21] F. Cadini, A. Gioletta, A Bayesian Monte Carlo-based algorithm for the estimation of small failure probabilities of systems affected by uncertainties, Reliability Engineering & System Safety 153 (2016) 15–27. `doi:https://doi.org/10.1016/j.ress.2016.04.003`.

[22] 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`.

[23] P. I. Frazier, A tutorial on Bayesian optimization, arXiv preprint arXiv:1807.02811 (2018).

[24] J. Lee, Generalized Bernoulli process with long-range dependence and fractional binomial distribution, Dependence Modeling 9 (1) (2021) 1–12. `doi:https://doi.org/10.1515/demo-2021-0100`.

[25] M. T. Sichani, S. R. Nielsen, C. Bucher, Applications of asymptotic sampling on high dimensional structural dynamic problems, Structural Safety 33 (4-5) (2011) 305–316. `doi:https://doi.org/10.1016/j.strusafe.2011.05.002`.

[26] S. Sun, X. Li, H. Liu, K. Luo, B. Gu, Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34 (7) (2015) 1096–1109. `doi:https://doi.org/10.1109/TCAD.2015.2404895`.

[27] M. Rashki, Hybrid control variates-based simulation method for structural reliability analysis of some problems with low failure probability, Applied Mathematical Modelling 60 (2018) 220–234. `doi:https://doi.org/10.1016/j.apm.2018.03.009`.

[28] K. Cheng, Z. Lu, S. Xiao, J. Lei, Estimation of small failure probability using generalized subset simulation, Mechanical Systems and Signal Processing 163 (2022) 108114. `doi:https://doi.org/10.1016/j.ymssp.2021.108114`.

[29] Y. Ibrahim, Observations on applications of importance sampling in structural reliability analysis, Structural Safety 9 (4) (1991) 269–281. `doi:https://doi.org/10.1016/0167-4730(91)90049-F`.

[30] S.-K. Au, J. Beck, Important sampling in high dimensions, Structural Safety 25 (2) (2003) 139–163. `doi:https://doi.org/10.1016/S0167-4730(02)00047-4`.

[31] M.-H. Chen, Q.-M. Shao, J. G. Ibrahim, Monte Carlo methods in Bayesian computation, Springer Science & Business Media, 2012.

[32] D. J. MacKay, D. J. Mac Kay, Information theory, inference and learning algorithms, Cambridge University Press, 2003.

[33] 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. `doi:https://doi.org/10.1177/1748006X19844666`.

[34] S. Marelli, B. Sudret, Uqlab: A framework for uncertainty quantification in Matlab, in: Vulnerability, uncertainty, and risk: quantification, mitigation, and management, 2014, pp. 2554–2563. `doi:https://doi.org/10.1061/9780784413609.257`.

[35] M. Moustapha, S. Marelli, B. Sudret, UQLab user manual – Active learning reliability, Tech. rep., Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, report UQLab-V2.0-117 (2022).

[36] K. Breitung, The geometry of limit state function graphs and subset simulation: Counterexamples, Reliability Engineering

& System Safety 182 (2019) 98–106. `doi:https://doi.org/10.1016/j.ress.2018.10.008`.

[37] Z. Wen, H. Pei, H. Liu, Z. Yue, A sequential Kriging reliability analysis method with characteristics of adaptive sampling regions and parallelizability, Reliability Engineering & System Safety 153 (2016) 170–179. `doi:https://doi.org/10.1016/j.ress.2016.05.002`.

[38] X. Huang, Y. Zhang, Reliability–sensitivity analysis using dimension reduction methods and saddlepoint approximations, International Journal for Numerical Methods in Engineering 93 (8) (2013) 857–886. `doi:https://doi.org/10.1002/nme.4412`.

[39] C. Dang, P. Wei, M. G. Faes, M. A. Valdebenito, M. Beer, Interval uncertainty propagation by a parallel bayesian global optimization method, Applied Mathematical Modelling 108 (2022) 220–235. `doi:https://doi.org/10.1016/j.apm.2022.03.031`.