# Estimation of response expectation bounds under parametric p-boxes by combining Bayesian global optimization with unscented transform

Chen Ding[1], Chao Dang[2], Matteo Broggi[3], and Michael Beer[4]

[1]Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany. Email: chen.ding@irz.uni-hannover.de

[2]Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany (corresponding author). Email: chao.dang@irz.uni-hannover.de

[3]Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany. Email: broggi@irz.uni-hannover.de

[4]Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany; Institute of Risk and Uncertainty, University of Liverpool, Peach St., Liverpool L69 7ZF, UK; International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, China. Email: beer@irz.uni-hannover.de

## ABSTRACT

In engineering analysis, propagating parametric probability boxes (p-boxes) remains a challenge since a computationally expensive nested solution scheme is involved. To tackle this challenge, this paper proposes a novel optimization-integration method to propagate parametric p-boxes, mainly focusing on estimating the lower and upper bounds of structural response expectation for linear and moderately nonlinear problems. A model-based optimization scheme, named Bayesian global optimization, is first introduced to explore the space of distribution parameters. Subsequently, an efficient numerical integration method, named unscented transform, is employed to estimate the response expectation with a given set of distribution parameters. Compared to existing optimization-integration methods, the proposed method has three advantages. First, the response expectation

bounds are successively estimated, allowing for the reuse of samples generated from the lower bound estimation in the upper bound estimation. Second, the approximation error introduced by the numerical integration method is considered. Third, computational efficiency in both the optimization and integration processes is improved. Four applications are investigated to validate the effectiveness of the proposed method, showing its ability to balance computational efficiency and accuracy when evaluating response expectation bounds.

**KEYWORD**

Imprecise probability propagation, Parametric probability box, Response expectation bounds, Bayesian global optimization, Unscented transform, Gaussian process

## INTRODUCTION

Practical engineering problems are often rife with aleatory uncertainties that are irreducible and stem from random nature, such as the uncertainties in material properties, external loads, operating environments and etc. In many cases, the information for describing such uncertainties can be insufficient, ambiguous, fragmentary, or indeterminate (Beer et al. 2013). In this regard, epistemic uncertainties that result from a lack of knowledge or information should also be considered. Such mixed uncertainty can be characterized by the imprecise probability model such as the parametric probability box (p-box) model (Ferson and Hajagos 2004; Faes et al. 2021). For a parametric p-box model, aleatory uncertainty is represented by a set of probability distributions with known distribution types, while epistemic uncertainty is reflected by the imprecise distribution parameters that can be described by intervals. In order to reflect the influence of input imprecise probabilities on structural responses, imprecise probability propagation is of great significance in engineering structural analysis.

In general, the state-of-the-art methods for propagating parametric p-boxes can be classified into two categories: double-loop methods and single-loop methods. As a straightforward approach, the double-loop method treats the epistemic and aleatory uncertainties through a nested loop structure. Double-loop Monte Carlo simulation (DLMCS) (Bruns and Paredis 2006) samples different sets of distribution parameters in the outer loop, and for each distribution parameter set, Monte Carlo simulation (MCS) is performed to estimate the output of interest in the inner loop. DLMCS works regardless of nonlinear properties and dimensionalities of the problem at hand. However, it is quite computationally expensive, since both loops require a considerably large number of samples in order to ensure the estimation accuracy of the output of interest. Although some improved DLMCS methods, such as interval Monte Carlo method (Zhang et al. 2010; Zhang et al. 2012) and the vertex-based DLMCS (Vertex-MCS) (Xiao et al. 2016), are developed to reduce the number of samples in total, their scopes of application and efficiency are limited.

To improve the computational efficiency, an outer-loop optimization can be adopted, where imprecise distribution parameters are treated as design variables to be optimized and the lower and upper bounds on the output of interest are regarded as two separate optimization objectives. In the inner loop, the output of interest at a certain design point can be estimated by aleatory uncertainty propagation methods. Take the example of capturing the bounds on a response expectation, numerical integration methods can be adopted in the inner loop to evaluate the response expectation under fixed distribution parameters. The integration of outer-loop optimization and inner-loop numerical integration methods can be collectively referred to as optimization-integration methods. Typical optimization-integration methods are the optimized parameter sampling (OPS) (Bruns and Paredis 2006; Bruns 2006), optimized univariate dimension-reduction method (OUDRM) (Liu et al. 2018) and optimized sparse grid numerical integration method (OSGNI) (Liu et al. 2019). Such existing methods rely on gradient-based optimizers, which can easily converge to a local optimum. In this sense, the resulting response expectation bounds may be underestimated. Although some global optimization algorithms, like the genetic algorithm (Pedroni and Zio 2015), can help mitigate this issue, the optimization process requires a large number of objective function calls, which can be time-consuming when dealing with objective functions that are expensive to evaluate. In the inner loop, some efficient numerical integration methods, such as univariate dimension-reduction method (Liu et al. 2018) and sparse grid numerical integration (Liu et al. 2019), are employed. Nevertheless, the computational efficiency within the inner loop can be further enhanced, especially for linear and moderately nonlinear problems. Additionally, existing optimization-integration methods do not account for the approximation error introduced by numerical integration methods, potentially resulting in inaccuracies in the derived response expectation bounds. Furthermore, these methods are unable to fully leverage the data generated by the lower or upper estimation that has already been performed.

On the other hand, to alleviate the computational burden of the double-loop framework, many single-loop methods have been recently developed, such as the extended Monte Carlo simulation (EMCS) (Wei et al. 2014), non-intrusive imprecise stochastic simulation (NISS) (Wei et al. 2019), non-intrusive imprecise probabilistic integration (NIPI) (Wei et al. 2021b), collaborative and adaptive Bayesian optimization (CABO) (Wei et al. 2021a), and parallel Bayesian quadrature optimization (PBQO) (Dang et al. 2022a). Note that existing single-loop methods typically rely on constructing an augmented uncertainty space consisting of both aleatory and epistemic uncertainties, which increases the dimensionality to be dealt with. Although some single-loop methods such as CABO and PBQO may require less response function calls compared with double-loop methods, it becomes difficult to estimate the output of interest for problems with high-dimensional augmented uncertainty space. Besides, EMCS and NISS are not capable for propagating parametric p-boxes with distribution parameters that are supported in a wide range.

Therefore, there is still a need to develop a method for parametric p-box propagation with not only reasonable accuracy and efficiency, but also fine applicability. The main focus of this paper is on capturing the response expectation bounds that reflect the effect of epistemic uncertainty on the statistical characteristics of the response. A new optimization-integration method is presented to estimate the response expectation bounds, especially for linear and moderately nonlinear problems. The proposed method combines two advanced and efficient strategies to greatly reduce the computational efforts. Specifically speaking, to facilitate the optimization process, a model-based optimization scheme, named Bayesian global optimization (BGO) (Jones et al. 1998), is employed. By using the BGO, the original expensive-to-evaluate objective function can be predicted by an cheap-to-evaluate Bayesian model. To consider the effects of approximation errors introduced by numerical integration in estimating the response expectation, noisy Gaussian process (GP) model is adopted in this study. Such GP model can be updated adaptively according to an effective improvement strategy, so as to obtain the globally effective optima, i.e., the optimal distribution parameters corresponding to the response expectation bounds, with fewer computational efforts. When estimating response expectation on the set of distribution parameters obtained from the optimization process, a highly efficient numerical integration method, called unscented transform (UT) (Julier and Uhlmann 1997b; Wan and Van Der Merwe 2000; Jia et al. 2013), is implemented. UT is able to provide estimated results up to third degree of algebraic accuracy, which should be acceptable for linear and moderately nonlinear problems. Besides, the number of simulations grows only linearly with the dimension of p-box variables. It is worth mentioning that compared with existing optimization-integration methods, the proposed method takes into account the approximation errors brought by UT. Moreover, the proposed method estimates the lower and upper response expectation bounds in a sequential manner, where the samples generated for the lower bound evaluation can be further reused for the upper bound evaluation to reduce unnecessary waste of computational efforts.

The remaining of the paper is organized as follows: Section "Problem statement" introduces the mathematical formulation of the response expectation bounds considering input variables described by parametric p-boxes. Section "Proposed method" presents the proposed optimization-integration approach that combines the BGO with the UT. Section "Test examples" investigates four test examples to illustrate the feasibility of the proposed method. Conclusions are given in section "Concluding remarks".

To enhance readability, a list of acronyms used in this paper is provided in Table 1.

## PROBLEM STATEMENT

Consider a response function that describes the input-output relationship of a structural system as $Y = g(\boldsymbol{X})$. Here, $g(\cdot)$ represents a deterministic, continuous and real-valued mapping function; $\boldsymbol{X} = \left\{X_1, X_2, ..., X_{n_s}\right\}$ is an $n_s$ dimensional input vector of variables, where each variable is characterized by a parametric p-box model; $Y$ denotes a scalar output of interest, which is also a p-box variable. Let us denote $\boldsymbol{\theta} = \left(\theta_1, \theta_2, ..., \theta_{n_\theta}\right)$ as $n_\theta$-dimensional imprecise distribution parameter vector. The epistemic uncertainty in $\boldsymbol{\theta}$ is represented by a hyperrectangle, i.e., $\boldsymbol{\theta} = \left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]$, where $\underline{\boldsymbol{\theta}} = \left(\underline{\theta}_1, \underline{\theta}_2, ..., \underline{\theta}_{n_\theta}\right)$ denotes the lower bound and $\bar{\boldsymbol{\theta}} = \left(\bar{\theta}_1, \bar{\theta}_2, ..., \bar{\theta}_{n_\theta}\right)$ is the upper bound. Then, the joint probability density function (PDF) of $\boldsymbol{X}$ can be represented by $f\left(\boldsymbol{x}|\boldsymbol{\theta}\right)$. For convenience, all variables in $\boldsymbol{X}$ and distribution parameters in $\boldsymbol{\theta}$ are assumed to be mutually independent.

Under the above setting, the probability distribution and any statistical moments of $Y$ are also functions of $\boldsymbol{\theta}$. Taking the expectation of $Y$ as an example, it can be written as:

$$m\left(\boldsymbol{\theta}\right) = \int_{\mathbb{R}^{n_s}} g\left(\boldsymbol{x}\right) f\left(\boldsymbol{x}|\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $m\left(\boldsymbol{\theta}\right)$ represents the expectation of $Y$, the value of which depends on the value of $\boldsymbol{\theta}$. It is noted that for some practical engineering applications, the analysts may be more interested in obtaining the bounds on $m\left(\boldsymbol{\theta}\right)$ than in obtaining expressions for the response expectation function over the entire domain of $\boldsymbol{\theta}$. This is because the response expectation bounds enable to provide a possible range reflecting the effect of epistemic uncertainty on expectation of $Y$ (Wei et al. 2021a). In addition, evaluating the bounds on response expectation can be much more easier than capturing the overall behavior of $m\left(\boldsymbol{\theta}\right)$ over the full domain of $\boldsymbol{\theta}$. In this regard, this paper focuses on the estimation of the response expectation bounds.

The lower and upper bounds of response expectation can be obtained by finding the minimal and maximal values of $m\left(\boldsymbol{\theta}\right)$ within the hyperrectangle $\left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]$, which can be expressed as:

$$\underline{m} = \min_{\boldsymbol{\theta} \in \left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]} m\left(\boldsymbol{\theta}\right), \tag{2}$$

$$\overline{m} = \max_{\boldsymbol{\theta} \in \left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]} m\left(\boldsymbol{\theta}\right), \tag{3}$$

where $\underline{m}$ and $\overline{m}$ denote the lower and upper bounds of $m\left(\boldsymbol{\theta}\right)$, respectively. In the following, the optimal distribution parameter corresponding to $\underline{m}$ is denoted as $\boldsymbol{\theta}_{\text{min}}$, and the optimal distribution parameter corresponding to $\overline{m}$ is represented as $\boldsymbol{\theta}_{\text{max}}$.

Note that for most cases, the analytical expression of $m(\boldsymbol{\theta})$ is usually difficult and even impossible to obtain due to underlying complexity of Eq. (1). Hence, directly finding analytic solutions for the minimum and maximum values according to Eqs. (2)-(3) is rather difficult. Alternatively, we can resort to the optimization-integration method such that $\boldsymbol{\theta}$ can be first searched by optimization and the response expectation at a certain $\boldsymbol{\theta}$ found during the optimization process is then estimated based on a numerical integration method. Since the response function involved in Eq. (1) is usually a black-box function that is expensive to evaluate, a computationally efficient optimization-integration method that calls the response function as few as possible is highly desired. At the same time, such method should also enable to provide estimated response expectation bounds with acceptable accuracy. To achieve this aim, a novel optimization-integration method will be developed in the following.

## PROPOSED METHOD

In this section, a new optimization-integration method is developed to estimate the lower and upper bounds of $m(\boldsymbol{\theta})$ with reasonable accuracy and efficiency, where $\underline{m}$ and $\overline{m}$ are separately estimated one after the other. Note that the proposed method is able to make full use of the available information such that data obtained from the lower bound estimation can be further reused in the upper bound estimation, and thus avoiding unnecessary computational effort. Specifically, a model-based optimization method, named Bayesian global optimization (Jones et al. 1998), is employed in order to explore the space of distribution parameters. At each $\boldsymbol{\theta}$ found during the optimization process, one highly efficient numerical integration method, named unscented transform (Julier and Uhlmann 1997b; Wan and Van Der Merwe 2000; Jia et al. 2013), is introduced to evaluate $m(\boldsymbol{\theta})$.

### Bayesian global optimization

By making use of BGO, our basic idea is to assume a Bayesian model to $m(\boldsymbol{\theta})$, and then update the Bayesian model successively with additional observations according to an efficient infill sampling criterion (Jones et al. 1998; Dang et al. 2022b). Such infill sampling criterion enables to fully exploit the available observations, and strike a good tradeoff between exploitation and exploration for the selection of the new updating observations. In the following, the Bayesian model and infill sampling criterion adopted in the optimization process of proposed optimization-integration method are introduced in detail.

*Gaussian process model*

Following a Bayesian approach, the expensive-to-evaluate response expectation function can be treated with a Bayesian model. Commonly, Gaussian process (GP) model (Williams and Rasmussen 2006) is adopted in the BGO. Note that for most realistic modeling situations, we cannot obtain the true value of $m(\boldsymbol{\theta})$, but only a noisy version of $m(\boldsymbol{\theta})$. In this regard, we assume that the noisy version of $m(\boldsymbol{\theta})$, denoted as $\hat{m}(\boldsymbol{\theta})$, is equal to the true response expectation function $m(\boldsymbol{\theta})$ plus an additional noise $\epsilon$ such that $\hat{m}(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) + \epsilon$, where $\epsilon$ is assumed to follow a zero-mean Gaussian distribution with variance $\sigma_\epsilon^2$. The true response expectation $m(\boldsymbol{\theta})$ is assigned a GP prior such that $m_0(\boldsymbol{\theta}) \sim \mathrm{GP}(\beta(\boldsymbol{\theta}), \kappa(\boldsymbol{\theta}, \boldsymbol{\theta}'))$, where $\beta(\boldsymbol{\theta})$ and $\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are the prior mean and covariance (also called kernel) functions, respectively. There are many different forms of prior mean and covariance functions, which can be found in Ref. (Williams and Rasmussen 2006). In this work, a constant prior mean is adopted such that $\beta(\boldsymbol{\theta}) = \beta_0$ and $\beta_0 \in \mathbb{R}$. The squared exponential kernel function is employed here, which can be expressed as:

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_0^2 \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}')^{\mathrm{T}}\right), \tag{4}$$

where $\sigma_0^2$ denotes the overall variance; $\boldsymbol{\Sigma} = \mathrm{diag}\left(l_1^2, l_2^2, ..., l_{n_\theta}^2\right)$ is a diagonal matrix and $l_i, i = 1, ..., n_{\boldsymbol{\theta}}$ is the length scale in the $i$-th dimension. Under these settings, a total of $n_{\boldsymbol{\theta}} + 3$ free parameters are involved inside the GP model, which are referred to as the hyperparameters $\boldsymbol{\psi} = \left\{\beta_0, \sigma_0, \sigma_\epsilon, l_1, ..., l_{n_\theta}\right\}$ and can be inferred from a set of observations.

Suppose that we have obtained $\mathcal{N}$ noisy observations. Denote such training dataset as $\mathcal{D} = \left\{\boldsymbol{\Theta}, \hat{\mathcal{M}}(\boldsymbol{\Theta})\right\}$, where $\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}^{(1)}; \boldsymbol{\theta}^{(2)}; ...; \boldsymbol{\theta}^{(\mathcal{N})}\right\}$ is a sample matrix with size $(\mathcal{N} \times n_\theta)$, and $\hat{\mathcal{M}}(\boldsymbol{\Theta}) = \left\{\hat{m}\left(\boldsymbol{\theta}^{(1)}\right), ..., \hat{m}\left(\boldsymbol{\theta}^{(\mathcal{N})}\right)\right\}^{\mathrm{T}}$ is an $(\mathcal{N} \times 1)$ response expectation vector whose components have been evaluated. Based on the training dataset $\mathcal{D}$, the hyperparameters can be optimally determined by maximizing the log marginal likelihood function (Williams and Rasmussen 2006):

$$\boldsymbol{\psi}^{\star} = \arg\max_{\boldsymbol{\psi}} \left(\log \left(p \left(\hat{\mathcal{M}} \left(\boldsymbol{\Theta}\right) | \boldsymbol{\Theta}, \boldsymbol{\psi}\right)\right)\right), \tag{5}$$

where

$$\log \left(p \left(\hat{\mathcal{M}} \left(\boldsymbol{\Theta}\right) | \boldsymbol{\Theta}, \boldsymbol{\psi}\right)\right) = -\frac{1}{2} \left(\hat{\mathcal{M}} \left(\boldsymbol{\Theta}\right) - \beta_0\right)^{\mathrm{T}} \left(\boldsymbol{K} + \sigma_{\epsilon}^2 \boldsymbol{I}\right)^{-1} \left(\hat{\mathcal{M}} \left(\boldsymbol{\Theta}\right) - \beta_0\right) - \frac{1}{2} \log \left(\left|\boldsymbol{K} + \sigma_{\epsilon}^2 \boldsymbol{I}\right|\right) - \frac{\mathcal{N}}{2} \log \left(2\pi\right), \tag{6}$$

in which $\boldsymbol{K}$ is an $(\mathcal{N} \times \mathcal{N})$ covariance matrix with $(i, j)$-th entry as $\kappa\left(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(j)}\right)$; $\boldsymbol{I}$ is an $(\mathcal{N} \times \mathcal{N})$ identity matrix. For more details, the interested readership may refer to Ref. (Williams and Rasmussen 2006).

Once the hyperparameters are determined, a posterior distribution of $m\left(\boldsymbol{\theta}\right)$ can be obtained by conditioning on $\mathcal{D}$. At a new observation $\boldsymbol{\theta}$, the posterior of $m\left(\boldsymbol{\theta}\right)$ follows a normal distribution such that $m_{\mathcal{N}}\left(\boldsymbol{\theta}\right) \sim \mathrm{N}\left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right), \sigma_{\mathcal{N}}^2\left(\boldsymbol{\theta}\right)\right)$. Here, the posterior mean $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right)$ is employed as the predictor of response expectation in the optimization process, and the posterior variance $\sigma_{\mathcal{N}}^2\left(\boldsymbol{\theta}\right)$ is the measure of prediction uncertainty. $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right)$ and $\sigma_{\mathcal{N}}^2\left(\boldsymbol{\theta}\right)$ can be expressed in closed form:

$$\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right) = \beta\left(\boldsymbol{\theta}\right) + \boldsymbol{\kappa}\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right) \left(\boldsymbol{K} + \sigma_{\epsilon}^2 \boldsymbol{I}\right)^{-1} \left(\hat{\mathcal{M}}\left(\boldsymbol{\Theta}\right) - \beta\left(\boldsymbol{\Theta}\right)\right), \tag{7}$$

$$\sigma_{\mathcal{N}}^2\left(\boldsymbol{\theta}\right) = \kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}\right) - \boldsymbol{\kappa}\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right) \left(\boldsymbol{K} + \sigma_{\epsilon}^2 \boldsymbol{I}\right)^{-1} \boldsymbol{\kappa}\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right)^{\mathrm{T}}, \tag{8}$$

in which $\boldsymbol{\kappa}\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right)$ is a $(1 \times \mathcal{N})$ covariance vector between $\boldsymbol{\theta}$ and $\boldsymbol{\Theta}$, and its $i$-th component is $\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}\right)$; $\beta\left(\boldsymbol{\Theta}\right)$ is an $(\mathcal{N} \times 1)$ expectation vector with $i$-th component as $\beta\left(\boldsymbol{\theta}^{(i)}\right)$.

*Expected improvement criterion*

To infer the response expectation bounds $\underline{m}$ and $\overline{m}$ from fewer training samples, an efficient infill sample strategy combined with the GP model is desired. Note that the aim of such strategy is to find the promising points where to evaluate the objective function by extracting as much as possible knowledge from the current posterior GP. Along this line, the expected improvement (EI) criterion (Jones et al. 1998) could be a good choice. By maximizing the EI, new update points can be selected by exploiting the best existing solutions from the GP model and exploring the undeveloped design space that may contain potential optima. In this regard, here we adopt the EI criterion to find $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$ that respectively corresponding to $\underline{m}$ and $\overline{m}$, where the lower bound $\underline{m}$ is estimated first and the upper bound $\overline{m}$ is evaluated subsequently.

**EI criterion for lower bound optimization** Let $\boldsymbol{\theta}^{\star}_{\min} = arg\min_{1\leqslant j\leqslant N}\left\{\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)\right\}$ be the current best solution to lower expectation bound $\underline{m}$ obtained from the training dataset $\mathcal{D}$. We are aiming to search for a new sample point $\boldsymbol{\theta}$ that enables to bring about an improvement beyond the current lower response expectation bound at point $\boldsymbol{\theta}^{\star}_{\min}$. The expectation of such improvement conditional on $\boldsymbol{\Theta}$, denoted as $\mathcal{L}^{\mathrm{EI}}_{\min}(\boldsymbol{\theta})$, can be expressed as (Jones et al. 1998):

$$\mathcal{L}^{\mathrm{EI}}_{\min}(\boldsymbol{\theta}) = \mathbb{E}\left[\max\left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) - \mu_{\mathcal{N}}(\boldsymbol{\theta}), 0\right)\right] = \begin{cases} \mathbb{E}\left[\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) - \mu_{\mathcal{N}}(\boldsymbol{\theta})\right], \text{if } \mu_{\mathcal{N}}(\boldsymbol{\theta}) < \mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) \\ 0, \text{ otherwise} \end{cases}, \tag{9}$$

where $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) = \min_{1\leqslant j\leqslant N}\left\{\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)\right\}$. The analytical expression of the above EI function can be derived as (Jones et al. 1998):

$$\mathcal{L}^{\mathrm{EI}}_{\min}(\boldsymbol{\theta}) = \left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) - \mu_{\mathcal{N}}(\boldsymbol{\theta})\right)\Phi\left(\frac{\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) - \mu_{\mathcal{N}}(\boldsymbol{\theta})}{\sigma_{\mathcal{N}}(\boldsymbol{\theta})}\right) + \sigma_{\mathcal{N}}(\boldsymbol{\theta})\,\phi\left(\frac{\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) - \mu_{\mathcal{N}}(\boldsymbol{\theta})}{\sigma_{\mathcal{N}}(\boldsymbol{\theta})}\right), \tag{10}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ represent the cumulative distribution function (CDF) and PDF of the standard normal distribution, respectively. The new sample point, denoted as $\boldsymbol{\theta}^{+}_{\min}$, is determined by maximizing $\mathcal{L}^{\mathrm{EI}}_{\min}(\boldsymbol{\theta})$, i.e.,

$$\boldsymbol{\theta}^{+}_{\min} = \arg\max_{\boldsymbol{\theta}\in\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]} \mathcal{L}^{\mathrm{EI}}_{\min}(\boldsymbol{\theta}). \tag{11}$$

The first term in Eq. (10) prefers the point related to smaller $\mu_{\mathcal{N}}(\boldsymbol{\theta})$, while the second term in Eq. (10) prefers the sample point that has larger prediction uncertainty $\sigma_{\mathcal{N}}(\boldsymbol{\theta})$. Hence, a good tradeoff between model exploitation and exploration can be achieved by the EI criterion. Note that since

$\mathcal{L}_{\min}^{\mathrm{EI}}(\boldsymbol{\theta})$ is usually multi-modal, additional global optimization algorithms are desired to solve Eq. (11). Herein, one recently developed global optimization algorithm, called Equilibrium Optimizer (EO) algorithm (Faramarzi et al. 2020), is employed.

A stopping criterion is required here to indicate when to stop the lower bound optimization scheme. One common stopping criterion is to check whether the value of maximum EI is relatively small or not, i.e., $\max_{\boldsymbol{\theta}\in\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]} \mathcal{L}_{\min}^{\mathrm{EI}}(\boldsymbol{\theta}) < \mathcal{E}_m$, where $\mathcal{E}_m$ denotes the stopping tolerance that is usually prescribed by the users based on their requirement. Since the magnitude of $\mathcal{L}_{\min}^{\mathrm{EI}}(\boldsymbol{\theta})$ is usually unknown in advance, an improved stopping criterion that measures the relative error of the maximal EI (Huang et al. 2006) is adopted here, such as:

$$\frac{\max_{\boldsymbol{\theta}\in\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]} \mathcal{L}_{\min}^{\mathrm{EI}}(\boldsymbol{\theta})}{\max_{1\leqslant j\leqslant \mathcal{N}} \hat{m}\left(\boldsymbol{\theta}^{(j)}\right) - \min_{1\leqslant j\leqslant \mathcal{N}} \hat{m}\left(\boldsymbol{\theta}^{(j)}\right)} < \mathcal{E}_1, \tag{12}$$

where $\hat{m}\left(\boldsymbol{\theta}^{(j)}\right), 1 \leqslant j \leqslant \mathcal{N}$ is the estimated expectation in the current $\mathcal{D}$; the stopping tolerance $\mathcal{E}_1$ is suggested to take the magnitude of $0.1\% - 1\%$. If Eq. (12) is not satisfied, $\boldsymbol{\theta}_{\min}^{+}$ and the corresponding response expectation $\hat{m}\left(\boldsymbol{\theta}_{\min}^{+}\right)$ evaluated by a numerical integration method described in Section 4 are added to $\mathcal{D}$, and then a new round of lower bound optimization is implemented based on the enriched $\mathcal{D}$. To avoid possible premature convergence to suboptimal solutions, it is preferable to use a delayed judgement, i.e., to stop only when Eq. (12) is successively satisfied several times (e.g., three times).

**EI criterion for upper bound optimization** Once the lower bound optimization scheme ends, the upper bound optimization starts based on the training dataset $\mathcal{D}$ obtained from lower bound optimization. In this manner, the number of update points needed for upper bound estimation can be reduced and the current available training data can be further reused. Similarly, let $\boldsymbol{\theta}_{\max}^{\star} = arg\max_{1\leqslant j\leqslant \mathcal{N}}\left\{\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)\right\}$ be the current best solution to the upper expectation bound $\overline{m}$ observed so far. Then, the location of next evaluation $\boldsymbol{\theta}$ is determined by maximizing the EI over the current maximum posterior response expectation $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}_{\max}^{\star}\right)=\max_{1\leqslant j\leqslant\mathcal{N}}\left\{\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)\right\}$, i.e.,

$$\boldsymbol{\theta}_{\max}^{+}=\arg\max_{\boldsymbol{\theta}\in\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]}\ \mathcal{L}_{\max}^{\mathrm{EI}}\left(\boldsymbol{\theta}\right), \tag{13}$$

where $\boldsymbol{\theta}_{\max}^{+}$ denotes the new update point associated with the upper response expectation bound. The corresponding EI function, denoted as $\mathcal{L}_{\max}^{\mathrm{EI}}\left(\boldsymbol{\theta}\right)$, is defined in closed form (Dang et al. 2022b):

$$\begin{aligned}\mathcal{L}_{\max}^{\mathrm{EI}}\left(\boldsymbol{\theta}\right)&=\mathbb{E}\left[\max\left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right)-\mu_{\mathcal{N}}\left(\boldsymbol{\theta}_{\max}^{\star}\right),0\right)\right]\\&=\left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right)-\mu_{\mathcal{N}}\left(\boldsymbol{\theta}_{\max}^{\star}\right)\right)\Phi\left(\frac{\mu_{\mathcal{N}}(\boldsymbol{\theta})-\mu_{\mathcal{N}}\left(\boldsymbol{\theta}_{\max}^{\star}\right)}{\sigma_{\mathcal{N}}(\boldsymbol{\theta})}\right)+\sigma_{\mathcal{N}}\left(\boldsymbol{\theta}\right)\phi\left(\frac{\mu_{\mathcal{N}}(\boldsymbol{\theta})-\mu_{\mathcal{N}}\left(\boldsymbol{\theta}_{\max}^{\star}\right)}{\sigma_{\mathcal{N}}(\boldsymbol{\theta})}\right)\end{aligned}. \tag{14}$$

Here, EO algorithm (Faramarzi et al. 2020) is also employed to find $\boldsymbol{\theta}_{\max}^{+}$.

Similarly, the normalized version of stopping criterion for upper bound optimization scheme is adopted (Huang et al. 2006):

$$\frac{\max_{\boldsymbol{\theta}\in\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]}\mathcal{L}_{\max}^{\mathrm{EI}}\left(\boldsymbol{\theta}\right)}{\max_{1\leqslant j\leqslant\mathcal{N}}\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)-\min_{1\leqslant j\leqslant\mathcal{N}}\hat{m}\left(\boldsymbol{\theta}^{(j)}\right)}<\mathcal{E}_2, \tag{15}$$

where the stopping tolerance $\mathcal{E}_2$ can take the same value as $\mathcal{E}_1$ for convenience. If Eq. (15) is not satisfied, $\boldsymbol{\theta}_{\max}^{+}$ and corresponding response expectation $\hat{m}\left(\boldsymbol{\theta}_{\max}^{+}\right)$ estimated according to Section 4 are added to the training dataset $\mathcal{D}$. Then, another round of upper bound optimization is performed based on the enriched $\mathcal{D}$. The optimization scheme stops only when Eq. (15) is satisfied for three times consecutively.

**Remark.** Note that it is also possible to perform the upper bound optimization first and then the lower bound optimization. In this case, the training dataset $\mathcal{D}$ obtained from the upper bound optimization will be used as the initial training dataset for the lower bound optimization.

## Unscented transform

As observed from Eq. (1), the evaluation of $m\left(\boldsymbol{\theta}\right)$ at a fixed design point $\boldsymbol{\theta}$ becomes a deterministic but still difficult-to-evaluate integration. In this regard, one may resort to use the numerical integration method to approximate such deterministic integration. Denote the approximated solution of $m(\boldsymbol{\theta})$ at a fixed observation $\boldsymbol{\theta}$ as $\hat{m}$. Using the numerical integration method, $\hat{m}$ can be expressed as:

$$\hat{m} = \sum_{r=1}^{N_q} w_r g\left(\boldsymbol{\chi}_r\right), \tag{16}$$

in which $N_q$ is the number of integration points; $\boldsymbol{\chi}_r$ is the $r$-th integration point, and $w_r$ is the corresponding $r$-th weight.

Under this setting, there is a need for an efficient method to evaluate $\hat{m}$. The unscented transform (UT) (Julier and Uhlmann 1997b; Wan and Van Der Merwe 2000; Jia et al. 2013) is adopted in our work, since UT is computationally more efficient while maintaining acceptable accuracy, compared to MCS, univariate dimension reduction method, and sparse grid numerical integration utilized in existing optimization-integration methods (Bruns and Paredis 2006; Liu et al. 2018; Liu et al. 2019). The UT was first introduced by Jeffrey Uhlmann (Julier and Uhlmann 1997b) in the field of nonlinear Kalman filter, which enables to calculate the expectation of a random vector propagated through a nonlinear transformation. The basic idea of UT is to first select a finite number of sample points, also known as sigma points, transform these sigma points by a nonlinear transformation, and finally perform a weighted summation of the transformed sigma points to obtain an estimate of the response expectation. The sigma points are obtained by sampling in the original Gaussian distribution according to certain rules, and the corresponding weights satisfy the results of the weighted sum of the sigma points with the same mean and variance of the Gaussian distribution. Accordingly, the sigma points and corresponding weights can be respectively given by (Julier and Uhlmann 1997b; Julier and Uhlmann 1997a):

$$\begin{cases} \boldsymbol{\gamma}_1 = [0, ..., 0]^{\mathrm{T}}, & w_1 = \frac{\kappa}{n_s+\kappa}, \quad r = 1 \\ \boldsymbol{\gamma}_r = \sqrt{n_s+\kappa}\,\boldsymbol{e}_{r-1}, & w_r = \frac{1}{2(n_s+\kappa)}, \; r = 2, \cdots, n_s+1 \\ \boldsymbol{\gamma}_r = -\sqrt{n_s+\kappa}\,\boldsymbol{e}_{r-n_s-1}, & w_r = \frac{1}{2(n_s+\kappa)}, \; r = n_s+2, \cdots, 2n_s+1, \end{cases} \tag{17}$$

where $\boldsymbol{e}_{r-1}$ is the $n_s$-dimensional unit vector with the $(r-1)$-th element being 1; $\kappa$ is the scaling factor that enables to tune the accuracy of moment approximations. According to Ref. (Julier and Uhlmann 1997b), it is suggested to take $\kappa = 3 - n_s$ for sigma points following the Gaussian distribution. In this manner, we have

$$\begin{cases} \boldsymbol{\gamma}_1 = [0, ..., 0]^{\mathrm{T}}, & w_1 = \frac{3-n_s}{3}, \; r = 1 \\ \boldsymbol{\gamma}_r = \sqrt{3}\boldsymbol{e}_{r-1}, & w_r = \frac{1}{6}, \quad r = 2, \cdots, n_s + 1 \\ \boldsymbol{\gamma}_r = -\sqrt{3}\boldsymbol{e}_{r-n_s-1}, & w_r = \frac{1}{6}, \quad r = n_s + 2, \cdots, 2n_s + 1. \end{cases} \tag{18}$$

Based on the above sigma points and corresponding weights, the response expectation can be estimated such that:

$$\hat{m} = \sum_{r=1}^{N_q} w_r g\left(\Gamma^{-1}\left(\boldsymbol{\gamma}_r \,|\boldsymbol{\theta}\right)\right), \tag{19}$$

where $N_q = 2n_s + 1$, which is highly efficient to evaluate the response expectation; $\Gamma^{-1}\left(\cdot\,|\boldsymbol{\theta}\right)$ represents the isoprobabilistic transformation that transform sigma points from the standard Gaussian space to the original input random space. In this regard, the integration points can be regarded as the transformed sigma points, where the transformation relationship is $\boldsymbol{\chi}_r = \Gamma^{-1}\left(\boldsymbol{\gamma}_r \,|\boldsymbol{\theta}\right), r = 1, ..., N_q$. It is worth mentioning that responses corresponding to those $N_q$ sigma points involved in Eq. (19), i.e., $g\left(\Gamma^{-1}\left(\boldsymbol{\gamma}_r \,|\boldsymbol{\theta}\right)\right), r = 1, ..., N_q$, can be evaluated in parallel.

Note that the sigma points are generated by matching the moments of Gaussian random variables up to the second order. In addition, all odd-ordered moments of a Gaussian variable are zero. Therefore, the UT is able to estimate response expectation at a fixed $\boldsymbol{\theta}$ up to the third order, regardless of the dimension of input variables (Julier and Uhlmann 1997a; Wan et al. 2001). Nevertheless, the UT is somehow difficult to adapt to problems with strong nonlinearities (Julier 2002). In this regard, it is reasonable to expect that the use of UT to evaluate the response expectation can have acceptable accuracy and efficiency for linear and moderately nonlinear problems. However, the approximation error introduced by UT needs to be considered, as it may affect the accuracy of the outer-loop optimization within the distribution parameter space. To take into account the approximation error in the optimization process, the estimated response expectation at a fixed design point $\boldsymbol{\theta}$, i.e., $\hat{m}\left(\boldsymbol{\theta}^{(i)}\right)$, is treated as a noisy observation in the training dataset $\mathcal{D}$.

**Step-by-step procedure**

Once both the stopping criteria associated with the lower and upper bounds are satisfied in the optimization process, the lower and upper response expectation bounds can be obtained from the final training dataset $\mathcal{D} = \left\{\boldsymbol{\Theta}, \hat{\mathcal{M}}\left(\boldsymbol{\Theta}\right)\right\}$, such as:

$$\underline{m} = \min_{1 \leqslant j \leqslant \mathcal{N}} \hat{m}\left(\boldsymbol{\theta}^{(j)}\right), \tag{20}$$

$$\overline{m} = \max_{1 \leqslant j \leqslant \mathcal{N}} \hat{m}\left(\boldsymbol{\theta}^{(j)}\right), \tag{21}$$

where the current $\mathcal{N}$ is the sample size of the final obtained $\mathcal{D}$. Accordingly, the total number of response function calls required by response expectation bound estimation is $N = \mathcal{N} \times N_q = \mathcal{N} \times (2n_s + 1)$. To distinguish the stopping criterion involved in lower bound estimation with that involved in upper bound estimation, the first criterion is named criterion 1, and the second is named criterion 2 in the following. A flowchart of the proposed method is shown in Fig. 1. To illustrate the procedure of proposed method, here we take the evaluation of the lower bound of response expectation as an example, and a brief procedure is summarized as follows:

**Step 1**: Initialization. Set the initial sample size $\mathcal{N}_{\text{ini}}$ and stopping tolerance $\mathcal{E}_1$. Create the initial training set $\mathcal{D} = \left\{\boldsymbol{\Theta}, \hat{\mathcal{M}}\left(\boldsymbol{\Theta}\right)\right\}$ of size $\mathcal{N}_{\text{ini}}$ by two steps. First, randomly sample $\mathcal{N}_{\text{ini}}$ distribution parameters $\boldsymbol{\theta}$ from the hyperrectangle $\left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]$ by adopting the Latin hypercube sampling (LHS) method, and form $\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, ..., \boldsymbol{\theta}^{(\mathcal{N}_{\text{ini}})}\right\}^{\mathrm{T}}$. Then, employ the UT to estimate $\mathcal{N}_{\text{ini}}$ response expectations $\hat{m}\left(\boldsymbol{\theta}\right)$ at each component of $\boldsymbol{\Theta}$, and accumulate these resultant estimated expectations as $\hat{\mathcal{M}}\left(\boldsymbol{\Theta}\right) = \left\{\hat{m}\left(\boldsymbol{\theta}^{(1)}\right), ..., \hat{m}\left(\boldsymbol{\theta}^{(\mathcal{N}_{\text{ini}})}\right)\right\}^{\mathrm{T}}$. Denote the current sample size as $\mathcal{N}$, where $\mathcal{N} = \mathcal{N}_{\text{ini}}$ at present.

**Step 2**: Optimization for finding $\boldsymbol{\theta}^{+}_{\text{min}}$. This step involves first training a noisy GP model of $m\left(\boldsymbol{\theta}\right)$ based on the current training set $\mathcal{D}$ such that $m_{\mathcal{N}}\left(\boldsymbol{\theta}\right) \sim \text{GP}\left(\mu_{\mathcal{N}}\left(\boldsymbol{\theta}\right), \sigma^2_{\mathcal{N}}\left(\boldsymbol{\theta}\right)\right)$. The training of noisy GP model is realized by using the *fitrgp* function in the Matlab "Statistic and Machine Learning Toolbox", where the initial value for the noise standard deviation $\sigma_\epsilon$ is set to be 0.002. Then, the current best lower bound $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right)$ is specified by $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right) = \min\left\{\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{(1)}\right), \mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{(2)}\right), ..., \mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{(\mathcal{N})}\right)\right\}$. The new update point $\boldsymbol{\theta}^{+}_{\min}$ is selected from the hyper-rectangle $\left[\underline{\boldsymbol{\theta}}, \bar{\boldsymbol{\theta}}\right]$ by maximizing EI over $\mu_{\mathcal{N}}\left(\boldsymbol{\theta}^{\star}_{\min}\right)$, where the EO algorithm is employed.

**Step 3**: Check the stopping criterion 1. To accommodate stochastic evaluations, criterion 1 in Eq. (12) is checked by three times successively. If the criterion 1 is satisfied, end the updating process and output the current $\mathcal{D}$ as the initial training set for upper bound optimization; otherwise, go to step 4.

**Step 4**: Evaluation of the response expectation at $\boldsymbol{\theta}^{+}_{\min}$. The response expectation at the new update point $\boldsymbol{\theta}^{+}_{\min}$, i.e., $\hat{m}\left(\boldsymbol{\theta}^{+}_{\min}\right)$, is evaluated by the UT according to Eq. (19). A total of $N_q = 2n_s + 1$ sigma points and corresponding weights involved in UT are generated by Eq. (18).

**Step 5**: Enrichment of the training dataset. The new update point $\boldsymbol{\theta}^{+}_{\min}$ and corresponding expectation value $\hat{m}\left(\boldsymbol{\theta}^{+}_{\min}\right)$ are added into the training set $\mathcal{D}$. Then, set $\mathcal{N} = \mathcal{N} + 1$ and go to Step 2 to perform a new round of optimization.

## TEST EXAMPLES

In this section, four test examples are investigated to verify the feasibility of the proposed method. In all cases, the size of initial training dataset takes $\mathcal{N}_{\text{ini}} = \min\{2n_\theta, 10\}$, and the stopping tolerances for both lower bound and upper bound estimations take $\mathcal{E}_1 = \mathcal{E}_2 = \mathcal{E} = 0.002$. To illustrate the advantages of the proposed method, two existing optimization-integration methods, i.e., OSGNI (Liu et al. 2019) and OUDRM (Liu et al. 2018), are performed for comparison in all examples. Both of these two methods employ the *fmincon* algorithm with sequential quadratic programming (SQP) method in Matlab for searching the optimal distribution parameters in the optimization process, where the termination tolerances for first-order optimality and step size are set to be $10^{-6}$. At a certain set of distribution parameters, the OSGNI adopts the sparse grid numerical integration method (SGNI) (Heiss and Winschel 2008) using the nested quadrature rule with Gaussian weights to evaluate the response expectation, where the accuracy level $k_{\text{acc}}$ representing the order of polynomial used for fitting is prescribed. The OUDRM employs the univariate dimension reduction method (UDRM) (Rahman and Xu 2004) for expectation estimation, in which the number of Gauss-Hermite points used, denoted as $N_{\mathrm{G}}$, is given in advance. For all examples, $k_{\mathrm{acc}} = 3$ and $N_{\mathrm{G}} = 6$, unless otherwise specified in the example. Furthermore, in the first two examples, we also compare the results obtained by DLMCS (Bruns and Paredis 2006) method, Vertex-MCS (Dong and Shah 1987) method, and OPS (Bruns and Paredis 2006) method. Note that the outer loop of OPS is also performed by adopting MATLAB function *fmincon* with SQP algorithm, while the inner loop of OPS employs the MCS with $10^4$ runs. All the above methods are implemented in MATLAB on the same computer with Intel Core i7-11800H at 2.30 GHz and 32GB of RAM.

**Example 1: a two-dimensional toy example**

A two-dimensional toy example is first investigated, whose response function is given by:

$$y = g\left(x_1, x_2\right) = 1 + \frac{\left(x_1 - 1\right)^3}{9} + \frac{\left(x_2 - 1\right)^3}{16}, \tag{22}$$

where $x_1$ and $x_2$ are both Gaussian random variables with non-deterministic distribution parameters, i.e., mean and standard deviation. The mean parameters of $x_1$ and $x_2$, denoted as $\mu_1$ and $\mu_2$, take the same interval value $[-1, 3]$. And both the standard deviation parameters of $x_1$ and $x_2$, i.e., $\sigma_1$ and $\sigma_2$, are set as $[0.5, 3]$.

In this example, the lower and upper bounds of response expectation are estimated by the analytical method, DLMCS, Vertex-MCS, OPS, OSGNI, OUDRM and the proposed method. Since this example is simple, the OUDRM employs the UDRM using $N_{\mathrm{G}} = 2$ Gauss-Hermite points in the inner loop. To examine the robustness, each method is repeatedly performed 10 times. The average results obtained by each method and the corresponding average total number of response function calls (denoted as $N$) are presented in Table 2, along with the average number of simulations associated with the lower and upper bounds (denoted as $N_L$ and $N_U$). Additionally, the coefficients of variation (COVs) for the estimated bounds are reported. The analytical solution of response expectation can be easily derived as $\mu_{\mathrm{true}} = 1 + \frac{1}{9}\left(\mu_1 - 1\right)\left[\left(\mu_1 - 1\right)^2 + 3{\sigma_1}^2\right] + \frac{1}{16}\left(\mu_2 - 1\right)\left[\left(\mu_2 - 1\right)^2 + 3{\sigma_2}^2\right]$, which provides the analytical lower and upper bounds of response expectation as $\underline{m} = -9.7639$ and $\overline{m} = 11.7639$, respectively. Note that since the UT has third-order algebraic accuracy, the response expectation estimated using the UT should be accurate for any given set of distributed parameters in this example. Compared with the analytical results, the proposed method obtains both the lower and upper bounds of response expectation in a robust and accurate manner. On average, only a total of $N = N_{\mathrm{L}} + N_{\mathrm{U}} = 76 + 27 = 103$ response function calls are required, where $\mathcal{N}_{\mathrm{ini}}$ is included in $N_{\mathrm{L}}$. The OSGNI and OUDRM enable to provide quite accurate bound results, however, these two existing methods require more response function calls compared with the proposed method. In this sense, more computational efforts are required by the OSGNI and OUDRM. In addition, as observed from Table 2, the Vertex-MCS and OPS are able to give relatively accurate bounds, but both require more than one million samples, which is considerably expensive. Unfortunately, the traditional and widely used DLMCS is unable to obtain accurate lower and upper bounds on response expectation. Besides, the COVs of DLMCS results are larger than those by other methods.

**Example 2: a 120-bar spatial truss structure**

Example 2 investigates a 120-bar spatial truss structure subjected to seven vertical nodal loads (Dang et al. 2021), shown in Figure 2. In this figure, the nodes that bear vertical loads are marked with red circles and numbers. The vertical displacement of the top node of this structure is of interest in this example, which is analyzed by a finite element software, OpenSees. Each member is modeled as a truss element. A total of 48 nodes and 120 elements are involved in the finite element model. The Young's modulus $E_0$, cross-sectional area of element $A$ and seven vertical nodal loads (i.e., $P_0, P_2, P_4, P_6, P_8, P_{10}, P_{12}$) are considered as input variables. Among them, $E_0$, $A$ and $P_0$ are p-box variables, and $P_2, P_4, P_6, P_8, P_{10}$ and $P_{12}$ are aleatory variables. The description of these nine input variables is provided in Table 3.

In this example, the expectation bounds of the response of interest are estimated by the Vertex-MCS, DLMCS, OPS, OSGNI, OUDRM and the proposed method, where the corresponding results are given in Table 4. We take the result obtained by the Vertex-MCS as the reference. As observed, the lower and upper response expectation bounds by the proposed method accord fairly well with the reference bounds. In addition, only $N = N_L + N_U = 247 + 57 = 304$ response function calls are required in the proposed method, which is within affordable computational limits. Unfortunately, other selected double-loop methods, i.e., the DLMCS, OPS, OSGNI and OUDRM, can only provide narrower bounds on the response expectation but require much more computational efforts.

**Example 3: a jet engine turbine blade**

The third example consists of a jet engine turbine blade under pressure loading, as illustrated in Fig. 3a (MATLAB 2022). The turbine blade is governed by two mechanical boundary conditions, namely the pressure loads $P_1$ and $P_2$ on the pressure and suction sides caused by the surrounding high-pressure gases, and the fixed Dirichlet boundary condition on the left side. The model is discretized by adopting the linear tetrahedral elements with maximum element size as 0.01 m, as shown in Fig. 3b. A total of 21252 nodes and 11794 elements are involved. This turbine blade is assumed to be made by the nickel-based alloy (NIMONIC 90) material, where the Young's modulus, coefficient of thermal expansion and Poisson's ratio are represented by $E$, $\alpha$ and $\nu$, respectively. Here, the maximum von Mises stress of the turbine blade caused by high pressure from surrounding gases is the response of interest, which can be obtained by performing linear stress analysis using the Matlab Partial Differential Equation (PDE) Toolbox (MATLAB 2022). Five p-box variables, i.e., $\{E, \alpha, \nu, P_1, P_2\}$, are considered in this example, of which the mean and standard deviation parameters are all bounded by intervals. The detailed description of these p-box variables is listed in Table 5. Fig. 3c shows a resultant von Mises stress nephogram obtained by performing one structural analysis with all input variables to be fixed at the midpoint of their mean parameter intervals. As seen, the maximum von Mises stress happens at the tip of the turbine blade.

The OSGNI, OUDRM, Vertex-MCS and proposed method are employed in this example to estimate the lower and upper bounds of response expectation, whose results are provided in Table 6. It can be observed that the proposed method is able to obtain the response expectation bounds that are almost identical to those by the Vertex-MCS. However, much fewer response function calls (specifically $N = 143 + 33 = 176$) are required by the proposed method, indicating that the proposed method has comparable accuracy but higher computational efficiency. In comparison, the OSGNI and OUDRM give narrower response expectation bounds but cost considerably larger computational efforts. Note that although this example involves only linear stress analysis, the large number of discrete elements results in a relatively long time for one evaluation of the response. We record the total computational times for all the methods implemented in this example, which are also given in Table 6. It is found that the proposed method takes much less time than OSGNI, OUDRM, and Vertex-MCS. Thus, it indicates that the proposed method is more computationally efficient for this example.

**Example 4: a crash box in the vehicle**

Last example investigates the frontal impact problem of a crash box impacted by a moving planar impactor. The crash box is an important energy absorbing component installed at the front of the vehicle, which determines the crashworthiness and ensures the safety of the vehicle. A quarter of a symmetric crash box (Reid 1998) shown in Fig. 4 is considered in this example, which is analyzed by the LS-DYNA software in symmetric multiprocessing (SMP) version. The crash box is built as a tube with an uncertain shell thickness $t$, and adopts a steel-like material modeled by a piecewise linear plastic model with Possion's ratio as 0.3, yield strength as 207 MPa, mass density as 7830 $\mathrm{kg/m^3}$, strain rate model as Cowper-Symmonds with parameter $C = 40$ and $p = 5$, and an uncertain Young's modulus $E$. The lower end of the crash box is fixed. A planar impactor, modeled as a rigid wall with imprecisely known mass $M_{\mathrm{wall}}$ and initial velocity $v_{\mathrm{wall}}$, crushes the crash box from the top downwards. Three triggers are applied to the crash box in order to achieve desired energy absorption and deformation pattern. The LS-DYNA keyword $*CONTACT\_AUTOMATIC\_SINGLE\_SURFACE$ is applied to formulate the contact between the impactor and the crash box. The simulation is terminated when the impactor stops moving or the total time reaches 15.01 ms. This example involves a total of 4 p-box variables with 8 imprecise distribution parameters, i.e., $\{M_{\mathrm{wall}}, v_{\mathrm{wall}}, E, t\}$, whose detailed description is listed in Table 7. Fig. 4c shows the deformation of the crash box under rigid wall impact, where $\{M_{\mathrm{wall}}, v_{\mathrm{wall}}, E, t\} = \{800\ \mathrm{kg}, 8.94\ \mathrm{m/s}, 200\ \mathrm{GPa}, 2\ \mathrm{mm}\}$. As observed, the impacted crash box deforms in a folding mode without global bending, showing its good ability to absorb the impact energy. In addition, under the same input conditions, the force-displacement curve of the rigid wall in the negative Z direction is illustrated in Fig. 5, which indicates that the investigated crash box undergoes irreversible nonlinear buckling deformation, and has a relatively strong nonlinearity.

The output response of interest is the average force of the complete force-displacement curve measured at the rigid wall. Note that this example also requires a long computational time to perform a simulation. In order to demonstrate the effectiveness of the proposed method, a comparison is made between the results obtained from the proposed method, the Vertex-MCS, OSGNI and UDRM, as summarized in Table 8, alongside the respective total computational times. As observed, the proposed method enables to provide lower and upper bounds on the expectation of the averaged rigid wall force that are quite close to those of Vertex-MCS, while the proposed method requires much fewer response function calls, specifically $N = 162 + 63 = 225$. In comparison, OSGNI and OUDRM produce narrower response expectation bounds but require a larger number of simulations. Moreover, the computational time for the proposed method is 2134.01 s, while the Vertex-MCS, OSGNI and OUDRM require 13245.27 s and 6639.07 s, respectively. Hence, this example illustrates that the proposed method can be applied not only to linear and weakly nonlinear problems, but also to problems with relatively strong nonlinearity.

## CONCLUDING REMARKS

In this paper, an efficient optimization-integration method is developed for estimating the lower and upper bounds of response expectation for linear and moderately nonlinear problems with inputs characterized by parametric p-boxes. The proposed method combines the Bayesian global optimization (BGO) with a highly efficient numerical integration method named unscented transform (UT), to sequentially evaluate lower and upper bounds on response expectations. An adaptively refined noisy Gaussian process (GP) model is adopted to explore the space of distribution parameters considering the approximation error introduced by UT. Besides, the sequential design strategy of BGO allows the proposed method to reuse the samples generated by the lower bound estimation in the upper bound estimation. In the process of response expectation at a given set of distribution parameters, the UT is quite efficient and can obtain the estimates of response expectation up to third accuracy. Four test examples are investigated to demonstrate the applicability to both linear and moderately nonlinear problems. For all of these examples, the results obtained by the proposed method use a reasonable number of response function calls. In addition, the resultant response expectation bounds are almost the same as the provided reference results, showing the effectiveness of the proposed method. Compared with some existing double-loop methods such as DLMCS, Vertex-MCS, OPS, OSGNI and OUDRM, the proposed method is able to acquire the results with acceptable accuracy and higher computational efficiency. It can also be observed from the four test examples that the accuracy of the proposed method is mainly affected by the complexity and nonlinearity of the problem at hand. For simpler problems, increasing the level of epistemic uncertainty does not affect the accuracy, while for more complex problems, a higher level of epistemic uncertainty tends to have a greater impact on the accuracy of the results.

Admittedly, since the approximated expectation by the UT has only up to third order accuracy, the proposed method is not suitable for addressing strong nonlinear problems and evaluating higher-order response moments. To mitigate this, we are actively exploring alternative numerical integration methods, such as the mixed degree cubature scheme (He et al. 2022), to capture higher-order moments within our framework. Besides, the BGO in the optimization process still suffers from the so-called “curse of dimensionality" problem, i.e., it may perform poorly for problems with more than 20 dimensions. Future work will focus on a time-saving method for evaluating bounds on higher-order response moments that is applicable to higher dimensional and stronger nonlinear problems.

## DATA AVAILABILITY STATEMENT

All data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

## ACKNOWLEDGMENTS

Chen Ding acknowledges the support of the European Union’s Horizon 2020 research and innovation programme under Marie Sklodowska-Curie project GREYDIENT – Grant Agreement n°955393. Chao Dang thanks the support from the China Scholarship Council (CSC). Michael Beer appreciates the support of National Natural Science Foundation of China under grant 72271025.

## REFERENCES


Beer, M., Ferson, S., and Kreinovich, V. (2013). "Imprecise probabilities in engineering analyses." *Mechanical Systems and Signal Processing*, 37(1-2), 4–29.

Bruns, M. and Paredis, C. J. (2006). "Numerical methods for propagating imprecise uncertainty." *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 4255, 1077–1091.

Bruns, M. C. (2006). "Propagation of imprecise probabilities through black box models." Ph.D. thesis, Georgia Institute of Technology.

Dang, C., Wei, P., Faes, M. G., and Beer, M. (2022a). "Bayesian probabilistic propagation of hybrid uncertainties: Estimation of response expectation function, its variable importance and bounds." *Computers & Structures*, 270, 106860.

Dang, C., Wei, P., Faes, M. G., Valdebenito, M. A., and Beer, M. (2022b). "Interval uncertainty propagation by a parallel Bayesian global optimization method." *Applied Mathematical Modelling*, 108, 220–235.

Dang, C., Wei, P., Song, J., and Beer, M. (2021). "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), 04021054–04021054.

Dong, W. and Shah, H. C. (1987). "Vertex method for computing functions of fuzzy variables." *Fuzzy Sets and Systems*, 24(1), 65–78.

Faes, M. G., Daub, M., Marelli, S., Patelli, E., and Beer, M. (2021). "Engineering analysis with probability boxes: A review on computational methods." *Structural Safety*, 93, 102092.

Faramarzi, A., Heidarinejad, M., Stephens, B., and Mirjalili, S. (2020). "Equilibrium optimizer: A novel optimization algorithm." *Knowledge-Based Systems*, 191, 105190.

Ferson, S. and Hajagos, J. G. (2004). "Arithmetic with uncertain numbers: rigorous and (often) best possible answers." *Reliability Engineering & System Safety*, 85(1-3), 135–152.

He, S., Xu, J., and Zhang, Y. (2022). "Reliability computation via a transformed mixed-degree cubature rule and maximum entropy." *Applied Mathematical Modelling*, 104, 122–139.

Heiss, F. and Winschel, V. (2008). "Likelihood approximation by numerical integration on sparse grids." *Journal of Econometrics*, 144(1), 62–80.

Huang, D., Allen, T. T., Notz, W. I., and Zeng, N. (2006). "Global optimization of stochastic black-box systems via sequential kriging meta-models." *Journal of Global Optimization*, 34(3), 441–466.

Jia, B., Xin, M., and Cheng, Y. (2013). "High-degree cubature kalman filter." *Automatica*, 49(2), 510–518.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). "Efficient global optimization of expensive black-box functions." *Journal of Global Optimization*, 13(4), 455–492.

Julier, S. J. (2002). "The scaled unscented transformation." *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, Vol. 6, IEEE, 4555–4559.

Julier, S. J. and Uhlmann, J. K. (1997a). "Consistent debiased method for converting between polar and cartesian coordinate systems." *Acquisition, Tracking, and Pointing XI*, Vol. 3086, SPIE, 110–121.

Julier, S. J. and Uhlmann, J. K. (1997b). "New extension of the kalman filter to nonlinear systems." *Signal processing, sensor fusion, and target recognition VI*, Vol. 3068, International Society for Optics and Photonics, 182–193.

Liu, H., Jiang, C., Jia, X., Long, X., Zhang, Z., and Guan, F. (2018). "A new uncertainty propagation method for problems with parameterized probability-boxes." *Reliability Engineering & System Safety*, 172, 64–73.

Liu, H., Jiang, C., Liu, J., and Mao, J. (2019). "Uncertainty propagation analysis using sparse grid technique and saddlepoint approximation based on parameterized p-box representation." *Structural and Multidisciplinary Optimization*, 59(1), 61–74.

MATLAB (2022). *Partial Differential Equation Toolbox*. (`https://www.mathworks.com/help/pde/ug/thermal-stress-analysis-of-jet-engine-turbine-blade.html`), The MathWorks Inc., Natick, Massachusetts, United States.

Pedroni, N. and Zio, E. (2015). “Hybrid uncertainty and sensitivity analysis of the model of a twin-jet aircraft.” *Journal of Aerospace Information Systems*, 12(1), 73–96.

Rahman, S. and Xu, H. (2004). “A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics.” *Probabilistic Engineering Mechanics*, 19(4), 393–408.

Reid, J. D. (1998). *LS-Dyna Examples Manual*. Livermore Software Technology Corporation, Livermore, California, United States.

Wan, E. A. and Van Der Merwe, R. (2000). “The unscented kalman filter for nonlinear estimation.” *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, IEEE, 153–158.

Wan, E. A., Van Der Merwe, R., and Haykin, S. (2001). “The unscented kalman filter.” *Kalman Filtering and Neural Networks*, 5(2007), 221–280.

Wei, P., Hong, F., Phoon, K.-K., and Beer, M. (2021a). “Bounds optimization of model response moments: a twin-engine Bayesian active learning method.” *Computational Mechanics*, 67(5), 1273–1292.

Wei, P., Liu, F., Valdebenito, M., and Beer, M. (2021b). “Bayesian probabilistic propagation of imprecise probabilities with large epistemic uncertainty.” *Mechanical Systems and Signal Processing*, 149, 107219.

Wei, P., Lu, Z., and Song, J. (2014). “Extended monte carlo simulation for parametric global sensitivity analysis and optimization.” *AIAA Journal*, 52(4), 867–878.

Wei, P., Song, J., Bi, S., Broggi, M., Beer, M., Lu, Z., and Yue, Z. (2019). “Non-intrusive stochastic analysis with parameterized imprecise probability models: I. performance estimation.” *Mechanical Systems and Signal Processing*, 124, 349–368.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*, Vol. 2. MIT press Cambridge, MA.

Xiao, Z., Han, X., Jiang, C., and Yang, G. (2016). “An efficient uncertainty propagation method for parameterized probability boxes.” *Acta Mechanica*, 227(3), 633–649.

Zhang, H., Mullen, R. L., and Muhanna, R. L. (2010). “Interval monte carlo methods for structural reliability.” *Structural Safety*, 32(3), 183–190.

Zhang, H., Mullen, R. L., and Muhanna, R. L. (2012). “Structural analysis with probability-boxes.” *International Journal of Reliability and Safety*, 6(1-3), 110–129.

## List of Tables

**TABLE 1.** List of acronyms

| Acronym | Definition |
|---|---|
| BGO | Bayesian global optimization |
| CABO | Collaborative and adaptive Bayesian optimization |
| COV | Coefficient of variation |
| DLMCS | Double-loop Monte Carlo simulation |
| EI | Expected improvement |
| EMCS | Extended Monte Carlo simulation |
| EO | Equilibrium optimizer |
| GP | Gaussian process |
| MCS | Monte Carlo simulation |
| NIPI | Non-intrusive imprecise probabilistic integration |
| NISS | Non-intrusive imprecise stochastic simulation |
| OPS | Optimized parameter sampling |
| OSGNI | Optimized sparse grid numerical integration method |
| OUDRM | Optimized univariate dimension-reduction method |
| p-box | Probability box |
| PBQO | Parallel Bayesian quadrature optimization |
| PDE | Partial differential equation |
| PDF | Probability density function |
| SGNI | Sparse grid numerical integration method |
| SMP | Symmetric multiprocessing |
| SQP | Sequential quadratic programming |
| UDRM | Univariate dimension-reduction method |
| UT | Unscented transform |
| Vertex-MCS | Vertex-based Monte Carlo simulation |

**TABLE 2.** Comparison of results by different methods (Example 1)

| Method | $\underline{m}$ | COV of $\underline{m}$ | $\overline{m}$ | COV of $\overline{m}$ | $N$ |
|---|---|---|---|---|---|
| Analytical | $-9.7639$ | - | 11.7639 | - | - |
| Vertex-MCS | $-9.7318$ | 0.79% | 11.7839 | 0.47% | $16 \times 10^5$ |
| DLMCS | $-8.1585$ | 4.75% | 9.8238 | 5.97% | $10^4 \times 10^4$ |
| OPS | $-9.8778$ | 0.37% | 11.6531 | 0.29% | $(896 + 1063) \times 10^4 = 1.959 \times 10^7$ |
| OSGNI | $-9.7639$ | 0.00% | 11.7639 | 0.00% | $270 + 270 = 540$ |
| OUDRM | $-9.7639$ | 0.00% | 11.7639 | 0.00% | $140 + 27 = 167$ |
| Proposed | $-9.7639$ | 0.00% | 11.7639 | 0.00% | $76 + 27 = 103$ |

Note: COVs relates to the coefficients of variation; for DLMCS and vertex-MCS, $N = N_{\mathrm{L}} \times N_{\mathrm{U}}$; for OPS, OSGNI, OUDRM and proposed method, $N = N_{\mathrm{L}} + N_{\mathrm{U}}$.

**TABLE 3.** Description of input variables (Example 2)

| Variable | Unit | Description | Distribution | Mean | Standard deviation |
|---|---|---|---|---|---|
| $E_0$ | MPa | Young's modulus | Truncated Normal | $[164800, 247200]$ | $[2060, 10300]$ |
| $A$ | $mm^2$ | Cross section area | Truncated Normal | $[900, 1100]$ | $[1, 5]$ |
| $P_0$ | kN | Vertical load | Lognormal | $[180, 220]$ | $[2, 10]$ |
| $P_2$ | kN | Vertical load | Lognormal | 200 | 2 |
| $P_4$ | kN | Vertical load | Lognormal | 200 | 2 |
| $P_6$ | kN | Vertical load | Lognormal | 200 | 2 |
| $P_8$ | kN | Vertical load | Lognormal | 200 | 2 |
| $P_{10}$ | kN | Vertical load | Lognormal | 200 | 2 |
| $P_{12}$ | kN | Vertical load | Lognormal | 200 | 2 |

Note: Trancated Normal means the values are all positive.

**TABLE 4.** Comparison of results by different methods (Example 2)

| Method | $\underline{m}$ (MPa) | $\overline{m}$ (MPa) | $N$ |
|---|---|---|---|
| Vertex-MCS | 25.2309 | 51.1475 | $64 \times 10^4$ |
| DLMCS | 25.9681 | 49.2774 | $10^3 \times 10^4$ |
| OPS | 27.8320 | 46.2661 | $(57 + 75) \times 10^4 = 132 \times 10^4$ |
| OSGNI | 27.8153 | 46.2762 | $1304 + 1304 = 2608$ |
| OUDRM | 27.8153 | 46.2762 | $432 + 368 = 800$ |
| Proposed | 25.2393 | 51.1061 | $247 + 57 = 304$ |

**TABLE 5.** Description of input variables (Example 3)

| Variable | Unit | Description | Distribution | Mean | Standard deviation |
|---|---|---|---|---|---|
| $E$ | Pa | Young's modulus | Truncated Normal | $[204.3, 249.7] \times 10^9$ | $[227, 1135] \times 10^9$ |
| $\alpha$ | 1/K | Coefficient of thermal expansion | Truncated Normal | $[1.143, 1.397] \times 10^{-5}$ | $[1.270, 6.350] \times 10^{-7}$ |
| $\nu$ | - | Poisson's ratio | Truncated Normal | $[0.243, 0.297]$ | $[0.270, 1.350] \times 10^{-2}$ |
| $P_1$ | Pa | Pressure load | Truncated Normal | $[45, 55] \times 10^4$ | $[5, 25] \times 10^3$ |
| $P_2$ | Pa | Pressure load | Truncated Normal | $[405, 495] \times 10^3$ | $[45, 225] \times 10^2$ |

**TABLE 6.** Comparison of results by different methods (Example 3)

| Method | $\underline{m}$ (MPa) | $\overline{m}$ (MPa) | $N$ | CPU time |
|---|---|---|---|---|
| Vertex-MCS | 93.8542 | 118.8419 | $1024 \times 10^3$ | 804115.49 s |
| OSGNI | 97.0640 | 115.0664 | $1122 + 1122 = 2244$ | 7464.80 s |
| OUDRM | 97.0640 | 115.0664 | $660 + 572 = 1232$ | 5402.62 s |
| Proposed | 94.1447 | 118.6347 | $143 + 33 = 176$ | 821.28 s |

Note: CPU time represents the total computational time.

**TABLE 7.** Description of input variables (Example 4)

| Variable | Unit | Description | Distribution | Mean | Standard deviation |
|---|---|---|---|---|---|
| $M_{\text{wall}}$ | kg | Mass of the rigid wall | Truncated Normal | $[760, 840]$ | $[8, 40]$ |
| $v_{\text{wall}}$ | m/s | Velocity of the rigid wall | Truncated Normal | $[8.10, 9.90]$ | $[0.09, 0.45]$ |
| $E$ | GPa | Young's modulus | Truncated Normal | $[195, 205]$ | $[2, 10]$ |
| $t$ | mm | shell thickness | Truncated Normal | $[1.90, 2.10]$ | $[0.02, 0.10]$ |

**TABLE 8.** Comparison of results by different methods (Example 4)

| Method | $\underline{m}$ (kN) | $\overline{m}$ (kN) | $N$ | CPU time |
|---|---|---|---|---|
| Vertex-MCS | 7.7243 | 9.4776 | $256 \times 10^2$ | 166403.64 s |
| OSGNI | 8.0823 | 9.2635 | $1353 + 297 = 1650$ | 13245.27 s |
| OUDRM | 8.0893 | 9.2379 | $672 + 216 = 888$ | 6639.07 s |
| Proposed | 7.8018 | 9.4252 | $162 + 63 = 225$ | 2134.01 s |

## List of Figures

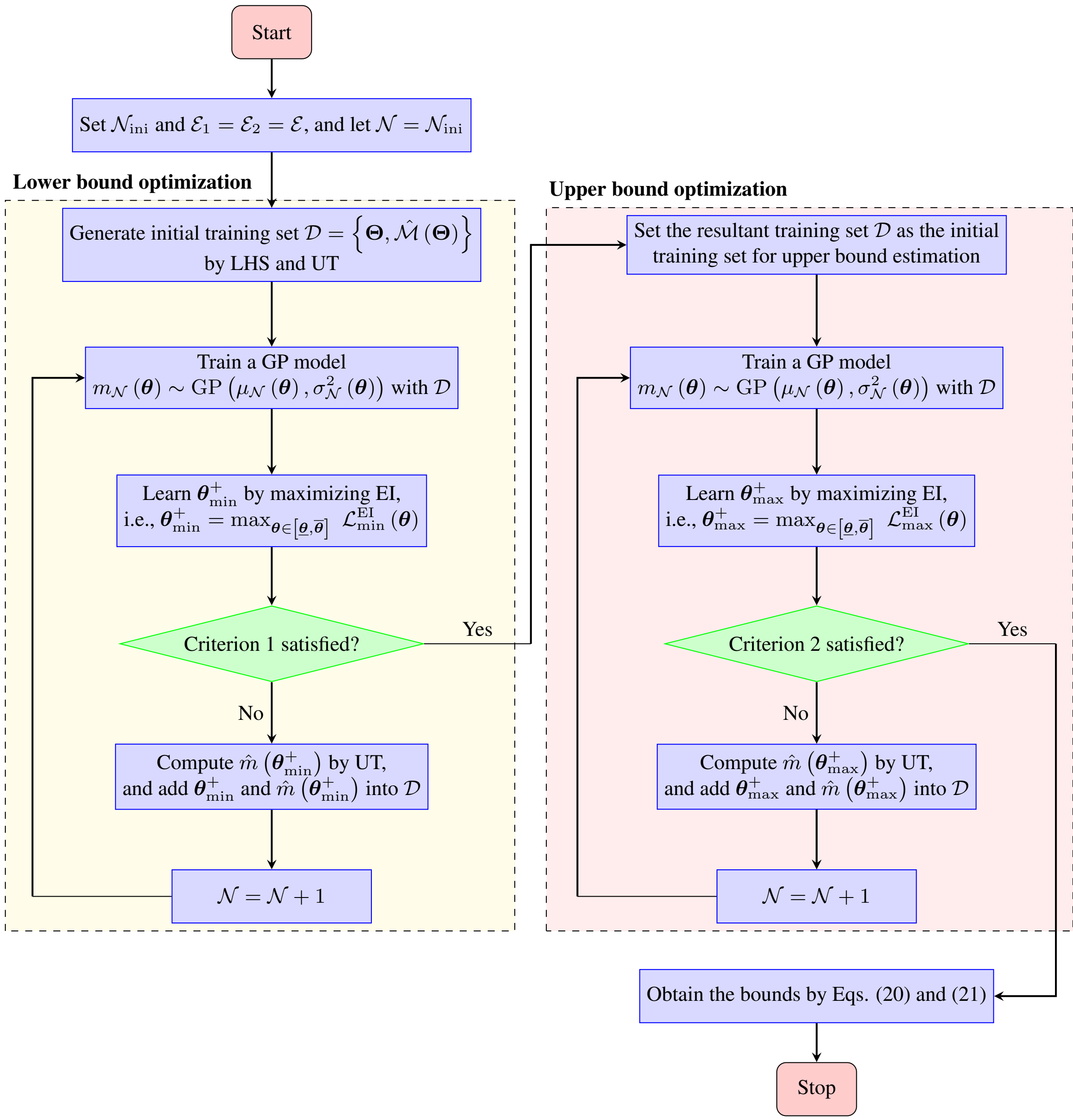



**Fig. 1.** Flowchart of the proposed method

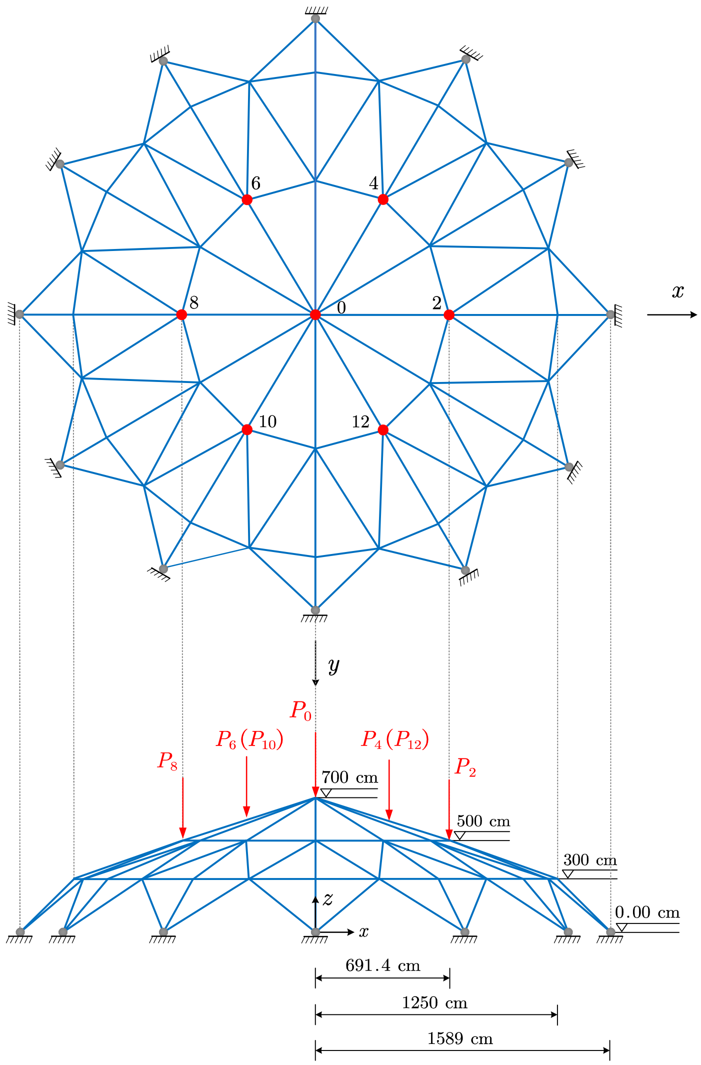
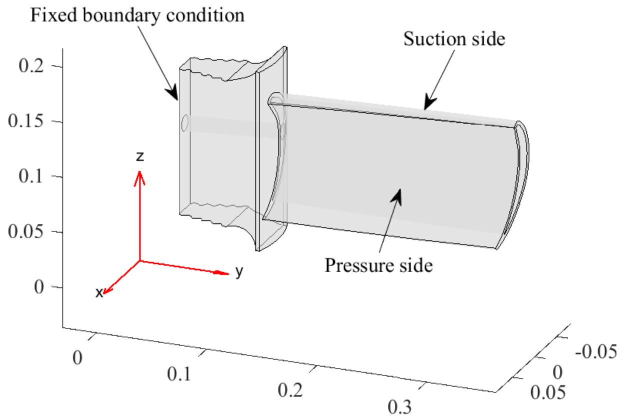
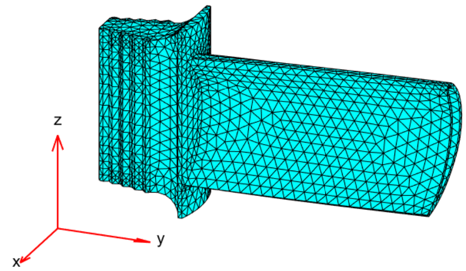


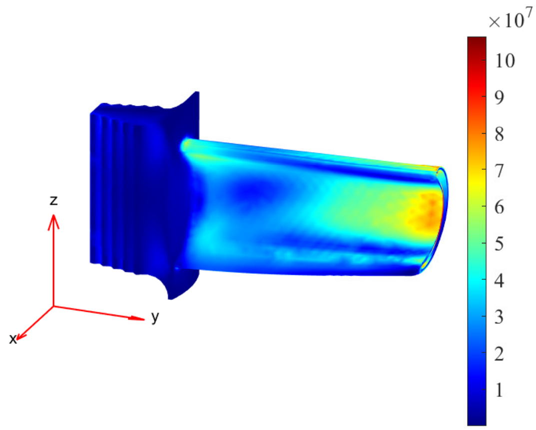

**Fig. 2.** Diagram of 120-bar spatial frame




**(a)** Geometry of the turbine blade (dimensions in m)

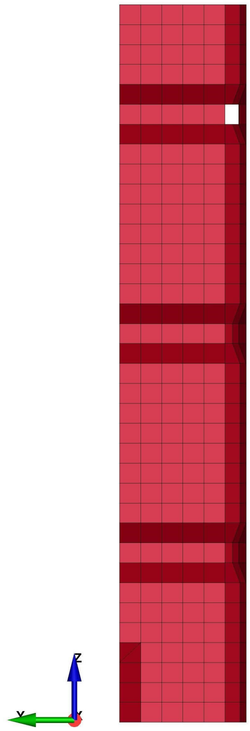



**(b)** Meshed model of the turbine blade

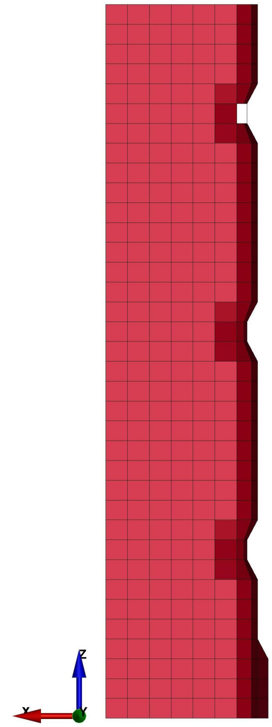



**(c)** Von Mises stress nephogram of structural analysis

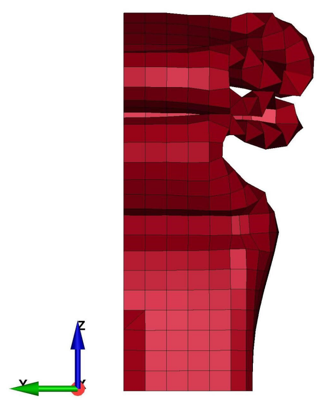
**Fig. 3.** Geometry, mesh diagram and von Mises stress nephogram of the jet engine turbine blade under pressure loads



**(a)** Front view of the meshed model



**(b)** Right view of the meshed model



**(c)** Deformation of the crash box under planar rigid wall impact

**Fig. 4.** Front and right view of the meshed model of a quarter of the crash box, and the deformation of the crash box under planar rigid wall impact

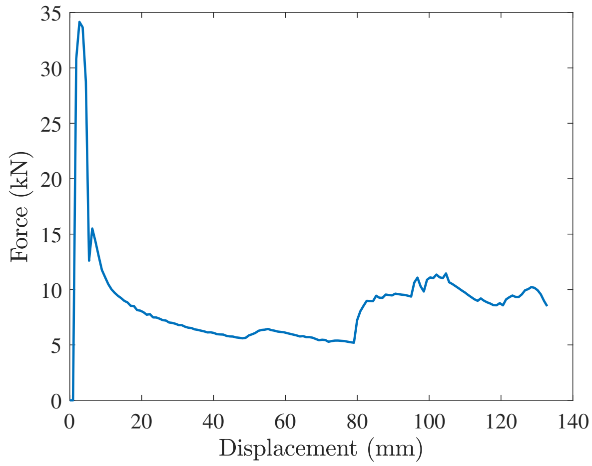



**Fig. 5.** Force-displacement curve of the planar impactor in the negative Z direction