# Bayesian Probabilistic Propagation of Imprecise Probabilities with Large Epistemic Uncertainty

Pengfei Wei [a,*,b], Fuchao Liu [a], Marcos Valdebenito [c], Michael Beer [b,d,e]

[a] *School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, Xi'an 710072, China*

[b] *Institute for Risk and Reliability, Leibniz Universität Hannover, Callinstr. 34, Hannover 30167, Germany*

[c] *Departmento de Obras Civiles, Universidad Tecnica Federico Santa Maria, Av. España 1680, Valparaiso, Chile*

[d] *Institute for Risk and Uncertainty, University of Liverpool, Peach Street, L69 7ZF Liverpool, UK*

[e] *International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, China*

**Abstract**: Efficient propagation of imprecise probability models is one of the most important, yet challenging tasks, for uncertainty quantification in many areas and engineering practices, especially when the involved epistemic uncertainty is substantial due to the extreme lack of information. In this work, a new methodology framework, named as "Non-intrusive Imprecise Probabilistic Integration (NIPI)", is developed for achieving the above target, and specifically, the distributional probability-box model and the estimation of the corresponding probabilistic moments of model responses are of concern. The NIPI owns two attractive characters. First, the spatial correlation information in both aleatory and epistemic uncertainty spaces, revealed by the Gaussian Process Regression (GPR) model, is fully integrated for deriving NIPI estimations of high accuracy by using Bayesian inference. Second, the numerical errors are regarded as a kind of epistemic uncertainty, by analytically propagating them, the posterior variances are derived for indicating the errors of the NIPI estimations. Further, an adaptive experiment design strategy is developed to accelerate the convergence of NIPI by making full use of the information of "contribution to posterior variance" revealed by the GPR model. The performance of the proposed methods is demonstrated by numerical and engineering examples.

**Keywords**: Uncertainty Quantification; Bayesian inference; Probabilistic Integration; Imprecise probabilities; Gaussian Process Regression; Epistemic Uncertainty;

## 1. Introduction

In real-world engineering applications such as structural health monitoring and structural reliability assessment, the uncertainties resulting from different sources, such as natural randomness, measurement errors, vagueness and abstraction of available information, have proved to be very ubiquitous, and prevent us from making reliable assessments and decisions. Commonly, those uncertainties are grouped as either aleatory uncertainty or epistemic uncertainty depending on whether they are reducible or not. The aleatory uncertainty is associated with the random nature of events or parameters, and cannot be reduced with more information, whereas, the epistemic uncertainty is associated with lack of information, and thus can be reduced when more information is collected [1][2]. Correctly distinguishing between different types of uncertainty, based on their nature and their effects on analysis, and properly characterizing these

* Corresponding author at: School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, Xi'an 710072, China
Email address: pengfeiwei@nwpu.edu.cn (Pengfei Wei)

uncertainties with mathematical models have proved to be of primary importance for reliable safety assessment and decision making [1].

Many uncertainty characterization models have been developed for filling the above gap (see, e.g., Refs. [3]-[5] for review of these models), among which the imprecise probability models, such as probability-box (*p*-box) model [6], evidence theory (also called Dempster-Shafer theory) [6], second-order probability model [7] and fuzzy probability model [8], have received great attention due to their hierarchical model structure, which allows separating the two kinds of uncertainties in all uncertainty quantification (UQ) analysis tasks such as uncertainty propagation, model updating, and reliability analysis. However, it is those hierarchical model structures that make the propagation of the imprecise probability models computationally much more challenging than that of precise probability models. The focus of this paper is on the efficient propagation of the imprecise probability models, especially when the epistemic uncertainty of the model inputs is very large, and specifically, we take the *p*-box model as an example.

The traditional algorithms for dealing with this kind of problems commonly involves a double-loop numerical procedure, and two strategies have been developed. The first strategy involves performing optimization in the outer-loop by regarding the non-deterministic distribution parameters of model inputs as design variables, and then applying stochastic analysis in the inner loop for each design of distribution parameters [9]. The second double-loop strategy is based on performing sampling in the outer-loop in order to draw a set of interval samples, and then performing numerical optimization in the inner-loop for each interval sample [10][11]. Both of the above double-loop strategies can be computationally quite expensive, especially for time-consuming computer simulators such as finite element models (FEMs) of complex structures. Another group of methods aims at reducing the vast computational cost of the first double-loop strategy by reusing the samples in each inner loop iteration. The typical developments of this group include the double-loop advanced line sampling developed in Ref. [12] and the extended Monte Carlo simulation (EMCS) developed in Ref. [13], and also reported in Ref. [14]. The main drawback of this group of methods is that the variability of the estimators can be substantial when the dimension of non-deterministic distribution parameters is high and/or when the input epistemic uncertainty is large.

For tackling the "curse of dimensionality", a general methodology framework, named as Non-intrusive Imprecise Stochastic Simulation (NISS), has been developed by the authors in a set of companion papers [15][16]. These developments provide a general framework for efficiently propagating the imprecise probability models, and for estimating the associated failure probability functions, with only one stochastic analysis. For reliability analysis, both the classical subset simulation [16] and line sampling [17] have been injected into the framework for dealing with the rare failure event analysis. These methods have also been generalized for propagating the imprecise probability models and non-probabilistic models (e.g., interval models) simultaneously [18]. However, our current experience shows that NISS works well for problems with relatively small epistemic uncertainty. In some real-world engineering applications, due to the extreme lack of information, the model inputs may involve substantial epistemic uncertainty, resulting in a large span of the support of the non-deterministic distribution parameters. For this type of problem, the NISS estimations often show large variations, and solving this type of problem is the aim of this work. Besides the above stochastic simulation techniques, surrogate model methods have also been developed for propagating the imprecise probability models (see, e.g., [19] and [20]), but we don't give more details for the sake of brevity.

This paper aims at developing a new methodology framework, named as "Non-intrusive

Imprecise Probabilistic Integration (NIPI)", for propagating the *p*-box models in which the supports of the non-deterministic distribution parameters are large, resulting from the large epistemic uncertainty involved, based on the Bayesian probabilistic integration [21][22]. The developed method has two appealing characteristics, which makes it extremely effective. First, the spatial correlation information in both aleatory uncertainty space and epistemic uncertainty space is integrated for inferring closed-form expressions for integrals, where the spatial correlation information is captured by a Gaussian Process Regression (GPR) model. It is shown that this spatial correlation information makes a substantial contribution to the accuracy of the NIPI estimations. Second, the numerical errors are modeled as a kind of epistemic uncertainty being analytically propagated to the NIPI estimations, making it possible to analytically quantify the discretization errors. Besides, an adaptive experiment design strategy originally developed by the authors in Ref. [23] is also injected into the NIPI framework for further reducing the required number of model calls, by making full use of the spatial correlation information. One numerical test example and two real-world engineering examples are introduced for demonstrating the advantages and disadvantages of the proposed method.

The rest of this paper is organized as follows. Section 2 states the problems to be solved in this work, followed by a brief review of the Bayesian probabilistic integration method in section 3. The NIPI method is then presented in section 4, and the adaptive design strategy for further improving the NIPI method is described in section 5. The test examples, as well as the discussion of results, are presented in section 6. Section 7 gives conclusions to this paper.

## 2. Problem Statement

The problem considered in this work is the efficient propagation of imprecise probability models, which have been well-established for characterizing polymorphic uncertainty [24]. As an example, we only consider the distributional *p*-box model. Let $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in\mathcal{X}$ denote the *n*-dimensional row vector of model inputs, each component of which is characterized by a *p*-box model $f_i(x_i|\boldsymbol{\theta}_i)$, with $\boldsymbol{\theta}_i$ being the non-deterministic distribution parameters of $x_i$. We rearrange the distribution parameter vector as $\boldsymbol{\theta}=(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\cdots,\boldsymbol{\theta}_n)=(\theta_1,\theta_2,\cdots,\theta_d)\in\Theta$, where $d$ refers to the total number of non-deterministic distribution parameters. For simplicity, it is assumed that all the input variables are independent, and the joint density function and cumulative distribution function are formulated as $f(\boldsymbol{x}|\boldsymbol{\theta})=\prod_{i=1}^{n}f_i(x_i|\boldsymbol{\theta}_i)$ and $F(\boldsymbol{x}|\boldsymbol{\theta})=\prod_{i=1}^{n}F_i(x_i|\boldsymbol{\theta}_i)$ respectively, with $F_i(x_i|\boldsymbol{\theta}_i)$ being the marginal distribution function of $x_i$.

The epistemic uncertainty of $\boldsymbol{x}$ is reflected by the uncertainty of $\boldsymbol{\theta}$ which is characterized by independent intervals $\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]$, where $\underline{\boldsymbol{\theta}}=\left(\underline{\boldsymbol{\theta}}_1,\underline{\boldsymbol{\theta}}_2,\cdots,\underline{\boldsymbol{\theta}}_n\right)=\left(\underline{\theta}_1,\underline{\theta}_2,\cdots,\underline{\theta}_d\right)$ is the lower bound, and $\overline{\boldsymbol{\theta}}=\left(\overline{\boldsymbol{\theta}}_1,\overline{\boldsymbol{\theta}}_2,\cdots,\overline{\boldsymbol{\theta}}_n\right)=\left(\overline{\theta}_1,\overline{\theta}_2,\cdots,\overline{\theta}_d\right)$ is the upper bound. For the developments in this work, an auxiliary probability distribution should be assumed for $\boldsymbol{\theta}$. However, one should note that this does not necessarily imply that $\boldsymbol{\theta}$ must be a random vector. This assumption is only necessary for numerical implementation. Denote the joint density and cumulative distribution functions as $p(\boldsymbol{\theta})=\prod_{j=1}^{d}p_j(\theta_j)$ and $P(\boldsymbol{\theta})=\prod_{j=1}^{d}P_j(\theta_j)$ respectively, where $p_j(\theta_j)$ and $P_j(\theta_j)$ indicate the marginal density and distribution functions of $\theta_j$ respectively.

The space $\mathcal{X}$ of input vector $\boldsymbol{x}$ is called ***aleatory space*** as the probability distribution of $\boldsymbol{x}$ characterizes the aleatory uncertainty of model inputs, and the space $\Theta$ is termed as ***epistemic space*** since the interval of $\boldsymbol{\theta}$ models the epistemic uncertainty of the inputs resulting from the lack of information on distribution parameters associated with the probabilistic description of $\boldsymbol{x}$.

The deterministic computer simulator of interest is represented by the $g$-function $y = g(\boldsymbol{x})$ with $y$ being the univariate model response. One should note that the developments in this work are also applicable to multivariate response cases, but only the univariate case is considered for simplicity. The target is then to estimate the resultant probabilistic characters of the model response $y$ such as the expectation and variance. However, due to the epistemic uncertainty presented in the model inputs, those statistics are no longer deterministic values, but interval quantities with bounds depending on the bounds of $\boldsymbol{\theta}$. We take the response expectation as an example, and it can be expressed as a $n$-dimensional integral, i.e.,

$$m(\boldsymbol{\theta}) = \int g(\boldsymbol{x}) f(\boldsymbol{x}|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x}. \tag{1}$$

With the above setting, the aleatory uncertainty present in model inputs is characterized by the probability distribution of $\boldsymbol{x}$, while that present in the model output is characterized by the probability distribution of $y$. The epistemic uncertainty of model inputs due to lack of information is modeled by the interval model of $\boldsymbol{\theta}$, and that propagated to the model output is described by the resultant interval of $m(\boldsymbol{\theta})$. For the second-order origin moment, one needs only to replace $g(\boldsymbol{x})$ with $g^2(\boldsymbol{x})$ in Eq. (1), thus below we only consider the estimation of response expectation.

The target of this work is then to learn the behavior of the function $m(\boldsymbol{\theta})$ visibly, especially when the epistemic uncertainty is large, and thus the intervals $\left[\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}\right]$ comprise a wide support, for which case, the NISS methods may exhibit poor performance.

## 3. Bayesian probabilistic Integration

We consider the integration of the function $g(\boldsymbol{u})$ with respect to the $n$-dimensional arguments $\boldsymbol{u}$ under independent standard Gaussian density weight, and we denote this integral as $\Pi[g(\boldsymbol{u})]$. The probabilistic integration rule is applied over a properly-trained (Bayesian) GPR model for $g(\boldsymbol{u})$, thus it is necessary to introduce the GPR model in advance.

Without any prior information on the behavior of $g(\boldsymbol{u})$, it is assumed that $g(\boldsymbol{u})$ is represented by a GPR model $\mathcal{GP}(b(\boldsymbol{u}), \kappa(\boldsymbol{u}, \boldsymbol{u}'))$, where $b(\boldsymbol{u})$ is the prior mean function which can be assumed to be any explicit form such as zero, constant and linear, and $\kappa(\boldsymbol{u}, \boldsymbol{u}')$ is the covariance function, which is also called kernel function, representing the strength of spatial correlation between the two sites $\boldsymbol{u}$ and $\boldsymbol{u}'$. Both the forms of mean function and kernel function reflect the prior knowledge of the behavior of the $g$-function. Many forms of kernel function have been developed, one can refer to Ref. [25] for more information. The one used in this work is the squared exponential kernel formulated as:

$$\kappa(\boldsymbol{u}, \boldsymbol{u}') = \sigma_0^2 \exp\left(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}')\Sigma^{-1}(\boldsymbol{u} - \boldsymbol{u}')^{\top}\right) \tag{2}$$

, where $\sigma_0^2$ is the hyper-parameter used for characterizing the (epistemic) variance at each site,

and $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ with $\sigma_i (i = 1, \ldots, n)$ being the length scale of $u_i$ and representing the strength of correlation along the *i*-th dimension.

Let $\mathcal{U}$ denote the training sample matrix with each row being a sample vector of $\boldsymbol{u}$, and $\mathcal{Y}$ indicates the column-wise vector of *g*-function value, i.e., $\mathcal{Y} = g(\mathcal{U})$. Then, based on the training data, the hyper-parameters in $b(\boldsymbol{u})$ and $\kappa(\boldsymbol{u}, \boldsymbol{u}')$ can be inferred by maximizing the likelihood function. The values of those hyper-parameters reflect information learned from the training data for the GPR model. Let $\hat{g}(\boldsymbol{u})$ denote the properly-trained GPR model, where the posterior mean at the site $\boldsymbol{u}$ is formulated as [25]:

$$\mathbb{E}_{\mathcal{D}}[\hat{g}(\boldsymbol{u})] = b(\boldsymbol{u}) + \boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top} K^{-1} (\mathcal{Y} - \boldsymbol{b}(\mathcal{U})) \tag{3}$$

, and the posterior covariance between two the sites $\boldsymbol{u}$ and $\boldsymbol{u}'$ is expressed as:

$$\mathrm{cov}_{\mathcal{D}}[\hat{g}(\boldsymbol{u}), \hat{g}(\boldsymbol{u}')] = \kappa(\boldsymbol{u}, \boldsymbol{u}') - \boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top} K^{-1} \boldsymbol{\kappa}(\boldsymbol{u}', \mathcal{U}) \tag{4}$$

, where $\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})$ is a column vector with the *i*-th element being $\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U}_i)$, $\mathcal{U}_i$ indicates the *i*-th row of $\mathcal{U}$, and $K$ is a matrix with the $(i, j)$-th component being $\kappa(\mathcal{U}_i, \mathcal{U}_j)$.

Based on Eq. (3) and (4), the integration $\Pi[\hat{g}(\boldsymbol{u})]$ for approximating the integral $\Pi[g(\boldsymbol{u})]$ is a Gaussian random variable with the posterior expectation and variance formulated by [21][23]:

$$\mathbb{E}_{\mathcal{D}}[\Pi(\hat{g}(\boldsymbol{u}))] = \Pi[b(\boldsymbol{u})] + \Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top}] K^{-1} (\mathcal{Y} - \boldsymbol{b}(\mathcal{U})) \tag{5}$$

, and

$$\mathbb{V}_{\mathcal{D}}[\Pi(\hat{g}(\boldsymbol{u}))] = \Pi\Pi[\kappa(\boldsymbol{u}, \boldsymbol{u}')] - \Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top}] K^{-1} \Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})] \tag{6}$$

, where $\Pi\Pi[\kappa(\boldsymbol{u}, \boldsymbol{u}')]$ represents the integral of $\kappa(\boldsymbol{u}, \boldsymbol{u}')$ with respect to both arguments under independent standard Gaussian density weights $\phi(\boldsymbol{u})$ and $\phi(\boldsymbol{u}')$. To generate the analytical values of Eq. (5) and Eq.(6), one needs to derive the closed-form expression for the three integrals $\Pi[b(\boldsymbol{u})]$, $\Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top}]$ and $\Pi\Pi[\kappa(\boldsymbol{u}, \boldsymbol{u}')]$. For the commonly used zero, constant, linear and even polynomial mean functions, the derivation of the analytical value of $\Pi[b(\boldsymbol{u})]$ is quite trivial, and we don't give details. The closed-form expressions for $\Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top}]$ and $\Pi\Pi[\kappa(\boldsymbol{u}, \boldsymbol{u}')]$ are formulated as [23]:

$$\Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})] = \sigma_0^2 |\Sigma^{-1} + I|^{-1/2} \exp\left[-\frac{1}{2} \mathrm{vec}\{\mathrm{diag}[\mathcal{U}(\Sigma + I)^{-1} \mathcal{U}^{\top}]\}\right] \tag{7}$$

, and

$$\Pi\Pi[k(\boldsymbol{u}, \boldsymbol{u}')] = \sigma_0^2 |2\Sigma^{-1} + I|^{-1/2} \tag{8}$$

, where $|\cdot|^{-1/2}$ indicates the inverse of the square root of the determinant of the argument, $I$ is the identity matrix with the diagonal elements being one and the other elements being zero, $\mathrm{vec}\{\mathrm{diag}[\cdot]\}$ indicates a column-wise vector generated with the diagonal elements of the argument matrix. The squared exponential kernel in Eq. (2) and the Gaussian density weight is one of the kernel-distribution pairs resulting in the closed-form expressions of both $\Pi[\boldsymbol{\kappa}(\boldsymbol{u}, \mathcal{U})^{\top}]$ and $\Pi\Pi[k(\boldsymbol{u}, \boldsymbol{u}')]$, which is a desirable property for probabilistic integration. There are also other kernel-distribution pairs satisfying this property, and one can refer to Ref. [25] for more details. In this work, all the developed algorithms are implemented with standard Gaussian density weight, thus only the squared exponential kernel is used. However, all the developments apply to

any type of probability distribution, as will be shown in the next section.

## 4. Probabilistic response function estimation

The uncertainty propagation task in this work mainly concerns the estimation of the **response expectation function (REF)** $m(\boldsymbol{\theta})$ defined by Eq. (1). For higher-order moment functions, the theoretical details are similar, and thus, they are not presented for the sake of brevity.

### 4.1. High-dimensional model representation

As mentioned above, the integrals in this work are all assumed to have standard Gaussian density weight, thus it is necessary to transform both $\boldsymbol{x}$ and $\boldsymbol{\theta}$ into standard Gaussian variables. This can be achieved by using Rosenblatt's transformation [26]. Take univariate input $x$ with distribution function $F(x|\theta)$ and scalar non-deterministic distribution parameter $\theta$ as an example. The auxiliary distribution function for $\theta$ is denoted as $P(\theta)$. Then $\theta$ can be transformed to a standard Gaussian variable $v$ by $\Phi(v)=P(\theta)$, and $x$ can be also transformed to standard Gaussian variable $u$ by $F(x|P^{-1}(\Phi(v)))=\Phi(u)$. In the multivariate case, the transformation pairs for $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are formulated as:

$$\begin{cases} \boldsymbol{\theta}=T(\boldsymbol{v})=P^{-1}[\Phi(\boldsymbol{v})] \\ \boldsymbol{v}=T^{-1}(\boldsymbol{\theta})=\Phi^{-1}[P(\boldsymbol{\theta})] \end{cases} \tag{9}$$

, and

$$\begin{cases} \boldsymbol{x}=S(\boldsymbol{u}|\boldsymbol{\theta})=S(\boldsymbol{u}|T(\boldsymbol{v}))=F^{-1}[\Phi(\boldsymbol{u})|T(\boldsymbol{v})] \\ \boldsymbol{u}=S^{-1}(\boldsymbol{u}|\boldsymbol{\theta})=S^{-1}(\boldsymbol{x}|T(\boldsymbol{v}))=\Phi^{-1}[F(\boldsymbol{x}|T(\boldsymbol{v}))]. \end{cases} \tag{10}$$

Let $\Pi_{1:n}(\cdot)$ denote the *n*-dimensional integral of a $(n+d)$-dimensional function with respect to its first *n* inputs with independent standard Gaussian weight. Then the conditional expectation in Eq. (1) can be rewritten as:

$$m(\boldsymbol{\theta})=\Pi_{1:n}[\mathcal{G}_1(\boldsymbol{u},\boldsymbol{\theta})] \tag{11}$$

, or further as

$$\mathcal{M}(\boldsymbol{v})=m(T(\boldsymbol{v}))=\Pi_{1:n}[\mathcal{G}_2(\boldsymbol{u},\boldsymbol{v})] \tag{12}$$

, where $\mathcal{G}_1(\boldsymbol{u},\boldsymbol{\theta})=g(S(\boldsymbol{u}|\boldsymbol{\theta}))$ and $\mathcal{G}_2(\boldsymbol{u},\boldsymbol{v})=g(S(\boldsymbol{u}|T(\boldsymbol{v})))$. With any one of $m(\boldsymbol{\theta})$ and $\mathcal{M}(\boldsymbol{v})$ being estimated properly, the other one can be easily derived.

In NISS methods, the REF is decomposed by High-Dimensional Model Representation (HDMR) into the summation of a set of functional components with input dimensions varying from 1 to *d*. This scheme brings at least two benefits. First, since the REF is mostly governed by low order component functions, the explicit estimation of these low order component function enables the analysts to learn the behavior of REF visibly. Second, the relative importance of each $\theta_i$ can be learned from those component functions, and provides useful information for directing the future information collection (so as to further reduce the epistemic uncertainty of model response), and also for reducing the dimensionality of the optimization procedure used for computing the bounds of model response expectation. In this subsection, we follow this idea to estimate the component functions by probabilistic integration.

In the NISS framework, two HDMR schemes are utilized, where the cut-HDMR decomposition is used in local NISS methods, while the RS (Random Sampling)-HDMR is utilized in global NISS methods [15][16]. In this work, only the RS-HDMR is used, thus below we use HDMR to denote RS-

HDMR. The HDMR decomposition of the REF $\mathcal{M}(\boldsymbol{v})$ is formulated as [27]:

$$\mathcal{M}(\boldsymbol{v}) = \mathcal{M}_0 + \sum_{i=1}^{d} \mathcal{M}_i(v_i) + \sum_{i=j+1}^{d} \sum_{j=1}^{d-1} \mathcal{M}_{ij}(\boldsymbol{v}_{ij}) + \cdots + \mathcal{M}_{1:n}(\boldsymbol{v}) \tag{13}$$

, where $\boldsymbol{v}_{ij} = (v_i, v_j)$, and

$$\begin{aligned} \mathcal{M}_0 &= \Pi[\mathcal{M}(\boldsymbol{v})] = \Pi[\mathcal{G}_2(\boldsymbol{u},\boldsymbol{v})] \\ \mathcal{M}_i(v_i) &= \Pi_{-i}[\mathcal{M}(\boldsymbol{v})] - \mathcal{M}_0 = \Pi_{-i}[\mathcal{G}_2(\boldsymbol{u},\boldsymbol{v})] - \mathcal{M}_0 \\ \mathcal{M}_{ij}(\boldsymbol{v}_{ij}) &= \Pi_{-ij}[\mathcal{M}(\boldsymbol{v})] - \mathcal{M}_i - \mathcal{M}_j - \mathcal{M}_0 = \Pi_{-ij}[\mathcal{G}_2(\boldsymbol{u},\boldsymbol{v})] - \mathcal{M}_i - \mathcal{M}_j - \mathcal{M}_0 \end{aligned} \tag{14}$$

, with $\Pi_{-i}(\cdot)$ indicating the integral with respect to all input variables of the integrand except $v_i$, and $\Pi_{-ij}(\cdot)$ being integral with respect to all arguments of the integral except $\boldsymbol{v}_{ij} = (v_i, v_j)$.

### 4.2. Bayesian Inference of HDMR components

Next, consider the estimation of $\mathcal{M}(\boldsymbol{v})$ as well as its HDMR components. For ease of illustration, let $\boldsymbol{w} = (\boldsymbol{u},\boldsymbol{v})$. Given a set of training samples $(\mathcal{W},\mathcal{Y})$ with each row of $\mathcal{W}$ being a sample point of $\boldsymbol{w}$ and each component of the column-wise vector of $\mathcal{Y}$ being the corresponding value of model response function $\mathcal{G}_2(\boldsymbol{w})$. Then a GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ can be trained with the posterior mean and posterior variance at any new site $\boldsymbol{w}$ being:

$$\mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{G}}_2(\boldsymbol{w})\right] = b(\boldsymbol{w}) + \boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top} K^{-1}(\mathcal{Y} - \boldsymbol{b}(\mathcal{W})) \tag{15}$$

, and

$$\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{G}}_2(\boldsymbol{w})\right] = \kappa(\boldsymbol{w},\boldsymbol{w}) - \boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top} K^{-1} \boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W}) \tag{16}$$

, where $b(\boldsymbol{w})$ is the prior basis function, $\kappa(\boldsymbol{w},\boldsymbol{w}')$ is the kernel function, and $K$ is the covariance matrix for training data $\mathcal{W}$ computed with the kernel $\kappa(\boldsymbol{w},\boldsymbol{w}')$.

◆ **Constant HDMR Component**

Eq. (14) reveals that the constant HDMR component $\mathcal{M}_0$ is actually a $(n+d)$-dimensional integral with respect to all elements of $\boldsymbol{w}$, the estimation of $\mathcal{M}_0$ resulting from the GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ is a Gaussian random variable, and the posterior mean and posterior variance are formulated as:

$$\mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] = \mathbb{E}_{\mathcal{D}}\left[\Pi\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] = \Pi[b(\boldsymbol{w})] + \Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top}] K^{-1}(\mathcal{Y} - \boldsymbol{b}(\mathcal{W})) \tag{17}$$

, and

$$\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] = \mathbb{V}_{\mathcal{D}}\left[\Pi\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] = \Pi\Pi[\kappa(\boldsymbol{w},\boldsymbol{w}')] - \Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top}] K^{-1} \Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]. \tag{18}$$

The closed-form expressions for the integrals $\Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]$ and $\Pi\Pi[\kappa(\boldsymbol{w},\boldsymbol{w}')]$ in Eq. (17) and Eq. (18) can be similarly formulated as (7) and (8), and we don't repeat them.

For the first- and second- order HDMR components, the posterior estimations resulting from the GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ are one- and two- dimensional GP models, respectively, and the posterior means and expectations can be analytically derived from $\hat{\mathcal{G}}_2(\boldsymbol{w})$.

◆ **First-order HDMR Component Functions**

The posterior mean of the first-order HDMR component is formulated as:

$$\mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_i(v_i)\right] = \Pi_{-i}[b(\boldsymbol{w})] + \Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^\top]K^{-1}(\mathcal{Y} - \boldsymbol{b}(\mathcal{W})) - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] \tag{19}$$

, where the derivation of $\Pi_{-i}[b(\boldsymbol{w})]$ is trivial, and we don't give details, the closed-form expression of the integral $\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^\top]$ is formulated as:

$$\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] = \sigma_0^2 |\Sigma_{-i}^{-1} + I|^{-1/2} \exp\begin{bmatrix} -\frac{1}{2}\mathrm{vec}\{\mathrm{diag}[\mathcal{W}_{,-i}(\Sigma_{-i} + I)^{-1}\mathcal{W}_{,-i}^\top]\} \\ -\frac{1}{2\sigma_{v,i}^2}(v_i - \mathcal{W}_{,i})^2 \end{bmatrix} \tag{20}$$

, with $\mathcal{W}_{,i}$ being the column-wise sample of $v_i$, $\mathcal{W}_{,-i}$ indicating the sample matrix generated by removing $\mathcal{W}_{,i}$ from $\mathcal{W}$, $\sigma_{v,i}$ referring to the length scale hyper-parameter of $v_i$, and $\Sigma_{-i}$ denoting the diagonal matrix generated by removing the $(n+i)$-th column and $(n+i)$-th row from $\Sigma$.

The posterior variance of the first-order component is given as:

$$\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_i(v_i)\right] = \mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] + \mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] - 2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right), \hat{\mathcal{M}}_0\right] \tag{21}$$

, where $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]$ is given in Eq. (18), and $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ is formulated as:

$$\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left[\hat{\mathcal{G}}_2(\boldsymbol{w})\right]\right] = \Pi_{-i}\Pi_{-i}[\kappa(\boldsymbol{w}, (\boldsymbol{w}'_{-i}, v_i))] - \Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^\top]K^{-1}\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] \tag{22}$$

, where $(\boldsymbol{w}'_{-i}, v_i)$ represents the (*n*+*d*)-dimensional row vector equal to $\boldsymbol{w}'$, except its (*n*+*i*)-th component, which is equal to $v_i$, $\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^\top]$ has been given in Eq. (20), and

$$\Pi_{-i}\Pi_{-i}[\kappa(\boldsymbol{w}, (\boldsymbol{w}'_{-i}, v_i))] = \sigma_0^2 |2\Sigma_{-i}^{-1} + I|^{-1/2}. \tag{23}$$

The posterior covariance in Eq. (21) is given by:

$$\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right), \hat{\mathcal{M}}_0\right] = \Pi\Pi_{-i}[\kappa(\boldsymbol{w}, \boldsymbol{w}')] - \Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]^\top K^{-1}\Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] \tag{24}$$

, where $\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]$ is given by Eq. (20), and $\Pi\Pi_{-i}[\kappa(\boldsymbol{w}, \boldsymbol{w}')]$ indicates the integral with respect to $\boldsymbol{w}_{-i}$ and $\boldsymbol{w}'$, which is explicitly formulated as

$$\Pi\Pi_{-i}[\kappa(\boldsymbol{w}, \boldsymbol{w}')] = \sigma_0^2 |2\Sigma_{-i}^{-1} + I|^{-1/2} (\sigma_{v,i}^{-2} + 1)^{-1/2} \exp\left(-\frac{1}{2(\sigma_{v,i}^2 + 1)} v_i^2\right). \tag{25}$$

The mathematical derivations of Eqs. (21)-(25) are given in Appendix A.

◆ **Second-order HDMR Component**

The second-order HDMR component $\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})$ resulting from the GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ is then a two-dimensional GPR model. The posterior mean is formulated as:

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})\right] = {} & \Pi_{-ij}[b(\boldsymbol{w})] + \Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^\top]K^{-1}(\mathcal{Y} - \boldsymbol{b}(\mathcal{W})) \\ & - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_i(v_i)\right] - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_j(v_j)\right] - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] \end{aligned} \tag{26}$$

, where $\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]$ is explicitly formulated as:

$$\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] = \sigma_0^2 |\Sigma_{-ij}^{-1} + I|^{-1/2} \exp\begin{bmatrix} -\frac{1}{2}\mathrm{vec}\{\mathrm{diag}[\mathcal{W}_{,-ij}(\Sigma_{-ij} + I)^{-1}\mathcal{W}_{,-ij}^\top]\} \\ -\frac{1}{2}(\boldsymbol{v}_{ij} - \mathcal{W}_{,ij})^\top \Sigma_{ij}^{-1}(\boldsymbol{v}_{ij} - \mathcal{W}_{,ij}) \end{bmatrix} \tag{27}$$

, with $\Sigma_{ij}$ indicating the $(2\times 2)$-dimensional diagonal matrix composed of $\sigma_{v,i}^2$ and $\sigma_{v,j}^2$, $\Sigma_{-ij}$ indicates the $(n+d-2)\times(n+d-2)$-dimensional diagonal matrix generated by removing $\Sigma_{ij}$

from $\Sigma$, $\mathcal{W}_{,ij}$ is the sample matrix of $\boldsymbol{v}_{ij}$, $\mathcal{W}_{,-ij}$ refers to the sample matrix of $\boldsymbol{w}_{-ij}$, and $\boldsymbol{w}_{-ij}$ indicates the $(n+d-2)$-dimensional vector generated by removing $\boldsymbol{v}_{ij}$ from $\boldsymbol{w}$.

The closed-form expression of the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})\right]$ is cumbersome, and for simplification, it is listed in Appendix B.

Based on the above formulas, we can generate the posterior means and variances for all the zero-, first- and second- order HDMR component functions of the probabilistic response function $\mathcal{M}(\boldsymbol{v})$, as by-products of the properly-trained GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$, which are commonly sufficient for understanding the behavior of $\mathcal{M}(\boldsymbol{v})$. However, if necessary, higher-order HDMR component functions can be similarly derived, but we don't go further for simplicity. By replacing $v_i$ in $\hat{\mathcal{M}}_i(v_i)$ with $v_i = T^{-1}(\theta_i)$, we can generate the posterior mean and variance of the first-order HDMR component of $\theta_i$, which can be very useful for learning the behavior of $m(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

## 5. Adaptive Experiment Design

It is unquestionably that the performance of the probabilistic integration relies on the training data set $\mathcal{D}$ used for training the GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$. A simple way to create this data set is simple random sampling or quasi-random samplings such as Latin-hypercube sampling [28] and Sobol' low-discrepancy sequence [29]. However, all those random sampling design strategies do not take the behavior of *g*-function into consideration, thus, of course, cannot provide the optimal design for the training data. In principle, given a small set of training data, the properly-trained GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ delivers information on the contribution of the GPR prediction error to the posterior variance of the integration. This kind of information can be valuable for identifying the design point that, by adding it to the training data provides the greatest reduction on the posterior variance of the integration. Based on this idea, we introduce an adaptive experiment design strategy for actively generating training data.

The core of this algorithm is the so-called learning function. In this paper, we use the learning function we defined in our previous work [23]. The definition of the learning function is formulated as:

$$h(\boldsymbol{w}) = h^{+}(\boldsymbol{w}) \cdot \phi(\boldsymbol{w}) \tag{28}$$

, where

$$\begin{aligned} h^{+}(\boldsymbol{w}) &= \Pi'\left[\mathrm{cov}_{\mathcal{D}}\left(\hat{\mathcal{G}}_2(\boldsymbol{w}), \hat{\mathcal{G}}_2(\boldsymbol{w}')\right)\right] \\ &= \Pi'[\kappa(\boldsymbol{w}, \boldsymbol{w}')] - \boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top} K^{-1} \Pi'[\boldsymbol{\kappa}(\boldsymbol{w}', \mathcal{W})] \end{aligned} \tag{29}$$

, with $\Pi'[\cdot]$ indicating the integral with respect to $\boldsymbol{w}'$. From Eq. (18), it is seen that the following equation holds:

$$\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] = \Pi[h^{+}(\boldsymbol{w})] = \int h(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}. \tag{30}$$

Eqs. (28)-(30) reveal that the learning function $h(\boldsymbol{w})$ measures the contribution of the prediction error at the point $\boldsymbol{w}$ to the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]$, with the consideration of its spatial correlation information over all the other points in both the aleatory and epistemic space. We name this kind of information as "contribution to posterior variance" at point $\boldsymbol{w}$, and the

learning function as Posterior Variance Contribution (PVC) function. Then, by adding the point $\boldsymbol{w}$ with the maximum value of $h(\boldsymbol{w})$ to the training data $\mathcal{D}$, it is expected that the greatest reduction of the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]$ can be achieved. The closed-form expression of the learning function can be derived. The expression for the integral $\Pi'[\boldsymbol{\kappa}(\boldsymbol{w}', \mathcal{W})]$ in Eq. (29) is similar to Eq. (7). The closed-form expression for $\Pi'[\kappa(\boldsymbol{w}, \boldsymbol{w}')]$ is formulated as:

$$\Pi'[k(\boldsymbol{w}, \boldsymbol{w}')] = \sigma_0^2 |\Sigma^{-1} + I|^{-1/2} \exp\left[-\frac{1}{2}\boldsymbol{w}(\Sigma + I)^{-1}\boldsymbol{w}^{\top}\right]. \tag{31}$$

The learning function is usually multi-modal, thus it is recommended to search the design point $\boldsymbol{w}^+ = \operatorname{argmax} h(\boldsymbol{w})$ by using a global optimization algorithm. In this paper, the particle swarm code in the Matlab Global Optimization Toolbox is utilized.

Based on the PVC function, an adaptive experiment design strategy is developed, and the associated flowchart for implementation is schematically shown in Fig. 1. It involves first creating a small set of training data $\mathcal{D}$ with which a GPR model can be trained, and the corresponding learning function can be analytically derived. Then, the point $\boldsymbol{w}^+$ can be specified as $\boldsymbol{w}^+ = \operatorname{argmax} h(\boldsymbol{w})$ and added to the training data set $\mathcal{D}$. The above process is implemented repeatedly until a stopping criterion is satisfied. In this paper, the stopping criterion is selected as the posterior Coefficient Of Variation (COV) of $\hat{\mathcal{M}}_0$ being less than a small value $\varepsilon$, i.e., $\sqrt{\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]} \Big/ \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] < \varepsilon$. However, it is found that the above stopping criteria may result in posterior distribution with small variation but large bias when the initial training data size is too small. To avoid this situation, it is suggested to use a delayed judgment strategy, which means finishing the algorithm only when the above stopping criteria is satisfied for several (e.g., three) times in succession.

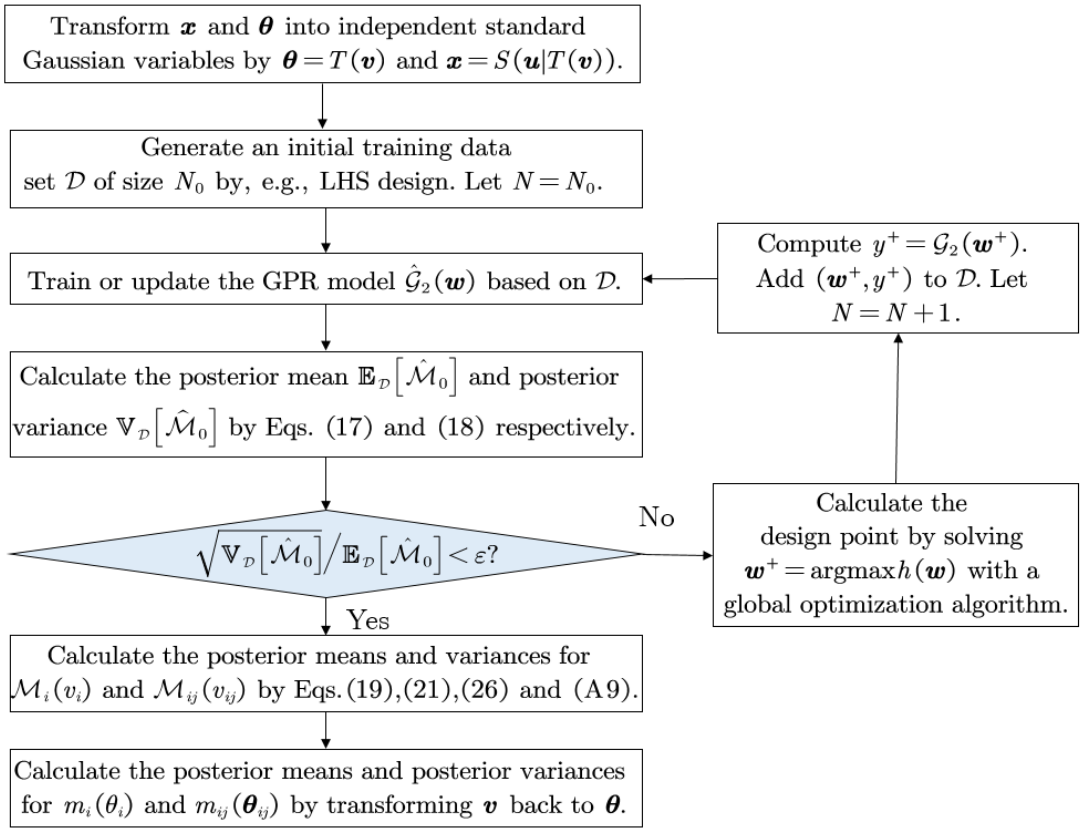


Figure 1 Flowchart of the adaptive experiment design.

## 6. Test Examples

### 6.1. General remarks

For numerical implementation, the auxiliary probability distribution $P(\boldsymbol{\theta})$ should be pre-assumed for the non-deterministic distribution parameters $\boldsymbol{\theta}$ so as to create training data for both $\boldsymbol{u}$ and $\boldsymbol{v}$. Similar to NISS method [18], any type of probability distribution can be utilized, and the type of probability distribution will not affect the behavior of $\mathcal{M}(\boldsymbol{v})$, but only affects its HDMR component functions as well as their posterior estimation. The optimal probability distribution for $P(\boldsymbol{\theta})$ definitely depends on the behavior of the *g*-function, which is beyond the scope of this paper. Without loss of generality, it is assumed that the elements of $\boldsymbol{\theta}$ are independent, and follow uniform distribution. It is found that, by setting the support of this auxiliary distribution, used for generating the training data, larger than the real support $\left[\underline{\boldsymbol{\theta}},\overline{\boldsymbol{\theta}}\right]$, the performance of estimation can be improved for the area around the bounds. Therefore, the support of the auxiliary distribution is can be relaxed as $\left[\underline{\boldsymbol{\theta}}-\epsilon_1,\overline{\boldsymbol{\theta}}+\epsilon_2\right]$, where $\epsilon_1$ and $\epsilon_2$ are both positive parameters, and can be attributed with any values that make $-2.2\leqslant\Phi^{-1}\left(U\left(\underline{\boldsymbol{\theta}}|\underline{\boldsymbol{\theta}}-\epsilon_1,\overline{\boldsymbol{\theta}}+\epsilon_2\right)\right)\leqslant-1.5$ and $1.5\leqslant\Phi^{-1}\left(U\left(\overline{\boldsymbol{\theta}}|\underline{\boldsymbol{\theta}}-\epsilon_1,\overline{\boldsymbol{\theta}}+\epsilon_2\right)\right)\leqslant2.2$ hold, with $U\left(\cdot|\underline{\boldsymbol{\theta}}-\epsilon_1,\overline{\boldsymbol{\theta}}+\epsilon_2\right)$ being the CDF of uniform distribution bounded by $\left[\underline{\boldsymbol{\theta}}-\epsilon_1,\overline{\boldsymbol{\theta}}+\epsilon_2\right]$, and $\Phi^{-1}(\cdot)$ indicating the CDF of standard Gaussian distribution. For all the three test examples in this section, we use the constant mean function $b(\boldsymbol{u})=b_0$ for the GPR model.

### 6.2. Illustrative example

We use the same toy example as in Ref. [15] to illustrate the proposed NIPI method, and to compare its performance with the global NISS method. The $g$-function of this toy example is formulated as:

$$y = g(\boldsymbol{x}) = 1 - \frac{(x_1 - 1)^2}{a^2} - \frac{(x_2 - 1)^3}{b^2} \tag{32}$$

, where $a = 3$, $b = 4$, $x_1$ and $x_2$ are both Gaussian random variables with non-deterministic distribution parameters. The mean parameters of the two variables are denoted as $\mu_1$ and $\mu_2$, both of which have interval support $[-3, 3]$, and the STandard Deviation (STD) parameters are indicated by $\sigma_1$ and $\sigma_2$ with the support being $[0.5, 3]$. One should note that, with the above setting, the epistemic uncertainty of the model inputs is much larger than that considered in Ref. [15].

We first consider the NIPI method without active learning. For the numerical implementation, $\epsilon_1$ and $\epsilon_2$ are both set to be five percent of the bound length $\bar{\theta}_i - \underline{\theta}_i$ for each dimension of $\boldsymbol{\theta}$ for both NIPI and NISS. The training data size for NIPI is set to be 200, and the sample size for NISS is set to be 2,000. For both methods, the samples are generated by LHS design. We first discuss the results of the constant HDMR component $\mathcal{M}_0$, which are shown in Figure 2. One note that the NISS estimators also follow Gaussian distribution, thus the results of both methods are shown with Gaussian density. Obviously, the posterior mean of NIPI is closer to the true value than the mean estimation of NISS, but the posterior variance of NIPI is much smaller than that of NISS. This indicates that NIPI provides much better estimation than NISS, for the constant HDMR component, although its computational cost is only one-tenth of that of NISS.

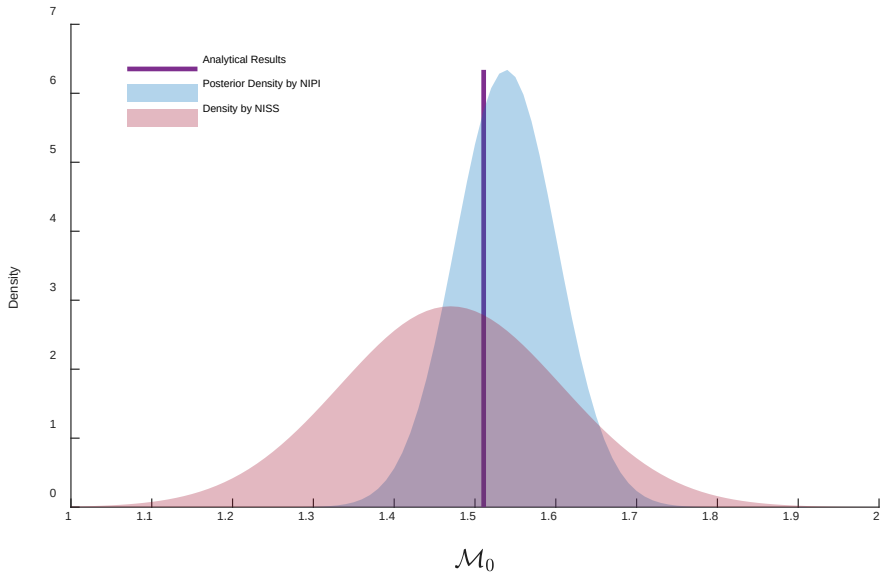


Figure 2 Results for constant HDMR component of the toy example.

The results of the first-order component functions generated by both NIPI and NISS are then reported in Figure 3, together with the analytical results for comparison. Both the mean estimates and the 99% Confidence Intervals (CIs) generated by both methods are reported. We first examine the mean estimates. Obviously, the posterior means produced with NIPI of all the four first-order component functions match very well with the analytical results over the whole supports.

Conversely, the mean estimates of NISS show good agreement with the analytical results only in some small regions. The above phenomenon suggests that NIPI provides much better mean estimates for all the first-order components. Next, we analyze the confidence intervals shown in Figure 3. Obviously, for all these four components, NIPI provides much tighter and more accurate confidence intervals in the whole region. It is also shown that, NISS provides very poor estimation near the edges of the epistemic intervals, especially for the component function of $\sigma_2$. This is caused by the large variation of the density ratio functions presented in NISS estimators. Whereas, in the NIPI estimator, such kind of ratio function does not exist. Based on the above analysis, it can be concluded that, if the input epistemic uncertainty has small support, both NISS and NIPI can provide satisfactory estimation, but for cases with large epistemic uncertainty, NIPI is much more powerful. Besides, NIPI is much more efficient than NISS for this example since the required number of $g$-function calls is only one-tenth of that of NISS.

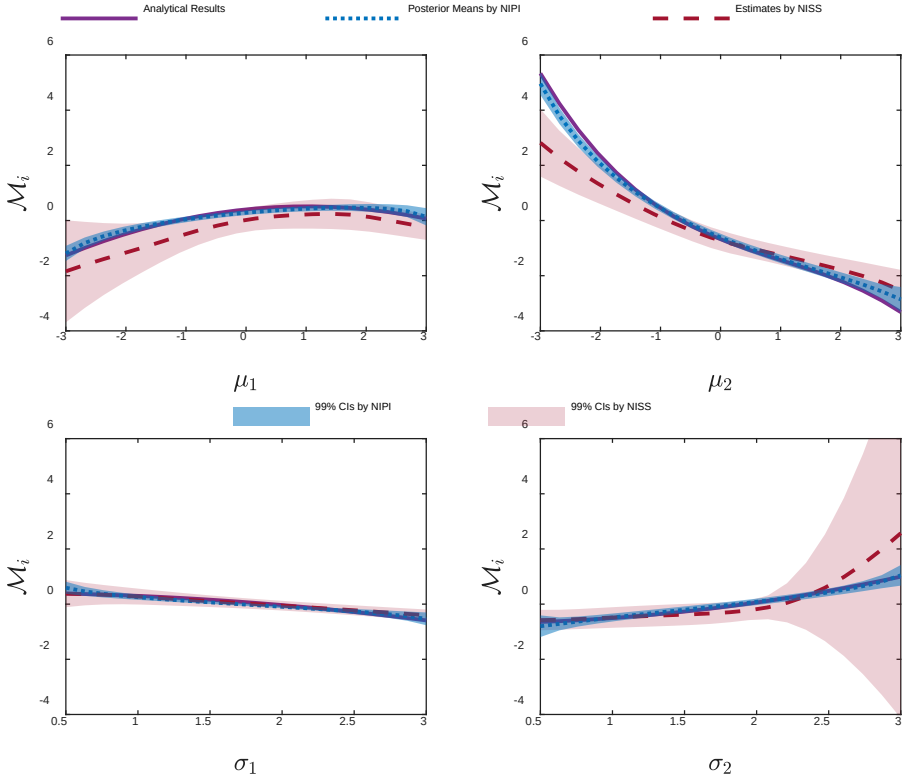


Figure 3 First-order HDMR component functions of the toy example.

The second-order HDMR components are then computed based on the same GPR model, and it is found that only the one involving $(\mu_2, \sigma_2)$ is influential, whose results are shown in Figure 4, together with the analytical results for comparison. We don't show the results of NISS since they are much poorer than those generated by NIPI. Figure 4 shows that the posterior mean matches well with the analytical result, and the posterior STD is very small, indicating the high accuracy of NIPI for estimating the second-order component functions.

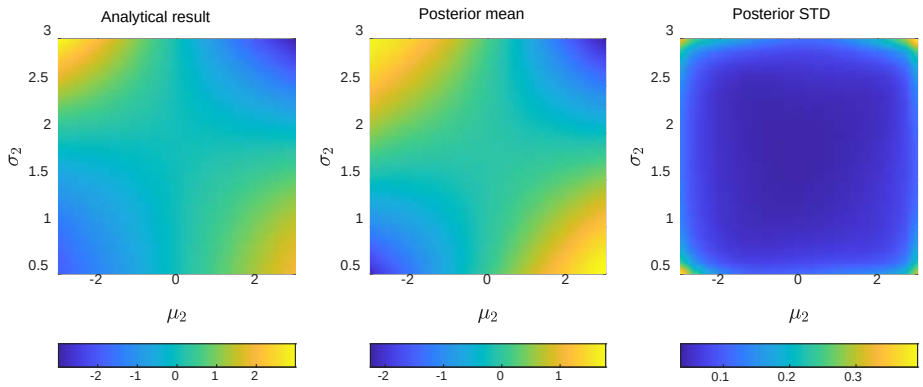


Figure 4 NIPI results of the influential second-order HDMR component function of $\mu_2$ and $\sigma_2$ for the toy example.

Next, we illustrate the potential of the adaptive experiment design for further improving the efficiency of NIPI. In Figure 2, the posterior COV of the NIPI estimate is computed to be 0.04, thus we take this value as the stopping criteria of the adaptive design. That is once the posterior COV being less than 0.04 for three times in succession, the adaptive algorithm stops. We start the adaptive design with 50 training samples generated by LHS design. With the above setting, the algorithm adaptively produces 35 more training data before touching the stopping criteria. The posterior density of the constant component generated with these adaptively designed points is compared with the one generated with 200 LHS design points in Figure 5. As can be seen, both probability distributions cover the analytical results with high confidence, indicating that both results are accurate. The posterior variance by adaptive design is a little bit smaller than that generated without adaptive design.

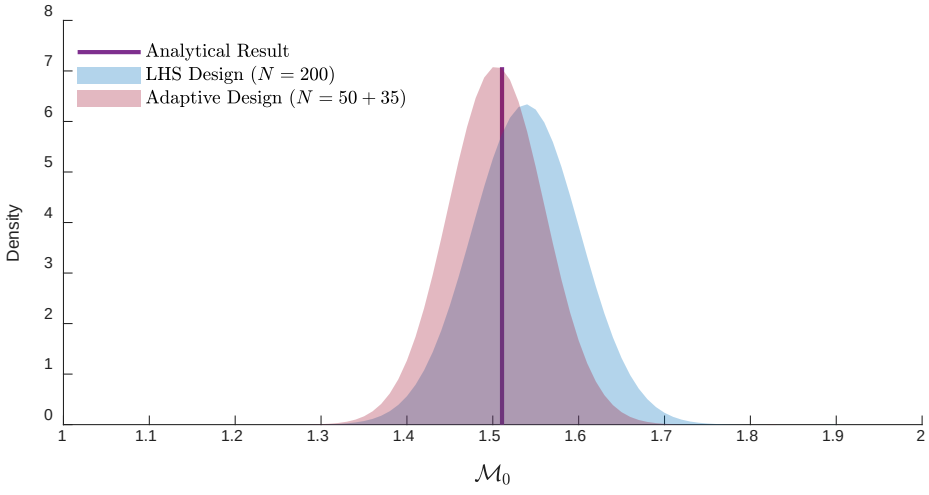


Figure 5 Posterior densities of the constant HDMR component of the toy example generated by NIPI with LHS design and adaptive design.

For illustrating the process of the adaptive experiment design, the evolution of the 99% confidence intervals and the COV of $\mathcal{M}_0$ with respect to the training sample size $N$ are shown in Figure 6. As a comparison, the evolution process of NIPI without adaptive design but with the same set of 50 initial training points is also schematically shown in Figure 6. It can be seen that the

adaptive NIPI converges much faster than the NIPI without adaptive design, indicating that the adaptive design can substantially reduce the number of required *g*-function calls. It is also shown in Figure 6 that, for adaptive NIPI, during the design process, the posterior variation does not decrease monotonically with respect to the increment of sample size. The stopping criterion can be satisfied when the posterior variance is small but the posterior mean shows large bias, although this situation is not seen often. However, if the posterior COV is less than the threshold (0.04) for several times in succession (see the part with $N > 80$), it is believed that the algorithm has converged with high confidence. Thus, by using the delayed judgment, the algorithm can be much more robust.

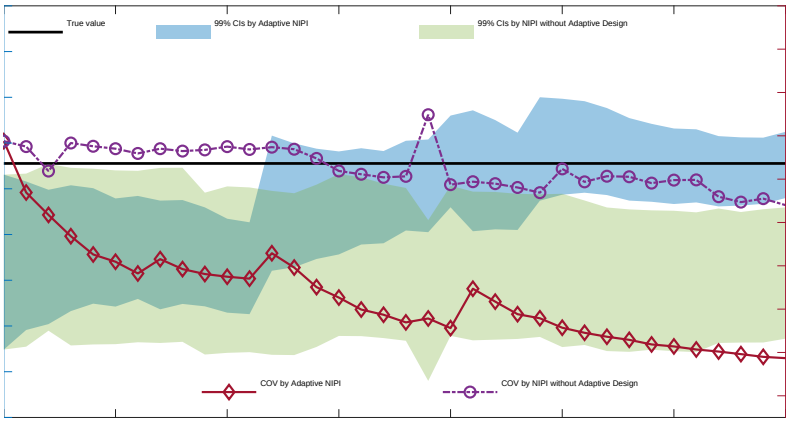


Figure 6 Evolution process of posterior confidence intervals and COVs of $\mathcal{M}_0$, where the left axis shows the evolution of the 99% confidence intervals and the right axis shows the evolution of the posterior COV.

The results of first-order component functions generated with and without adaptive design are then compared in Figure 7. It is shown that the steady-state curves are all bounded by the confidence intervals generated with both strategies, and it is hard to say which strategy produces better results.

Based on the above analysis, we believe that the results generated with and without adaptive design are of the same quality. However, by introducing the adaptive design strategy, the required number of *g*-function calls is reduced from 200 to 85, which can be quite substantial for real-world engineering applications where one *g*-function call takes several minutes or even several hours.

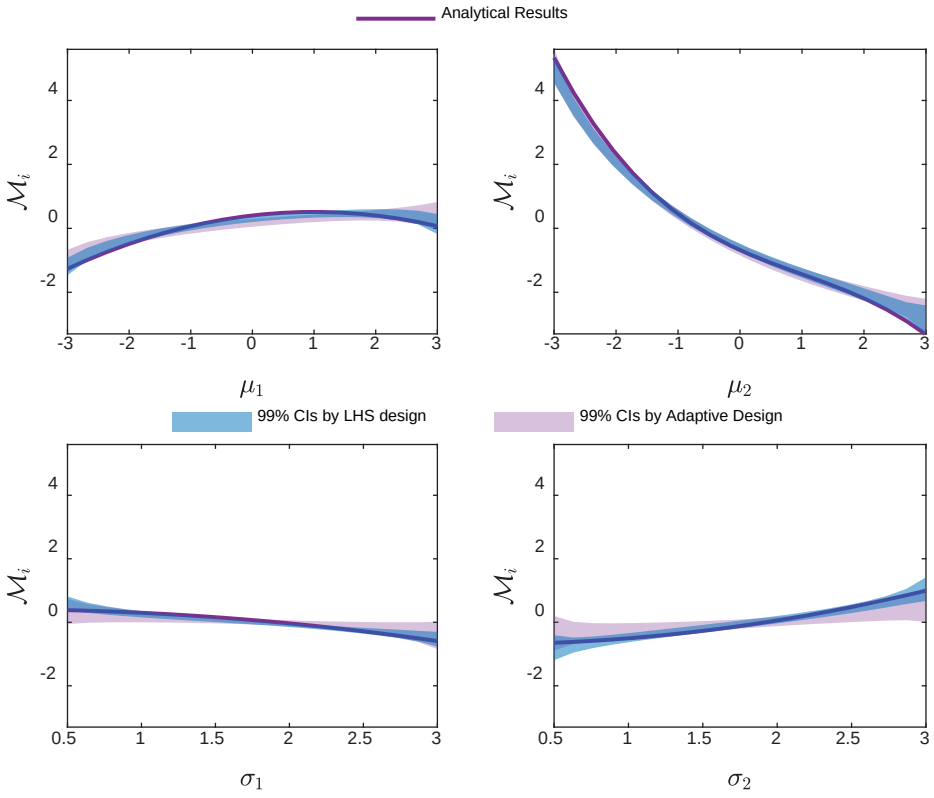


Figure 7 Comparison of first-order component functions results generated by NIPI with and without adaptive design.

### 6.3. A steady-state confined seepage model

Consider a model simulating the steady-state confined seepage below a dam adapted from Ref. [30]. The elevation of the dam is schematically shown in Figure 8. The soil below the dam consists of one impermeable layer and two permeable layers. In this example, the water height $h_D$ in the upstream side and the permeability properties of the two permeable layers are assumed to be input random variables. The water height $h_D$ is assumed to be a precise random variable following uniform distribution with support between 7 [m] and 10 [m]. The permeability properties are all assumed to be imprecise random variables characterized by lognormal *p*-box, and are described in Table 1. One can refer to Ref. [30] for the detailed description of this model. The governing PDE of this model is formulated as [30]:

$$k_{xx,i}\frac{\partial^2 h_W}{\partial x^2}+k_{yy,i}\frac{\partial^2 h_W}{\partial y^2}=0,\ i=1,2 \tag{33}$$

, where $h_W=h_D+20\ [\mathrm{m}]$ indicates the hydraulic head over segment AB. The above PDE is numerically solved by the finite element method with 3413 nodes and 1628 quadratic triangular elements. Once the above PDE being solved, the seepage $q$ at the downstream side of the dam, measured in units of volume over time over distance, can be calculated by:

$$q=-\int_{\mathrm{CD}} k_{yy,2}\frac{\partial h_W}{\partial y}\mathrm{d}x. \tag{34}$$

The target of this application is to learn the behavior of the function of the expectation $\mathbb{E}[q]$ with respect to the eight non-deterministic distribution parameters given in Table 1. The order of magnitude of the model response is relatively small (as expected for seepage problems), and for

simplicity, we do all the analysis by multiplying the model response $q$ by $10^6$.

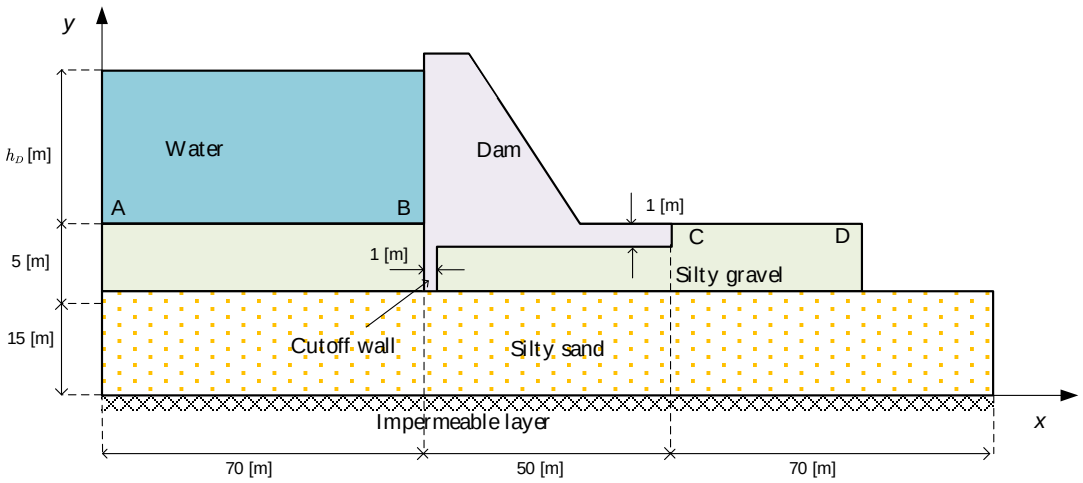


Figure 8 Schematic illustration of the dam.

Table 1 Description of the four imprecise random input variables of the seepage model.

| Input variables | Description | Distribution type | Mean of logarithmic values ($[\log([\mathrm{m/s}])]$) | STD of logarithmic values ($[\log([\mathrm{m/s}])]$) |
|---|---|---|---|---|
| $k_{xx,1}$ | Horizontal permeability of silty sand layer | Lognormal | $\mu_1 \in [-16.2297, -14.0464]$ | $\sigma_1 \in [0.1245, -1.2686]$ |
| $k_{yy,1}$ | Vertical permeability of silty sand layer | Lognormal | $\mu_2 \in [-16.2297, -14.0464]$ | $\sigma_2 \in [0.1245, -1.2686]$ |
| $k_{xx,2}$ | Horizontal permeability of silty gravel layer | Lognormal | $\mu_3 \in [-16.2297, -14.0464]$ | $\sigma_3 \in [0.1245, -1.2686]$ |
| $k_{yy,2}$ | Vertical permeability of silty gravel layer | Lognormal | $\mu_4 \in [-16.2297, -14.0464]$ | $\sigma_4 \in [0.1245, -1.2686]$ |

The analytical results for this example are unavailable due to the implicit model response function, thus we use the Double-Loop Monte Carlo Simulation (DLMCS) procedure with 2,000 samples in each inner-loop step to create the reference solutions [9]. The results of the constant component generated by NIPI with and without active learning are then reported in Figure 9. One notes that, for constant component, the estimation of Monte Carlo Simulation (MCS) is the same as those of DLMCS and NISS. The posterior COV of constant component generated by NIPI with 200 LHS samples is 0.012, thus for the adaptive NIPI algorithm, the stopping criteria is set to that $\sqrt{\mathbb{V}_{\mathcal{D}}\left(\hat{\mathcal{M}}_0\right)}\Big/\mathbb{E}_{\mathcal{D}}\left(\hat{\mathcal{M}}_0\right) < 0.012$ happens three times in succession. As can be seen from Figure 9, both the adaptive NIPI and that without adaptive design produces robust and accurate estimates for the constant HDMR component. The posterior variances of both procedures are quite close, but by introducing the adaptive design, the total number of required $g$-function calls is reduced from 200 to 72.

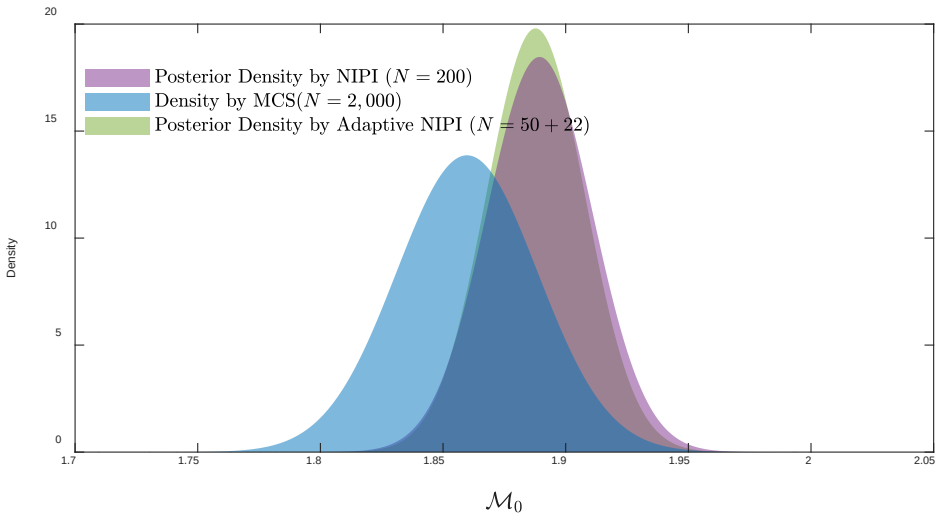


Figure 9 Results for constant HDMR component of the seepage model

The results for the first-order component functions by NISS ($N = 2 \times 10^3$), NIPI ($N = 200$), Adaptive NIPI ($N = 72$) and double-MCS ($N = 8 \times 20 \times 2 \times 10^3 = 3.2 \times 10^5$) are then reported in Figure 10. One notes the double-loop MCS results are regarded as the reference solution since the estimates are unbiased and the variations are very small. As can be seen, the results generated by both NIPI and adaptive NIPI are in good agreement with the reference solutions. However, the NISS method, although consumed much more $g$-function calls, provides poor estimations. This is due to the large span of the support of the eight distribution parameters. As can be seen, among the eight distribution parameters, the last three have no obvious effect on the expectation of the seepage, and the other five distribution parameters show approximately linear increment effects on the model response expectation. Among all the eight distribution parameters, $\mu_1$ possess the largest individual effect on the epistemic uncertainty of $\mathbb{E}[q]$, thus it is most beneficial to collect the data of $k_{xx,1}$ so as to efficiently reduce the epistemic uncertainty of $\mathbb{E}[q]$.

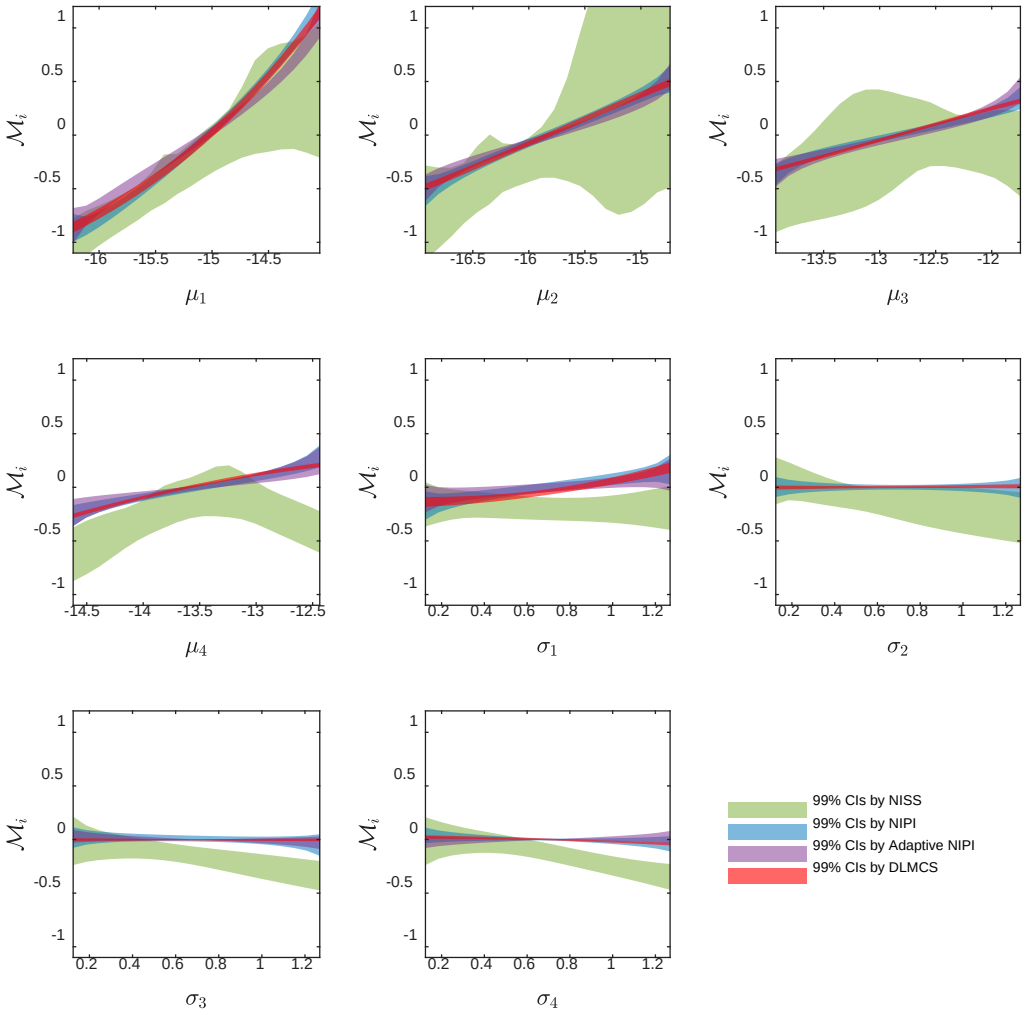


Figure 10 Results of first-order HDMR component functions of seepage model.

The posterior mean of the second-order component function generated by NIPI without adaptive design is then shown in Figure 11, and for simplicity, we only show the six most influential component functions. It is shown that, among all the second-order components, the one of $(\mu_1, \mu_2)$ is the most influential, but is less important than the first-order component of $\mu_1$.

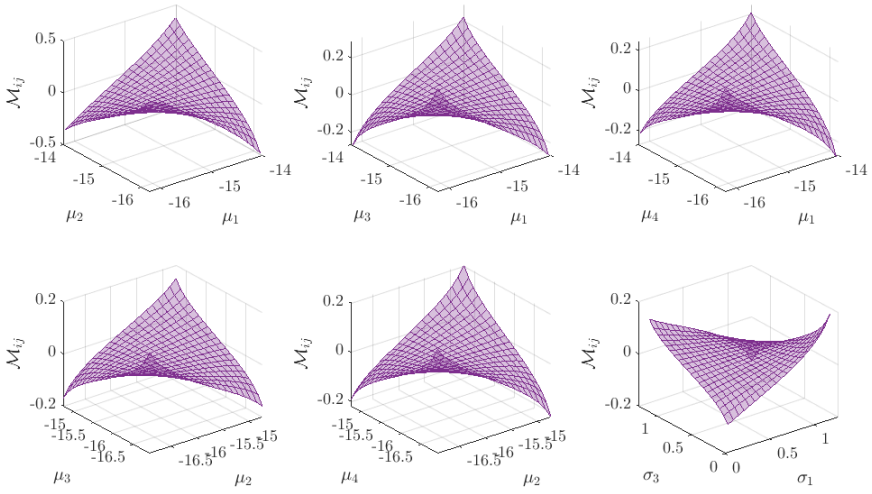

Figure 11 Posterior mean of the six most influential second-order HDMR component functions of the seepage model.

### 6.4. Multi-stage High-speed Axial Compressor System

Both of the previous two test examples have smooth model response function, and it is shown that both the adaptive NIPI and the NIPI without adaptive design works well for these examples. Here we introduce a system reliability analysis problem that involves non-smooth *g*-function. This model, originally developed in Ref. [31], concerns the reliability of a multi-stage high-speed axial compressor of the Institute for Turbomachinery and Fluid Dynamics at Leibniz University Hannover. One can refer to Refs. [31][32] for the detailed description of this model. The functional block diagram of this system is shown in Figure 12. This system is composed of eight components, each of which represents one of the rotor blade rows or stator blade rows. In this paper, these eight components are denoted as $\mathrm{R}_i\,(i=1,2,\cdots,8)$, and are classified into four categories denoted by T1~T4. It is assumed that the components of the same type have the same survival functions with equal but unknown distribution parameters. Following Refs. [31][32], we assume that all the four categories of components follow the exponential distribution, and their mean parameters are denoted as $\lambda_1$~$\lambda_4$. One notes that the failure rate of the *i*-th type of component is then $1/\lambda_i$. Due to the limited life data available, the true value of each distribution parameter cannot be learned, and their supports are all assumed to be [1, 5]. Let $t_1$~$t_8$ denote the failure time of the eight components, and $t_s$ indicate the system failure time. The *g*-function of the system failure time with respect to the failure times of components can then be formulated as:

$$t_s = g(\boldsymbol{t}) = \max_{i=1}^{8} t_{Pi} \tag{35}$$

, where $t_{Pi}$ $(i=1,2,\cdots,8)$ indicates the failure times of the eight path sets, which are formulated by:

$$\begin{aligned} t_{P1} &= \min(t_1,t_2,t_4,t_5) \quad t_{P5} = \min(t_3,t_4,t_5) \\ t_{P2} &= \min(t_1,t_2,t_4,t_6) \quad t_{P6} = \min(t_3,t_4,t_6) \\ t_{P3} &= \min(t_1,t_2,t_4,t_7) \quad t_{P7} = \min(t_3,t_4,t_7) \\ t_{P4} &= \min(t_1,t_2,t_4,t_8) \quad t_{P8} = \min(t_3,t_4,t_8) \end{aligned} \tag{36}$$

The output of interest is the Mean Time To Failure (MTTF) of the system, which is formulated as the expectation of the system failure time $t_s$, i.e., $\mathbb{E}(t_s)$. Obviously, $\mathbb{E}(t_s)$ is a function of $\lambda_1$~$\lambda_4$.

The *g*-function given in Eq. (35) is quite non-smooth since there are many minimum and maximum operators. In this case, the spatial correlation can be much weaker than that for the smooth *g*-function, which poses a challenge for NIPI.

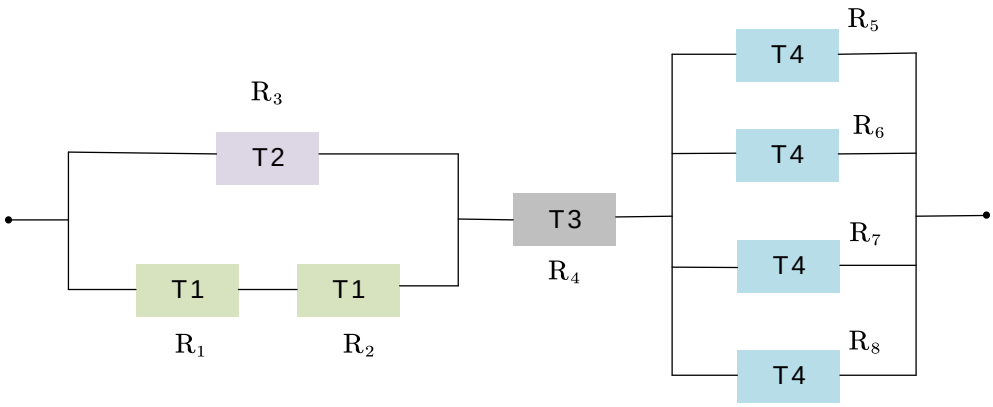


Figure 12 Block Diagram of the multistage high-speed axial compressor system.

For all the numerical implementation, we set both $\epsilon_1$ and $\epsilon_2$ as 0.4 for each $\lambda_i$. We use 500 training samples created by LHS design to implement NIPI, and the posterior density of the constant component is shown in Figure 13. The posterior COV is 0.016, thus we set $\sqrt{\mathbb{V}_{\mathcal{D}}\left(\hat{\mathcal{M}}_0\right)}\Big/\mathbb{E}_{\mathcal{D}}\left(\hat{\mathcal{M}}_0\right)<0.016$ as the stopping criteria, and use 200 LHS samples to start the adaptive experiment design. The posterior density generated by this adaptive NIPI procedure is also presented in Figure 13. The reference solution is generated by MCS with $10^4$ samples, and is provided in Figure 13. As can be seen, although the posterior COVs generated by NIPI with and without adaptive design is almost the same, the posterior mean generated by NIPI is closer to the reference mean. The adaptive NIPI requires slightly less *g*-functions calls than that generated without adaptive design, but the adaptive design procedure introduces extra computational cost for searching the maximum values of the learning function. The evolution details of both NIPI algorithms with the same 200 initial training samples are then compared in Figure 14. As can be seen, the rate of COV reduction generated with the adaptive design is almost the same as that generated without adaptive design. It is believed that, by introducing the adaptive design, no obvious benefit is obtained. The essential reason behind this phenomenon is that, due to the non-smoothness of the *g*-function in this example, the spatial correlation is much weaker than that in the last two test examples with smooth *g*-functions, causing that non-training point contributes equally to the posterior variance in most iteration steps. Thus, for non-smooth and computationally cheap *g*-functions like the one in this example, the adaptive design is not recommended. This also why the NIPI method requires more training samples in this example than that in the previous two examples.

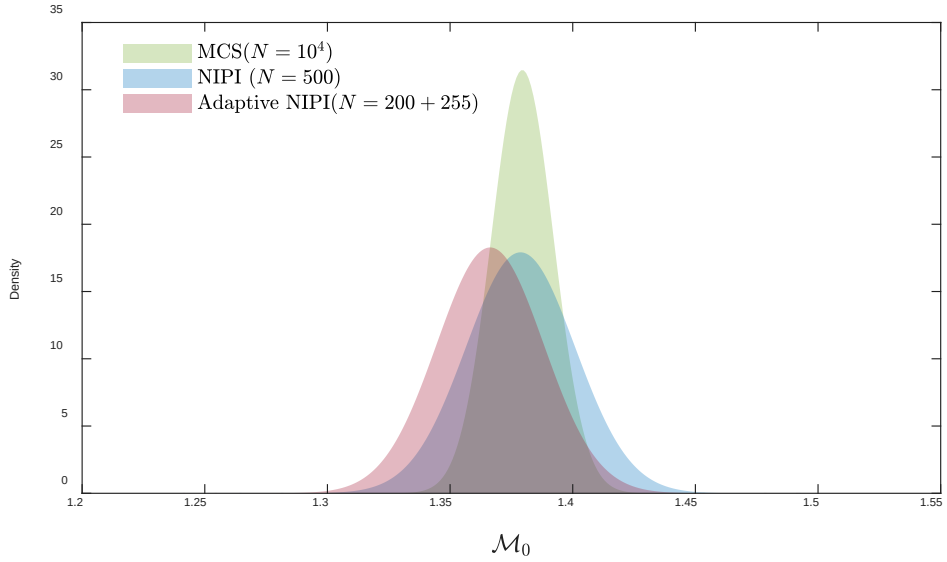


Figure 13 Results for the constant component of the compressor system.

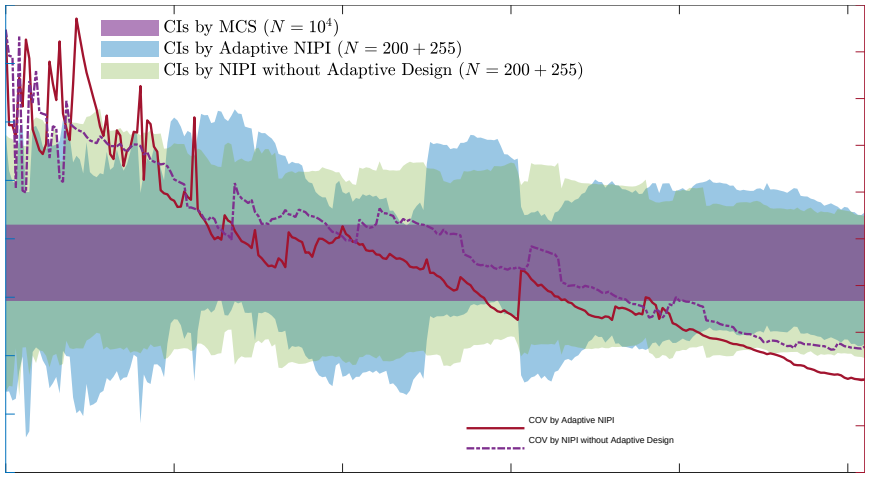


Figure 14 Evolution details of the NIPI algorithms with and without adaptive design for the compressor system.

The 99% confidence intervals of the first-order HDMR component functions computed by NISS ($N$=$10^4$), NIPI ($N$=500) and adaptive NIPI ($N$=200+212) are shown in Figure 15, together with the reference 99% confidence intervals generated by double-loop MCS. As can be seen, both NIPI and adaptive NIPI produce accurate and robust results, which are all better than those generated by NISS. However, the quality of the NISS results is also acceptable. In terms of efficiency, it can be seen that both NIPI and adaptive NIPI require much fewer $g$-function calls than NISS. It is shown that the system MTTF increases monotonically with respect to the increment of each $\lambda_i$. This is fair since the system under consideration is a coherent system, indicating that the increment of component function time will not result in the decrement of system failure time. It is seen, that $\lambda_3$ is the most important distribution parameter, and for reducing the epistemic uncertainty on system MTTF, hence, collecting the failure data for T3 type of component should be the primary choice.

This is also fair since it can be seen from Figure 12, T3 type component is the weakest link of the system. The posterior means of the six second-order HDMR component functions are then reported in Figure 16. As can be seen, among each pair of distribution parameters, $\lambda_2$ and $\lambda_3$ have the most interaction effect on the system MTTF.

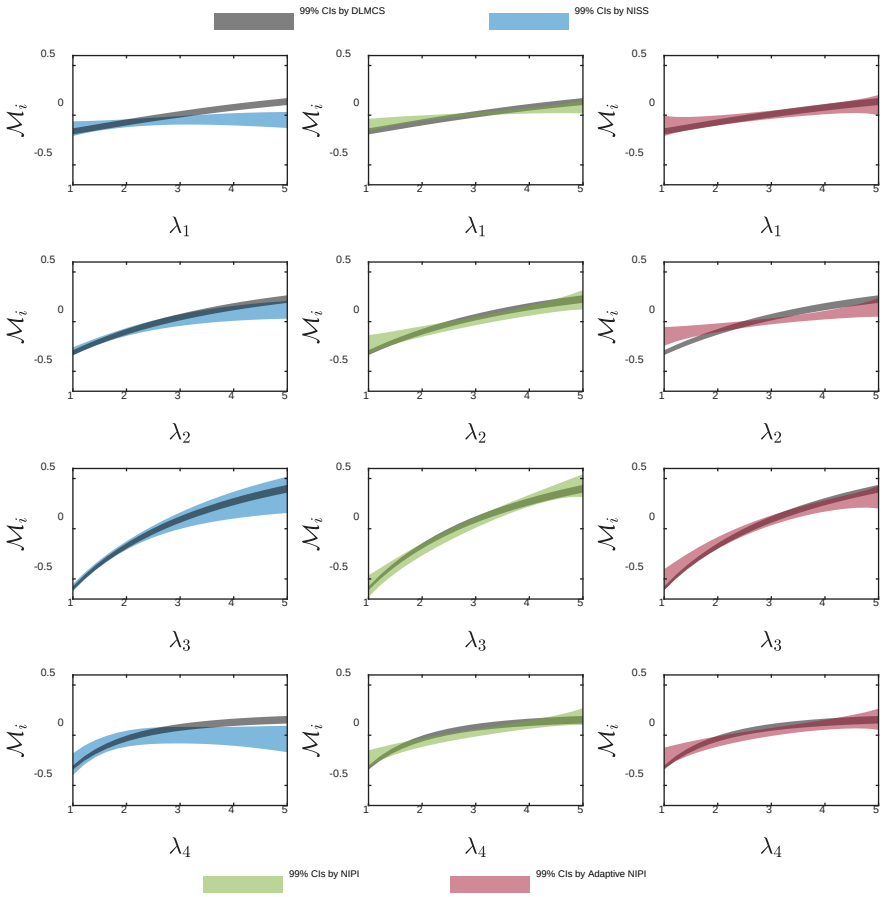


Figure 15 Results for first-order components of the compressor system, where the first column shows the results of NISS ($N$=$10^4$), the second column presents the results of NIPI ($N$=500), and the third column gives the results of adaptive NIPI ($N$=200+212). The reference results generated by double-loop MCS are also provided in each subplot.

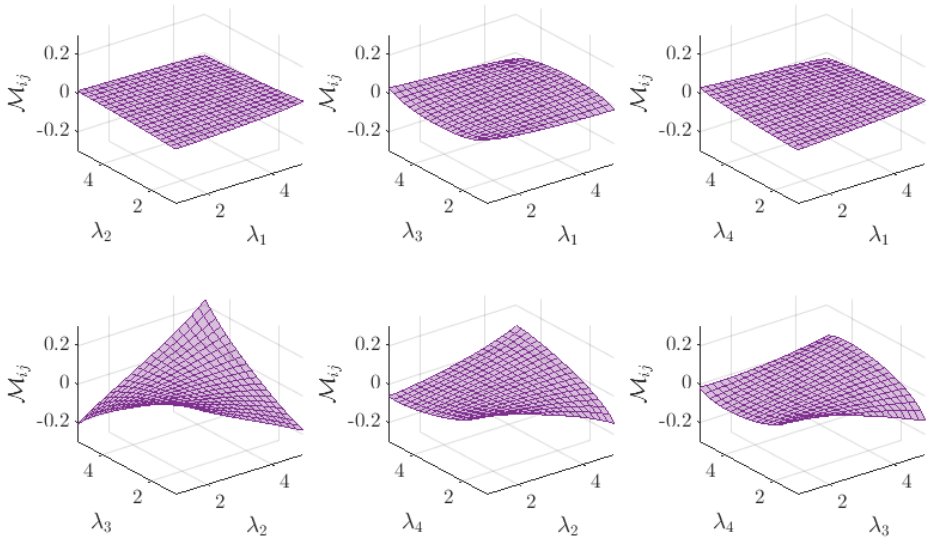


Figure 16 Posterior mean of the second-order components of the compressor system generated by NIPI.

To summarize, although the *g*-function in this example is non-smooth, and the spatial correlation in both aleatory and epistemic spaces is not as strong as that in the last two examples, the NIPI procedure still provides much better results, than NISS, in terms of both accuracy and efficiency. This indicates that, as long as the *g*-function is not totally non-smooth and it exhibits a certain extent of spatial correlation, the NIPI shows its advantages.

### 6.5. Final remarks

The results of the three test examples have demonstrated our arguments in the introduction and methodology sections, and also show new findings. The classical NISS method makes use of the spatial correlation information in the epistemic uncertainty space so as to formulate a set of estimators being Gaussian processes, and it is much more efficient than the double-loop procedures. Whereas, the NIPI method makes full use of the spatial correlation information in both aleatory space and epistemic space, and results show that NIPI is much more efficient than NISS for the test examples in this work. However, it should be noted that, in all the three test examples, the support of the epistemic space is very large, and the NISS method is not good at dealing with this type of problem. NISS performs well for small epistemic uncertainty, but may lose its advantage when the epistemic uncertainty is large, resulting from the large variation of the ratio functions in the NISS estimators (see Refs. [15][16] for more details.). Conversely, the NIPI method performs well no matter how large the epistemic uncertainty is.

By comparing the results, it is found that the NIPI method is extremely efficient for problems with smooth model response functions, and for non-smooth model response functions, the NIPI method will slightly lose its efficiency, but is still much more efficient for problems with large epistemic uncertainty. It is also shown that, by introducing the adaptive experiment design strategy to the NIPI method, the required number of *g*-function calls can be further reduced if the model response functions are smooth. However, as this improvement is obtained by making full use of the spatial correlation information integrated in the learning function, for problems with non-smooth model response functions where the spatial correlation is weak, the adaptive experiment design may lose its advantage.

All the results of the above test examples are limited to the REF for the simplicity of illustration.

However, the method is applicable for any orders of model response moments. Take the second-order origin moment as an example. For this case, the integrand in Eq. (1) should be replaced by $g^2(\boldsymbol{x})$, and all the remaining procedures keep the same. While the second-order central moment $m_2(\boldsymbol{\theta})$ is concerned, one can first formulate the corresponding integral as:

$$m_2(\boldsymbol{\theta}) = \frac{1}{2}\int [g(\boldsymbol{x}) - g(\boldsymbol{x}')]^2 f(\boldsymbol{x}'|\boldsymbol{\theta}) f(\boldsymbol{x}|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{x}' \mathrm{d}\boldsymbol{x} \tag{37}$$

, where $\boldsymbol{x}'$ is an independent replicate of $\boldsymbol{x}$. Based on Eq. (37), we need to replace the integrand to $[g(\boldsymbol{x}) - g(\boldsymbol{x}')]^2$ for estimating the second-order central moment function $m_2(\boldsymbol{\theta})$.

When using the NIPI method, one should avoid confusion of the two kinds of epistemic uncertainties involved. As explained in section 2, for all the model inputs, the epistemic uncertainty due to the lack of information is characterized by the interval models of the distribution parameters $\boldsymbol{\theta}$, which results in the epistemic uncertainty of the model response moments. The amount of epistemic uncertainty presented on the model output can be reflected by the span of the bounds of the response moments. In this paper, the exact values of the bounds are not of specific concern, but they can also be computed directly from the estimated model response moment functions. The specific concern here is to study quantitatively the functional relationship between the model response moments and the epistemic input parameters, and this information is reflected visibly by the HDMR components of the model response expectation function. Another kind of epistemic uncertainty is the numerical discretization error caused by the limited number of training data, and it is measured and properly controlled by using the posterior variance of the NIPI estimates. This source of epistemic uncertainty does not reflect the input epistemic uncertainty, and one should not be confused.

## 7. Conclusions

This paper presents two contributions for efficiently propagating the distributional p-box models based on the probabilistic integration method. First, the NIPI method is developed for analytically deriving the posterior mean and variance of the HDMR components of the probabilistic response function, which are useful for learning the behavior of the function of the probabilistic responses (e.g., response expectation and variance) with respect to non-deterministic input distribution parameters. Second, an adaptive design strategy is developed for further improving the efficiency of NIPI by identifying the optimal training points actively. The above two developments make full use of the spatial correlation information in both the aleatory space and epistemic space to improve the accuracy of the numerical integration, and the numerical errors are regarded as a source of epistemic uncertainty being analytically propagated to the integration results. Results of examples show that, for smooth model response functions, the proposed method is very efficient, and for non-smooth response function, the efficiency will decrease, but is still more efficient than NISS.

Compared with the NISS methodology framework, there are still several challenges to be fixed. First, the spatial correlation information is learned by the kernel function in the GPR model, thus for very high-dimensional problems (e.g., $n + d > 100$), the proposed method can be ineffective,

and proper dimensionality reduction techniques need to be developed. Second, the NIPI method is currently not applicable for estimating the failure probability function since the integrand is the non-smooth indicator function of the failure domain which commonly shows large unbalance in the aleatory space of model inputs. There should be some special treatments to deal with such kind of problems. Both challenges will be explored in future work.

**Acknowledgment**

This work is supported by the National Natural Science Foundation of China (NSFC 51905430). The first author is also supported by the Alexander von Humboldt Foundation of Germany and the Top International University Visiting Program for Outstanding Young Scholars of Northwestern Polytechnical University. The third author would also like to acknowledge the support from ANID (National Agency for Research and Development, Chile) under its program FONDECYT, grant number 1180271.

**Appendix A: Posterior Variances of First-order HDMR Components**

The posterior variance of the first-order component function $\mathcal{M}_i(v_i)$ induced by the GPR model $\hat{\mathcal{G}}_2(\boldsymbol{w})$ can be derived as:

$$\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_i(v_i)\right] = \mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] + \mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] - 2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right), \hat{\mathcal{M}}_0\right] \tag{A1}$$

, where the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]$ is given by Eq.(18), and the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ is further derived as:

$$\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left[\hat{\mathcal{G}}_2(\boldsymbol{w})\right]\right] = \Pi_{-i}\Pi_{-i}[\kappa(\boldsymbol{w}, (\boldsymbol{w}'_{-i}, v_i))] - \Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top}]K^{-1}\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] \tag{A2}$$

with $\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top}]$ given in Eq. (20), and $\Pi_{-i}\Pi_{-i}[\kappa(\boldsymbol{w}, (\boldsymbol{w}'_{-i}, v_i))]$ indicating the integral with respect to $\boldsymbol{w}_{-i}$ and $\boldsymbol{w}'_{-i}$, which is analytically solved as:

$$\Pi_{-i}\Pi_{-i}[\kappa(\boldsymbol{w}, (\boldsymbol{w}'_{-i}, v_i))] = \sigma_0^2 |2\Sigma_{-i}^{-1} + I|^{-1/2}. \tag{A3}$$

In Eq. (A1), the covariance term is derived as:

$$\begin{aligned}
&\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right), \hat{\mathcal{M}}_0\right] \\
&= \int \left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right) - \Pi_{-i}\left[\mathbb{E}_{\mathcal{D}}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]\right]\left[\hat{\mathcal{M}}_0 - \mathbb{E}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]\right] p\left(\hat{\mathcal{G}}_2\right) \mathrm{d}\hat{\mathcal{G}}_2 \\
&= \Pi'\Pi_{-i}\left[\int \left[\hat{\mathcal{G}}_2(\boldsymbol{w}) - \mathbb{E}_{\mathcal{D}}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]\left[\hat{\mathcal{G}}_2(\boldsymbol{w}') - \mathbb{E}_{\mathcal{D}}\left(\hat{\mathcal{G}}_2(\boldsymbol{w}')\right)\right] p\left(\hat{\mathcal{G}}_2\right) \mathrm{d}\hat{\mathcal{G}}_2\right] \\
&= \Pi'\Pi_{-i}\left[\mathrm{cov}_{\mathcal{D}}\left[\hat{\mathcal{G}}_2(\boldsymbol{w}), \hat{\mathcal{G}}_2(\boldsymbol{w}')\right]\right]
\end{aligned} \tag{A4}$$

, where $\Pi\Pi_{-i}[\cdot]$ indicates the integral with respect to $\boldsymbol{w}_{-i}$ and $\boldsymbol{w}'$, and its integrand is formulated as:

$$\mathrm{cov}_{\mathcal{D}}\left[\hat{\mathcal{G}}_2(\boldsymbol{w}), \hat{\mathcal{G}}_2(\boldsymbol{w}')\right] = \kappa(\boldsymbol{w}, \boldsymbol{w}') - \boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})^{\top}K^{-1}\boldsymbol{\kappa}(\mathcal{W}, \boldsymbol{w}')^{\top}. \tag{A5}$$

Substituting Eq. (A5) into Eq. (A4) yields:

$$\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right), \hat{\mathcal{M}}_0\right] = \Pi\Pi_{-i}[\kappa(\boldsymbol{w}, \boldsymbol{w}')] - \Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})]^{\top}K^{-1}\Pi[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] \tag{A6}$$

, where

$$\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w}, \mathcal{W})] = \sigma_0^2 |2\Sigma_{-i}^{-1} + I|^{-1/2} \exp\left(-\frac{1}{2\sigma_{v,i}^2}(v_i - \mathcal{W}_{,i})^2\right) \tag{A7}$$

, and

$$\begin{aligned}\Pi\Pi_{-i}[\kappa(\boldsymbol{w},\boldsymbol{w}')] &= \sigma_0^2|2\Sigma_{-i}^{-1}+I|^{-1/2}\Pi_i'\left[\exp\left(-\frac{1}{2\sigma_{v,i}^2}(v_i-v_i')^2\right)\right] \\ &= \sigma_0^2|2\Sigma_{-i}^{-1}+I|^{-1/2}(\sigma_{v,i}^{-2}+1)^{-1/2}\exp\left(-\frac{1}{2(\sigma_{v,i}^2+1)}v_i^2\right).\end{aligned} \tag{A8}$$

By substituting Eq.(18), Eq. (A2) and Eq. (A6) into Eq. (A1), we generate the closed-form expression for the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_i(v_i)\right]$ of the first-order HDMR components.

■

**Appendix B: Posterior Variances of Second-order HDMR Components**

The posterior variance of $\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})$ is derived as:

$$\begin{aligned}\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})\right] = {} & \mathbb{V}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]+\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]+\mathbb{V}_{\mathcal{D}}\left[\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]+\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right] \\ & -2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]-2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] \\ & +2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\hat{\mathcal{M}}_0\right]+2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] \\ & -2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\hat{\mathcal{M}}_0\right]-2\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\hat{\mathcal{M}}_0\right].\end{aligned} \tag{A9}$$

In the above equation, the closed-form expressions for $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ and $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ have been given in Eq. (A2), and that for $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_0\right]$ is given in Eq. (18). In Eq. (A9), the analytical expressions for the last two covariance terms are given by Eq. (A6). Therefore, we need only to derive the closed-form expressions for the variance term $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ and the remaining four covariance terms.

Similar to Eq. (A2), the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ can be derived as:

$$\begin{aligned}\mathbb{V}_{\mathcal{D}}\left[\Pi_{-ij}\left[\hat{\mathcal{G}}_2(\boldsymbol{w})\right]\right] = {} & \Pi_{-ij}\Pi_{-ij}[\kappa(\boldsymbol{w},(\boldsymbol{w}_{-ij}',\boldsymbol{v}_{ij}))] \\ & -\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})^{\top}]K^{-1}\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]\end{aligned} \tag{A10}$$

, where $(\boldsymbol{w}_{-ij}',\boldsymbol{v}_{ij})$ represents the ($n$+$d$)-dimensional row vector equal to $\boldsymbol{w}'$, except for the ($n$+$i$)-th and ($n$+$j$)-th components, which are equal to $v_i$ and $v_j$ respectively, $\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]$ is given by Eq. (27), and $\Pi_{-ij}\Pi_{-ij}[\kappa(\boldsymbol{w},(\boldsymbol{w}_{-ij}',\boldsymbol{v}_{ij}))]$ indicates the integral with respect to $\boldsymbol{w}_{-ij}$ and $\boldsymbol{w}_{-ij}'$, whose closed-form expression is derived as:

$$\Pi_{-ij}\Pi_{-ij}[\kappa(\boldsymbol{w},(\boldsymbol{w}_{-ij}',\boldsymbol{v}_{ij}))] = \sigma_0^2|2\Sigma_{-ij}^{-1}+I|^{-1/2}. \tag{A11}$$

Next we derive the expressions for the four covariance terms. Similar to Eq. (A6), the posterior covariance $\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ can be formulated as:

$$\begin{aligned}\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] = {} & \Pi_{-i}\Pi_{-ij}[\kappa(\boldsymbol{w},(\boldsymbol{w}_{-i}',v_i))] \\ & -\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]^{\top}K^{-1}\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]\end{aligned} \tag{A12}$$

, where the integrals $\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]$ and $\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]$ are given by Eq. (20) and Eq. (27) respectively, and the integral $\Pi_{-i}\Pi_{-ij}[\kappa(\boldsymbol{w},\boldsymbol{w}')]$ is further derived as:

$$\begin{aligned}\Pi_{-i}\Pi_{-ij}[\kappa(\boldsymbol{w},(\boldsymbol{w}'_{-i},v_i))] &= \sigma_0^2|2\Sigma_{-ij}^{-1}+I|^{-1/2}\Pi'_j\left[\exp\left(-\frac{1}{2\sigma_{v,j}^2}(v_j-v'_j)^2\right)\right] \\ &= \sigma_0^2|2\Sigma_{-ij}^{-1}+I|^{-1/2}(\sigma_{v,j}^{-2}+1)^{-1/2}\exp\left(-\frac{1}{2(\sigma_{v,j}^2+1)}v_j^2\right).\end{aligned} \tag{A13}$$

Note that the expression for $\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ can be deduced from Eq. (A12).

The posterior covariance $\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\hat{\mathcal{M}}_0\right]$ is similarly derived as:

$$\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-ij}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\hat{\mathcal{M}}_0\right] = \Pi\Pi_{-ij}[\kappa(\boldsymbol{w},\boldsymbol{w}')] - \Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]^{\top}K^{-1}\Pi[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})] \tag{A14}$$

, where $\Pi_{-ij}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]$ is given by Eq. (27), and $\Pi\Pi_{-ij}[\kappa(\boldsymbol{w},\boldsymbol{w}')]$ is expressed as:

$$\begin{aligned}\Pi\Pi_{-ij}[\kappa(\boldsymbol{w},\boldsymbol{w}')] &= \sigma_0^2|2\Sigma_{-ij}^{-1}+I|^{-1/2}\Pi'_{ij}\left[\exp\left(-\frac{1}{2}(\boldsymbol{v}_{ij}-\boldsymbol{v}'_{ij})\Sigma_{ij}^{-1}(\boldsymbol{v}_{ij}-\boldsymbol{v}'_{ij})^{\top}\right)\right] \\ &= \sigma_0^2|2\Sigma_{-ij}^{-1}+I|^{-1/2}|\Sigma_{ij}^{-1}+I|^{-1/2}\exp\left(-\frac{1}{2}\boldsymbol{v}_{ij}(\Sigma_{ij}+I)^{-1}\boldsymbol{v}_{ij}^{\top}\right).\end{aligned} \tag{A15}$$

The covariance term $\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right]$ can be similarly formulated as:

$$\begin{aligned}\mathrm{cov}_{\mathcal{D}}\left[\Pi_{-i}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right),\Pi_{-j}\left(\hat{\mathcal{G}}_2(\boldsymbol{w})\right)\right] &= \Pi_{-j}\Pi_{-i}[\kappa(\boldsymbol{w},(\boldsymbol{w}'_{-j},v_j))] \\ &\quad -\Pi_{-i}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]^{\top}K^{-1}\Pi_{-j}[\boldsymbol{\kappa}(\boldsymbol{w},\mathcal{W})]\end{aligned} \tag{A16}$$

, where

$$\Pi_{-j}\Pi_{-i}[\kappa(\boldsymbol{w},(\boldsymbol{w}'_{-j},v_j))] = \Pi\Pi_{-ij}[\kappa(\boldsymbol{w},\boldsymbol{w}')]. \tag{A17}$$

Till now, we generate the closed-form expressions for all the terms of posterior variances and posterior covariance in Eq. (A9), from which we can obtain the explicit expression for the posterior variance $\mathbb{V}_{\mathcal{D}}\left[\hat{\mathcal{M}}_{ij}(\boldsymbol{v}_{ij})\right]$ of the second-order component functions.

■

**References**


[1]. Helton J C, Johnson J D, Oberkampf W L, et al. Representation of analysis results involving aleatory and epistemic uncertainty. International Journal of General Systems, 2010, 39(6): 605-646.

[2]. Der Kiureghian A, Ditlevsen O. Aleatory or epistemic? Does it matter? Structural safety, 2009, 31(2): 105-112.

[3]. Helton J C, Johnson J D, Oberkampf W L. An exploration of alternative approaches to the representation of uncertainty in model predictions. Reliability Engineering & System Safety, 2004, 85(1-3): 39-71.

[4]. Faes M, Moens D. Recent trends in the modeling and quantification of non-probabilistic uncertainty. Archives of Computational Methods in Engineering, 2019: 1-39.

[5]. Beer M, Ferson S, Kreinovich V. Imprecise probabilities in engineering analyses. Mechanical systems and signal processing, 2013, 37(1-2): 4-29.

[6]. Ferson S, Kreinovich V, Grinzburg L, et al. Constructing probability boxes and Dempster-Shafer structures. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2015

[7]. Sankararaman S, Mahadevan S. Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. Reliability Engineering & System Safety, 2011, 96(7):

814-824.
[8]. Stein M, Beer M, Kreinovich V. Bayesian approach for inconsistent information. Information sciences, 2013, 245: 96-111.
[9]. Pedroni N, Zio E. Hybrid uncertainty and sensitivity analysis of the model of a twin-jet aircraft. Journal of Aerospace Information Systems, 2015, 12(1): 73-96.
[10]. Zhang H, Dai H, Beer M, et al. Structural reliability analysis on the basis of small samples: an interval quasi-Monte Carlo method. Mechanical Systems and Signal Processing, 2013, 37(1-2): 137-151
[11]. Alvarez D A, Uribe F, Hurtado J E. Estimation of the lower and upper bounds on the probability of failure using subset simulation and random set theory. Mechanical Systems and Signal Processing, 2018, 100: 782-801.
[12]. de Angelis M, Patelli E, Beer M. Advanced line sampling for efficient robust reliability analysis. Structural safety, 2015, 52: 170-182.
[13]. Wei P, Lu Z, Song J. Extended Monte Carlo simulation for parametric global sensitivity analysis and optimization. AIAA journal, 2014, 52(4): 867-878.
[14]. Zhang J, Shields M D. On the quantification and efficient propagation of imprecise probabilities resulting from small datasets. Mechanical Systems and Signal Processing, 2018, 98: 465-483.
[15]. Wei P, Song J, Bi S, et al. Non-intrusive stochastic analysis with parameterized imprecise probability models: I. Performance estimation. Mechanical Systems and Signal Processing, 2019, 124: 349-368
[16]. Wei P, Song J, Bi S, et al. Non-intrusive stochastic analysis with parameterized imprecise probability models: II. Reliability and rare events analysis. Mechanical Systems and Signal Processing, 2019, 126: 227-247
[17]. Song J, Valdebenito M, Wei P, et al. Non-intrusive imprecise stochastic simulation by line sampling. Structural Safety, 2020, 84: 101936
[18]. Song J, Wei P, Valdebenito M, et al. Generalization of non-intrusive imprecise stochastic simulation for mixed uncertain variables. Mechanical Systems and Signal Processing, 2019, 134: 106316
[19]. Schöbi R, Sudret B. Structural reliability analysis for p-boxes using multi-level meta-models. Probabilistic Engineering Mechanics, 2017, 48: 27-38
[20]. Sofi A, Muscolino G, Giunta F. Propagation of uncertain structural properties described by imprecise Probability Density Functions via response surface method. Probabilistic Engineering Mechanics, 2020: 103020
[21]. Briol F X, Oates C J, Girolami M, et al. Probabilistic integration: A role in statistical computation?. Statistical Science, 2019, 34(1): 1-22.
[22]. Cockayne J, Oates C J, Sullivan T J, et al. Bayesian probabilistic numerical methods. SIAM Review, 2019, 61(4): 756-789.
[23]. Wei P, Zhang X, Beer M. Adaptive experiment design for probabilistic integration. Submitted to CMAME,
[24]. Graf W, Götz M, Kaliske M. Analysis of dynamical processes under consideration of polymorphic uncertainty. Structural Safety, 2015, 52: 194-201.
[25]. Rasmussen C E, Williams C K I. Gaussian process for machine learning. Cambridge:

The MIT Press, 2005.

[26]. Rosenblatt M. Remarks on a multivariate transformation. The annals of mathematical statistics, 1952, 23(3): 470-472

[27]. Li G, Wang S W, Rabitz H. Practical approaches to construct RS-HDMR component functions. The Journal of Physical Chemistry A, 2002, 106(37): 8721-8733

[28]. Park J S. Optimal Latin-hypercube designs for computer experiments. Journal of statistical planning and inference, 1994, 39(1): 95-111

[29]. Kocis L, Whiten W J. Computational investigations of low-discrepancy sequences. ACM Transactions on Mathematical Software, 1997, 23(2): 266-294

[30]. Valdebenito M A, Jensen H A, Hernandez H B, et al. Sensitivity estimation of failure probability applying line sampling. Reliability Engineering & System Safety, 2018, 171: 99-111

[31]. Miro S, Willeke T, Broggi M, et al. Reliability Analysis of an Axial Compressor Based on One-Dimensional Flow Modeling and Survival Signature. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, 2019, 5(3):031003

[32]. Salomon J, Broggi M, Kruse S, et al. Resilience Decision-Making For Complex Systems. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, 2020,6(2):020901