Probabilistic Calibration of Model Parameters with Approximate Bayesian Quadrature and Active Machine Learning

Pengfei Wei[a,b,*], Masaru Kitahara[c], Matthias G.R. Faes[d,f], Michael Beer[e,f,g]

[a]*School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, PR China*
[b]*Advanced Power Research Institute of Northwestern Polytechnical University, Chengdu, Sichuan, PR China*
[c]*Department of Civil Engineering, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan*
[d]*Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler Strasse 5, 44227, Dortmund, Germany*
[e]*Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, Hannover 30167, Germany*
[f]*Department of Civil and Environmental Engineering, University of Liverpool, Liverpool L69 3BX, United Kingdom*
[g]*International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai 200092, PR China*

Abstract

Calibration of computational models with experimental or operational data to achieve desired prediction accuracy has been widely recognized as a crucial problem in reliability engineering. Bayesian model updating (BMU) has been developed as an appealing methodology framework to achieve this goal, but the existing methods range from very approximate but cheap (e.g., Laplace approximation and conjugate priors), less approximate but still a bit cheaper (e.g., approximate Bayesian computation) to quite expensive but very informative full Bayesian computation. The target of this work is to achieve accuracy of full Bayesian at cheap cost. The approximate Bayesian quadrature has emerged to be a highly appealing scheme to achieve this goal. In this work, we develop a family of new acquisition functions with closed-form expressions to accelerate the approximate Bayesian quadrature for addressing the BMU problem with desired accuracy. The proposed method leverages the information revealed by both mean predictions and their posterior covariance of the probabilistic regression model trained for approximating the likelihood function, and thus provides better trade-off between exploration and exploitation. Results of both numerical and engineering examples show that the proposed method is applicable to multi-model problems with high accuracy and efficiency.

Keywords: Bayesian model updating; Approximate Bayesian quadrature; Gaussian process regression; Acquisition function; Active machine learning

1. Introduction

Reliability assessment and design optimization based on physical models of failure have been a common treatment in reliability engineering, due to the increasing complexity of products and high cost or even unavailability of implementing physical reliability tests. To this end, the credibility of the physical models has been of special concern for theoretical development of reliability methods driven by physics of

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

failure. However, due to the complexity of service environments and failure mechanisms, it is difficult or even impossible to capture all the physical details with mathematical models, and what's more, numerical solution of the mathematical models may introduce extra prediction uncertainty resulted from numerical computation. Due to the above reasons, the established computational simulators, although requiring significant computational effort, may provide untrustworthy predictions. This has motivated the development of methods for calibrating the computational simulators, with the proper fusion of measurements generated from well-designed physical tests or the real-live operation of the structure under consideration.

To this day, several model calibration methodology frameworks have been developed, and can generally be divided into two groups, i.e., deterministic and non-deterministic methods. The deterministic methods aim at calibrating the model parameters by minimizing a cost function, and are realized with effective sensitivity information computed by e.g., adjoint methods [1]. The deterministic methods seek to find a point estimate of the parameters which best interpreting the measurements, and is commonly criticized for the inability of treating the multi-modal inference problems and instability for processing the problems with measurement noises. On the contrary, the non-deterministic framework aims at modeling the alternative types of root uncertainties resulting in the prediction uncertainty of a simulator with probability models or non-probabilistic models [2], and has received tremendous attentions in the past two decades since the milestone work in Ref. [3]. This work focuses on the probabilistic framework. Under this framework, the prediction uncertainty of a simulator is attributed to two sources, i.e., model bias and model parameters, and the Bayesian scheme is commonly used for statistical inference [4]. Under this framework, the problem can be further divided into to three subgroups [5], i.e., bias correction only, parameters calibration only, as well as bias correction and parameters calibration at once. In this work, we only consider the model parameter calibration problem. For this case, if some model parameters are intrinsically random, the resulted probabilistic calibration problem is termed as 'stochastic model updating', and the numerical solution to this problem can be more challenging [6, 7] as it involves several loops for numerical computation. For this study, we follow the formulation of most probabilistic model updating literature [8, 9], and treat the model parameters as deterministic-but-unknown variables with its epistemic uncertainty described by a subjective probability model.

The methods for probabilistic parameter calibration, also named as Bayesian Model Updating (BMU), has also been extensively studied in the past two decades, and the available methods can be categorized into four groups. The first group is 'Markov Chain Monte Carlo (MCMC) sampling' [10], which uses the discrete Markov chain to sequentially approach the posterior density of the model parameters. Following the classical Metropolis-Hastings (MH) sampler [10], many improved MCMC algorithms, such as the transitional MCMC [11, 12], the Hamiltonian Monte Carlo [13], the random-then-optimize scheme [14], the covariance-based MCMC [9], et al., have been developed for improving the performance for specific types of challenging BMU problems. Generally, the MCMC sampling schemes are computationally demanding for expensive-

to-evaluate simulator as each state requires at least one call of simulator, and can be less effective for multi-modal problems due to the low probability of transferring from one region to another disconnected and distant region.

The second group consists of methods based on Bayesian filter algorithms, which are originally developed for probabilistic prediction and updating of the state of a physical system, and relies on the state-space models. This group of methods includes Kalman filter (including its advanced versions), particle filter, quadrature filter, functional Bayesian filter, et al., one can refer to Ref. [15] for a review of selected Bayesian filter algorithms. These algorithms have also been preliminary extended for BMU problem. For example, the unscented Kalman filter [16] and the particle filter [17] have both been extended for BMU with reasonable modifications and improvements. Overall, the Bayesian filter algorithms is flexible as it can sequentially fuse the measurements batch by batch, and but the high computational cost and less suitability to multi-modal problems needs to be alleviated.

The third group of methods are usually named as Bayesian Updating with Structural Reliability (BUS) methods, which is originally from Ref. [18]. The basic idea is to reformulate the BMU problem as a structural reliability analysis problem, and thus to enable solving the BMU problems with the extensive well-developed structural reliability methods such as sampling methods (e.g., subset simulation and line sampling), moment-based methods (first-order and second order reliability method), etc. It should be noted that the equivalent reliability problems usually have highly nonlinear or even ill-behaved feature, and the associated probability of failure can be extremely small given many observations. What's more, one factor, denoted as α, needs to be determined as a prior of reliability analysis, which is non-trivial [19]. To alleviate alternative challenges, many active learning schemes based on neural network or Gaussian process regression have been combined with the BUS scheme, see e.g., Refs.[20, 19, 21]. Despite the above improvements, the performance for tackling multi-modal problems still needs to be improved.

The last group, which has received attention in the community of machine learning, is called Bayesian quadrature, which is originally developed for estimating multi-dimensional integral equations [22]. This scheme has been extended for estimating the 'evidence' term, defining by the integral of likelihood over prior density, and then also the posterior density with the resultant surrogate. Two key issues need to be addressed. The first concern the approximation of the likelihood with a probabilistic regression model to capture the prediction uncertainty [23], while the the second aims at devising acquisition functions for acquiring faster convergence rate [24]. For the later issue, most of the current developments are based on modification of the uncertainty sampling acquisition function, which does not make full use of the information of the probabilistic regression model for approximating the likelihood. However, these methods are shown to be of superiority for addressing the multi-modal problems in terms of both accuracy and efficiency, owing to the flexibility and interpretability of the Gaussian process regression model. It should be noted that there are also other developments which cannot be attributed to the above four groups, such as the transport map

theory [25], but we don't give more details as it is not the focus of this work.

Following the technical road-map of Bayesian quadrature, we develop a new family of acquisition functions in this work, with the aim of presenting more flexibility for balancing between exploration and exploitation. This is realized by approximating a proxy function, defined by the square root of the likelihood, with a Gaussian process model, and then formulating the acquisition function by leveraging both posterior mean and covariance information of the Gaussian process model. The closed-form expression of the new acquisition function is also presented, allowing for efficient and effective query for the active learning process. Results of the benchmark studies show that the proposed method is efficient in terms of both the number of model calls and the computer time, and is applicable for inferring multi-modal posterior density which is challenging but common in Bayesian model updating.

The remaining part is organized as follows. Section 2 briefly reviews some research materials, including the formulation of the BMU problem and the exact Bayesian quadrature for active learning of multi-dimensional integral. Second 3 presents the main development of this work, including the approximate Bayesian quadrature scheme, the new acquisition function, the closed-form expressions, and the details of algorithm, followed by benchmark studies in section 4. Section 5 concludes this work.

2. Theoretical foundations

2.1. Bayesian model updating

Let denote by $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\theta})$ the response function of the computational model, where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^\top \in \mathbb{X} \subseteq \mathbb{R}^n$ denotes the n-dimensional controllable input variables which may include, e.g., spatial variables, time variable and excitation; and $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_d)^\top \in \mathbb{T} \subseteq \mathbb{R}^d$ indicates the d-dimensional deterministic-but-unknown model parameters to be calibrated, which may include initial/boundary conditions, physical parameters, etc. Further, let denote by $y_{\text{obs}}^{(k)}$ a set of experimental measurements observed at a sequence of locations $\boldsymbol{x}_{\text{obs}}^{(k)}$, respectively, with $k = 1, 2, \cdots, N_{\text{obs}}$. Denote the experimental observation data as a set $\mathcal{D}_{\text{obs}} = \left\{(\boldsymbol{x}_{\text{obs}}^{(k)}, y_{\text{obs}}^{(k)})\right\}_{k=1}^{N_{\text{obs}}}$. Neglecting the model bias of $\mathcal{M}$, the relationship between each observation $y_{\text{obs}}^{(k)}$ and the model response can be formulated as:

$$y_{\text{obs}}^{(k)} = \mathcal{M}(\boldsymbol{x}_{\text{obs}}^{(k)}, \boldsymbol{\theta}) + \epsilon(\boldsymbol{x}_{\text{obs}}^{(k)}) \tag{1}$$

, where $\epsilon\left(\boldsymbol{x}_{\text{obs}}^{(k)}\right)$ is the measurement noise which can be described by a stochastic process defined on $\mathbb{X}$. In most publications, this stochastic process is assumed to be a Gaussian white noise, i.e., $\epsilon(\boldsymbol{x}) = \epsilon \sim \mathcal{N}\left(0, \sigma_{\text{n}}^2\right)$, with σ_{n}^2 being the variance of noise. The target of model parameter calibration is then to infer the true value or a probability distribution of $\boldsymbol{\theta}$ given the observations $\mathcal{D}_{\text{obs}}$, and perhaps, also a pre-specified prior probability distribution of $\boldsymbol{\theta}$. One notes that, when the model bias, denoted as $\delta(\boldsymbol{x})$, is also considered,

it requires modeling both the deterministic-but-unknown parameters and model bias with probabilistic models, and requires methods for joint updating of these two types of probabilistic models. This can be more challenging, and will not be treated in this work.

This problem can be formulated as either under a deterministic framework solved with optimization algorithms, or a probabilistic framework known as BMU, or specifically Bayesian updating of model parameters, as the model bias is neglected. It is known that the deterministic methods commonly lead to misleading deterministic results as the problems are commonly indefinite and an infinite alternative possible values of $\boldsymbol{\theta}$ may exist to fit the observations. The probabilistic framework formulated with Bayesian rule then turns out to be a compatible way for treating such kind of paradox. Given the assumption of Gaussian white noise of ϵ, the likelihood function of the observations $\mathcal{D}_{\text{obs}}$ can be formulated as:

$$p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right) = \prod_{k=1}^{N_{\text{obs}}} \phi(y_{\text{obs}}^{(k)} - \mathcal{M}(\boldsymbol{x}_{\text{obs}}^{(k)}, \boldsymbol{\theta})|\sigma_{\text{n}}^2) \tag{2}$$

, where $\phi\left(\cdot|\sigma_{\text{n}}^2\right)$ refers to the density function of the Gaussian distribution with zero mean and variance σ_{n}^2. Then, given a prior density $\boldsymbol{\theta}$ as $p\left(\boldsymbol{\theta}\right)$, the posterior density of $\boldsymbol{\theta}$ is expressed as:

$$p\left(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}}\right) = Z^{-1} p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \tag{3}$$

, following the classical Bayesian rule, where the normalizing constant Z is defined by the following d-dimensional integral:

$$Z = \int_{\mathbb{T}} p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \tag{4}$$

, and is termed as 'model evidence', or simply as 'evidence', as its value measures the strength of evidence of model after being calibrated. The numerical analysis task for BMU can the be simply described as estimating the posterior density $p\left(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}}\right)$, which is generally impossible to be solved analytically unless the prior and posterior are a pair of conjugate distribution. Moreover, due to the necessity of simulating multi-physical and/or multi-scale processes, the computational cost for each call of the simulator $\mathcal{M}\left(\boldsymbol{x}, \boldsymbol{\theta}\right)$, and then also the likelihood function $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)$, can be extremely expensive. Therefor, estimating $p\left(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}}\right)$, with desired accuracy and as few simulator calls as possible, has long been an endearing topic. Multiple branches of algorithms have been developed for this numerical task, which includes, sampling methods [8], transport map theory [25], Bayesian quadrature [26], structural reliability method [18], etc., as reviewed in the introduction section, where the Bayesian quadrature will be used in this work for further development due to its simplicity and potential for addressing concerned challenges. An approximate version of Bayesian quadrature will be developed, but in what follows we briefly review the exact version and also its active learning scheme.

2.2. Exact Bayesian Quadrature and Active Learning

We take the estimation of a general d-dimensional integral:

$$Z = \Pi_p [g] = \int_{\mathbb{T}} g(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \tag{5}$$

, as an example, to review the basic rationale of Bayesian quadrature, where $g(\boldsymbol{\theta})$ is an expensive-to-evaluate integrand, and $\Pi_p[\cdot] = \int_{\mathbb{T}} \cdot p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$ denotes the integral operator over $p(\boldsymbol{\theta})$. Following the classical polynomial quadrature, the Bayesian quadrature rule also consists of a set of nodes $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}^{(l)}\right]_{l=1}^{M}$ (a $M \times d$-dimensional matrix) and weights $\mathcal{W} = \left[w^{(l)}\right]_{l=1}^{M}$ (a M-dimensional column vector), and is presented as $Z = \sum_{l=1}^{M} y^{(l)} w^{(l)}$, with $y^{(l)} = g\left(\boldsymbol{\theta}^{(l)}\right)$. However, before the integrand is computed, we may have no prior information on the behavior of it, or we may know some information, such as the smoothness, of the integrand. The Bayesian quadrature is then initialized by assuming a prior Gaussian Process (GP) model for g, then inferring a posterior GP model for approximating the integrand conditioned on the quadrature nodes $\mathcal{S} = \left\{\left(\boldsymbol{\theta}^{(j)}, y^{(j)}\right)\right\}_{j=1}^{M}$, and ultimately deducing a posterior probability distribution for the integral Z. This posterior probability distribution provides a mean prediction for Z, and also a posterior variance or confidence interval for summarizing the prediction uncertainty, resulted from the prediction uncertainty of the GP model. Thus, we first briefly review the content of GP model.

The prior GP model of g is assumed to be $\hat{g}(\boldsymbol{\theta}) \sim \mathcal{GP}\left(m(\boldsymbol{\theta}), \kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right)$, where $m(\boldsymbol{\theta})$ is the prior mean function which can be assumed to be zero, constant or polynomial; and $\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ is the prior covariance function which is assumed to be a positive-definite kernel function, and reflects the prior information on the smoothness of the integrand. For example, a functional Hilbert space equipped with the commonly used squared exponential kernel is composed of nearly all possible smooth functions, while the Matérn kernel defines a Hilbert space consisting of functions with its derivatives up to a given order, which is also called Sobolev space [27], and is suitable for extrapolating less smooth functions. A Hilbert space equipped with a positive definite kernel $\kappa(\cdot, \cdot)$ is called a Reproducing Kernel Hilbert Space (RKHS), and is denoted as $\mathbb{H}$. In this work, we use the squared exponential kernel defined as:

$$\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right) = \sigma_0^2 \exp\left(-\sum_{i=1}^{d} \frac{\left(\theta_i - \theta_i'\right)^2}{2\sigma_i^2}\right) \tag{6}$$

, where σ_0^2 and $\left\{\sigma_i^2\right\}_{i=1}^{d}$ are the variance and scale correlation hyperparameters respectively. It should be noted that, due to the infinite number of available derivatives, utilization of squared exponential kernel tend to underestimate the prediction variance. Based on the above prior assumption, it is known that the column vector $\mathcal{Y} = \left[y^{(l)}\right]_{l=1}^{M}$ follows a M-dimensional Gaussian distribution, by maximizing the joint density (the likelihood) of which, the hyperparameters for defining m and κ can be estimated. Then, by conditioning on

the training nodes $\mathcal{S}=\left\{\left(\boldsymbol{\theta}^{(j)},y^{(j)}\right)\right\}_{j=1}^{M}$, the posterior GP model $\hat{g}_M\left(\boldsymbol{\theta}\right)\sim\mathcal{GP}\left(\mu_{g,M}\left(\boldsymbol{\theta}\right),c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right)$ can be inferred with the posterior mean $\mu_{g,M}$ and covariance $c_{g,M}$ being formulated as:

$$\mu_{g,M}\left(\boldsymbol{\theta}\right)=m\left(\boldsymbol{\theta}\right)+\kappa\left(\boldsymbol{\theta},\boldsymbol{\Theta}\right)\mathcal{K}^{-1}\left(\mathcal{Y}-m\left(\boldsymbol{\Theta}\right)\right) \tag{7a}$$

$$c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)=\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)-\kappa\left(\boldsymbol{\theta},\boldsymbol{\Theta}\right)\mathcal{K}^{-1}\kappa\left(\boldsymbol{\Theta},\boldsymbol{\theta}'\right) \tag{7b}$$

, where $\kappa\left(\boldsymbol{\theta},\boldsymbol{\Theta}\right)$ is a M-dimensional row vector with its l-th element being $\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(l)}\right)$, and $\mathcal{K}$ is the $M\times M$-dimensional Gram matrix with its (k,l)-th element being $\mathcal{K}_{kl}=\kappa\left(\boldsymbol{\theta}^{(k)},\boldsymbol{\theta}^{(l)}\right)$. Intuitively, the posterior mean $\mu_{g,M}$ presents a mean prediction, and the posterior variance $\sigma^2_{g,M}\left(\boldsymbol{\theta}\right)=c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}\right)$ summarizes the prediction uncertainty.

Replacing the integrand g in Eq. (5) with the GP $\hat{g}_M$ yields [28, 22]:

$$\hat{Z}_M=\Pi_p\left[\hat{g}_M\right]=\int_{\mathbb{T}}\hat{g}_M\left(\boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}. \tag{8}$$

It is then trivial to conclude that $\hat{Z}$ is a Gaussian random variable, denoted as $\hat{Z}_M\sim\mathcal{N}\left(\mu_{Z,M},\sigma^2_{Z,M}\right)$, resulted from that fact that the linear projection of GP on a deterministic function is Gaussian. The posterior mean $\mu_{Z,M}$ and variance $\sigma^2_{Z,M}$ are then formulated as [29]:

$$\mu_{Z,M}=\Pi_p\left[\mu_{g,M}\left(\boldsymbol{\theta}\right)\right]=\Pi_p\left[m\left(\boldsymbol{\theta}\right)\right]+\Pi_p\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\Theta}\right)\right]\mathcal{K}^{-1}\left(\mathcal{Y}-m\left(\boldsymbol{\Theta}\right)\right) \tag{9a}$$

$$\sigma^2_{Z,M}=\Pi_p\Pi'_p\left[c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right]=\Pi_p\Pi'_p\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right]-\Pi_p\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\Theta}\right)\right]\mathcal{K}^{-1}\Pi'_p\left[\kappa\left(\boldsymbol{\Theta},\boldsymbol{\theta}'\right)\right] \tag{9b}$$

, where $\Pi'_p\left[\cdot\right]=\int_{\mathbb{T}}\cdot p\left(\boldsymbol{\theta}'\right)\mathrm{d}\boldsymbol{\theta}'$ and $\Pi_p\Pi'_p\left[\cdot\right]=\int_{\mathbb{T}}\int_{\mathbb{T}}\cdot p\left(\boldsymbol{\theta}'\right)p\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta}'\mathrm{d}\boldsymbol{\theta}$. Similarly, the posterior mean $\mu_{Z,M}$ provides a mean prediction to the integral Z, while the posterior variance $\sigma^2_{Z,M}$ summarizes the prediction uncertainty. It has been shown repeatedly that, while the integrand is within the RKHS equipped with the kernel κ, the posterior variance equals exactly the square of the worst-case error [30], i.e., $\sigma^2_{Z,M}=\left(\sup_{\|g\|_{\mathbb{H}}\leqslant 1}\left|\Pi_p\left[g\right]-\mu_{Z,M}\right|\right)^2$, where $\|\cdot\|_{\mathbb{H}}$ refers to the norm operator defined in the RKHS $\mathbb{H}$. Thus the posterior variance provides a reasonable quantification of the prediction uncertainty of $\mu_{Z,M}$. Further, the guarantee of convergence of this quadrature rule for any integrand in $\mathbb{H}$ has also been repeatedly proved for both random and adaptive design of quadrature points [31, 32]. Due to the richness of a RKHS, these theoretical results present sufficient confidence for extracting the potential of using Bayesian quadrature for solving the BMU problems and beyond, by simply addressing the numerical error with the posterior variance $\sigma^2_{Z,M}$.

Further, by assumption of a zero prior mean $m\left(\boldsymbol{\theta}\right)\equiv 0$, the posterior mean quadrature rule of Eq. (9a) can be written as a classical quadrature form as $\mu_{Z,M}=\sum_{l=1}^{M}w^{(l)}y^{(l)}$ with the weight vector being

computed by $\mathcal{W}^\top = \Pi_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right)\right]\mathcal{K}^{-1}$. By further assumption of the kernel form, nearly all the classical quadrature rules, such as the Monte Carlo quadrature rule [24] and the many Gauss quadrature rules [33], can be recovered from this posterior mean rule, and thus Bayesian quadrature can be seen as a general extension of these classical rules. One more appealing feature beyond this is that, benefiting from the probabilistic description of the quadrature rule, the quadrature nodes can be obtained following an adaptive design scheme, instead of the traditional random or deterministic designs without considering the behavior of the integrand. This adaptive scheme provides great potential for accelerating the convergence.

An adaptive design scheme, or called active learning, is usually driven by an acquisition function which measures the reward of a design. Two popular acquisition functions are the Uncertainty Sampling (US) function [24] and the Posterior Variance Contribution (PVC) function [29] which are formulated as:

$$\mathcal{A}_{\mathrm{US}}\left(\boldsymbol{\theta}\right) = \sigma^2_{g,M}\left(\boldsymbol{\theta}\right) p^2\left(\boldsymbol{\theta}\right) \tag{10}$$

, and

$$\mathcal{A}_{\mathrm{PVC}}\left(\boldsymbol{\theta}\right) = \Pi'_p\left[c_{g,M}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right] p\left(\boldsymbol{\theta}\right) = \left(\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right] - \kappa\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right)\mathcal{K}^{-1}\Pi'_p\left[\kappa\left(\boldsymbol{\Theta}, \boldsymbol{\theta}'\right)\right]\right) p\left(\boldsymbol{\theta}\right) \tag{11}$$

, respectively. The US function is indeed the prediction uncertainty (weighted by the squared density) of the GP model. As the prediction uncertainty of $\hat{Z}_M$ is uniquely governed by the prediction uncertainty of $\hat{g}_M$, it is expected that by adding the point with the highest US function value as quadrature node, the prediction uncertainty of the GP model, and thus of the quadrature, can be reduced to a large extent. The PVC function presents a more appealing mathematical interpretation and also performance. It is noted that the PVC function, on the one hand, is the integral of the posterior covariance of the GP model at the location $\boldsymbol{\theta}$ over the whole support $\mathbb{T}$, and on the other hand, measures the contribution of the prediction uncertainty of the GP model at $\boldsymbol{\theta}$ to the prediction uncertainty of the quadrature, by noting that $\int_{\mathbb{T}} \mathcal{A}_{\mathrm{PVC}}\left(\boldsymbol{\theta}\right)\mathrm{d}\boldsymbol{\theta} = \sigma^2_{Z,M}$. Thus, it is expected that, by adding the peak point of the PVC function as a new quadrature point, the prediction uncertainty of the quadrature is reduced by highest amount. As PVC leverages the posterior covariance information of the GP model [29], it is shown to be more effective than US in most cases.

Due to the multi-modal behavior of the US and PVC functions, heuristic algorithms such as particle swarm optimization and genetic algorithm are commonly suggested for searching the their global peak points, which brings the requirement that the acquisition function can be computed within trivial cost, and better, admits a closed-form expressions. For US function, the closed form expression is trivial, but for the PVC function, closed-form expressions are required for $\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right]$ and $\Pi'_p\left[\kappa\left(\boldsymbol{\Theta}, \boldsymbol{\theta}'\right)\right]$. This is also necessary for closed-form expressions of the posterior mean and variance of the quadrature. Fortunately, this is available given squared exponential kernel and Gaussian form of $p\left(\boldsymbol{\theta}\right)$. Assuming that $p\left(\boldsymbol{\theta}\right)$ is of standard Gaussian form with zero mean and unit variance, the closed-form expressions for the kernel means $\Pi_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\Theta}\right)\right]$,

$\Pi'_p\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right]$ and $\Pi_p\Pi'_p\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right]$ are reported in Appendix A (see also Refs. [28, 22, 29] for theoretical deduction). The PVC function has recently been applied for addressing the BMU problem by combining it with a transitional scheme [34], a sampling scheme [35], and the classical variational Bayeian inference [36]. The aim of this work is to introduce an alternative, yet simpler, approximate Bayesian quadrature strategy for BMU, and then develop a generalization of the PVC function, this way to enable higher flexibility for balancing the exploration and exploitation of the algorithm.

3. The Proposed Method

It has been widely recognized that, see e.g., Refs. [23, 31, 26], the exact Bayesian quadrature rule reviewed in the last section may caused large numerical bias if it is directly used for estimating the evidence in Eq. (4). This is due to that fact that the integrand in Eq. (4), which is exactly the likelihood function $p\left(\mathcal{D}|\boldsymbol{\theta}\right)$, is non-negative and its range of values may cover several orders of magnitude, resulting in large bias if the GP model is directly used for approximating the integrand. Some approximate tricks have been developed for accommodating this conflict, resulting in several approximate Bayesian quadrature schemes. One notes that, by saying 'approximate Bayesian quadrature', we mean that the quadrature rules are not exact, and requires an approximate description of posterior distribution of the answer. One should not confuse it with the concept 'Approximate Bayesian Computation (ABC)' in Bayesian updating. These tricks include, e.g., approximating the logarithm of the likelihood with GP [23] and approximating the square root of the likelihood with GP [26]. A function being approximated by a GP model in this context is called a 'proxy function'. With these treatments, linear approximation and moment-matching schemes have been developed for approximating the posterior features of the likelihood, and further those of the quadrature. Our development is motivated by the scheme of approximating the square root of the likelihood function presented in Ref. [26], and thus in what follows we present introduction to this trick and also some further mathematical developments for extending this trick.

3.1. Approximate Bayesian Quadrature: basic rationale

Instead of approximating the likelihood $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)$ with a GP, Ref. [26] proposed to approximate the proxy function defined as the square root $g\left(\boldsymbol{\theta}\right)=\sqrt{2\left(p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)-\alpha\right)}$ with a GP such that $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)=\alpha+0.5g^2\left(\boldsymbol{\theta}\right)$, where α is a small positive scalar, suggested to evolve with the training sample as $\alpha=0.8\min_{\boldsymbol{\theta}^{(l)}\in\boldsymbol{\Theta}}p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}^{(l)}\right)$. This transform has the benefit of halving the variation range of the function, which can be more suitable to be approximated by a GP. Then, by approximating the proxy function $g\left(\boldsymbol{\theta}\right)$ with the GP $\hat{g}_M\left(\boldsymbol{\theta}\right)\sim\mathcal{GP}\left(\mu_{g,M}\left(\boldsymbol{\theta}\right),c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right)$, the induced stochastic process $\hat{p}_M\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)=\alpha+0.5\hat{g}^2_M\left(\boldsymbol{\theta}\right)$ follows a non-central χ^2 with one degree of freedom, whose exact posterior mean and covariance do not admit closed-form expressions. Two tricks, e.g., linearization and moment matching, have been

developed for deducing approximate but closed-form expressions for these posterior features [26], and in this work, the linearization trick is utilized and introduced below.

By linearizing $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)=\alpha+0.5g^2\left(\boldsymbol{\theta}\right)$ at the mean $\mu_{g,M}\left(\boldsymbol{\theta}\right)$ with Taylor series, one obtain:

$$p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)\simeq\alpha+0.5\mu_{g,M}^2\left(\boldsymbol{\theta}\right)+\mu_{g,M}\left(\boldsymbol{\theta}\right)\left(g\left(\boldsymbol{\theta}\right)-\mu_{g,M}\left(\boldsymbol{\theta}\right)\right). \tag{12}$$

Following this linearization, the likelihood $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)$ can be approximated with a GP model $\hat{p}_M^{\mathcal{L}}\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)\sim\mathcal{GP}\left(\mu_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}\right),c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right)$, where the superscript '$\mathcal{L}$' refers to 'linearlization'. Based on Eq. (12), the posterior mean $\mu_{p,M}^{\mathcal{L}}$ and covariance $c_{p,M}^{\mathcal{L}}$ are then derived as:

$$\mu_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}\right)=\alpha+0.5\mu_{g,M}^2\left(\boldsymbol{\theta}\right) \tag{13a}$$

$$c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)=\mu_{g,M}\left(\boldsymbol{\theta}\right)c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\mu_{g,M}\left(\boldsymbol{\theta}'\right). \tag{13b}$$

It is clear that the above linearization trick enables approximating the likelihood with a GP model whose posterior mean and covariance admits closed-form expressions. Then, based on the Bayesian quadrature rule described by the first equalities of Eq. (9a) and Eq. (9b), the posterior mean and variance of the model evidence Z can be obtained as:

$$\begin{aligned}\mu_{Z,M}&=\Pi_p\left[\mu_{p,M}\left(\boldsymbol{\theta}\right)\right]\\&=\alpha+0.5\Pi_p\left[\mu_{g,M}^2\left(\boldsymbol{\theta}\right)\right]\\&=\alpha+0.5\left|2\boldsymbol{\Sigma}^{-1}+\mathbf{I}_d\right|^{-1/2}\mathcal{Y}^\top\mathcal{K}^{-1}\left[\kappa\left(\boldsymbol{\Theta},\boldsymbol{\Theta}\left|2\boldsymbol{\Sigma}\right.\right)\times\kappa\left(\boldsymbol{\Theta},-\boldsymbol{\Theta}\left|2\boldsymbol{\Sigma}+4\mathbf{I}_d\right.\right)\right]\mathcal{K}^{-1}\mathcal{Y}\end{aligned} \tag{14}$$

, and

$$\begin{aligned}\sigma_{Z,M}^2=&\Pi_p'\Pi_p\left[c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right]\\=&\Pi_p'\Pi_p\left[\mu_{g,M}\left(\boldsymbol{\theta}\right)c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\mu_{g,M}\left(\boldsymbol{\theta}'\right)\right]\\=&c\mathcal{Y}^\top\mathcal{K}^{-1}\left\{\begin{array}{l}\kappa\left(\boldsymbol{\Theta},\boldsymbol{\Theta}\left|3\boldsymbol{\Sigma}\right.\right)\times\kappa\left(\frac{4\boldsymbol{\Theta}}{3},-\frac{2\boldsymbol{\Theta}}{3}\left|2\boldsymbol{\Sigma}+4\mathbf{I}_d+\frac{2}{3}\boldsymbol{\Sigma}\right.\right)\\\times\kappa\left(\boldsymbol{u},\boldsymbol{v}\left|\left[\left(2\boldsymbol{\Sigma}+4\mathbf{I}_d\right)^{-1}+\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1}+\mathbf{I}_d\right.\right)\end{array}\right\}\mathcal{K}^{-1}\mathcal{Y}\\&-\left|2\boldsymbol{\Sigma}^{-1}+\mathbf{I}_d\right|^{-1}\mathcal{Y}^\top\mathcal{K}^{-1}\boldsymbol{\Omega}\mathcal{K}^{-1}\boldsymbol{\Omega}\mathcal{K}^{-1}\mathcal{Y}\end{aligned} \tag{15}$$

, respectively, where c is a constant computed by $c=\left|2\boldsymbol{\Sigma}^{-1}+\mathbf{I}_d\right|^{-1/2}\left|\left(2\boldsymbol{\Sigma}+4\mathbf{I}_d\right)^{-1}+\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}+\mathbf{I}_d\right|^{-1/2}$, $\boldsymbol{\Sigma}=\text{diag}\left\{\sigma_1^2,\sigma_2^2,\cdots,\sigma_d^2\right\}$ is a d-dimensional diagonal matrix with its i-th element being the scale-length parameter of the i-th dimension, and, $\boldsymbol{u}$, $\boldsymbol{v}$ as well as $\boldsymbol{\Omega}$ are all matrices of dimension $d\times d$, which are

computed by:

$$\boldsymbol{u} = \left[\frac{\Theta}{3}\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1} - \boldsymbol{\Theta}\left(2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)^{-1}\right]\left[\left(2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1} \tag{16a}$$

$$\boldsymbol{v} = -\frac{2\boldsymbol{\Theta}}{3}\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\left[\left(2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1} \tag{16b}$$

$$\boldsymbol{\Omega} = \kappa\left(\boldsymbol{\Theta}, \boldsymbol{\Theta}\,|2\boldsymbol{\Sigma}\right) \times \kappa\left(\boldsymbol{\Theta}, -\boldsymbol{\Theta}\,|2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right). \tag{16c}$$

In Eqs. (15)-(16), the notation $\kappa\left(\cdot,\cdot|\boldsymbol{\Sigma}^*\right)$ is used, and it is defined as the squared kernel governed by the weight matrix $\boldsymbol{\Sigma}^*$, and thus, e.g. for $\boldsymbol{u}$ and $\boldsymbol{v}$, the resultant covariance matrix is computed by:

$$\kappa\left(\boldsymbol{u}, \boldsymbol{v}|\boldsymbol{\Sigma}^*\right) = \sigma_0^2 \exp\left(-\frac{1}{2}\left(\boldsymbol{u} - \boldsymbol{v}\right)^\top \Sigma^{*-1}\left(\boldsymbol{u} - \boldsymbol{v}\right)\right). \tag{17}$$

The above closed-form expressions, adapted from the supplementary material of Ref. [37], are applicable for the case with Gaussian prior and squared exponential kernel.The details of the mathematical deductions of these expressions are also reported in Appendix B. Although those closed-form expressions appears to be complex, while being correctly used, it makes the algorithm being numerically efficient and robust. Unfortunately, for most of the other pairs of kernel and prior density, e.g., Matérn kernel, the closed-form expressions are intractable, and should resort to numerical method, e.g., Monte Carlo simulation, for estimating these quantities.

3.2. Generalized Posterior Variance Contribution Function for Active Learning

Based on the linearization trick and posterior covariance expressed by Eq. (13b), the PVC function for active learning of evidence can then be formulated as:

$$\mathcal{A}_{\mathrm{PVC}}^{\mathcal{L}}\left(\boldsymbol{\theta}\right) = \rho_M\left(\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \tag{18}$$

, where $\rho_M\left(\boldsymbol{\theta}\right)$ is a quantity of the same order with the posterior variance $\hat{p}_M^{\mathcal{L}}\left(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta}\right)$, and is defined by integrating the posterior covariance $c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ over the whole support of $\boldsymbol{\theta}'$, i.e.,

$$\rho_M\left(\boldsymbol{\theta}\right) = \Pi_p'\left[c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right] \tag{19}$$

Compared with the posterior variance $c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}\right)$, $\rho_M\left(\boldsymbol{\theta}\right)$ can also be interpreted as a 'variance', but since it integrates the spatial correlation information of the posterior GP model $\hat{p}_M^{\mathcal{L}}\left(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta}\right)$, it conveys richer information than $c_{p,M}^{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\theta}\right)$. It measures the contribution of the prediction uncertainty of $\hat{p}_M^{\mathcal{L}}\left(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta}\right)$ at an arbitrary location $\boldsymbol{\theta}$ to the prediction uncertainty of the model evidence Z. It is thus expected to achieve

the most reduction of prediction uncertainty of Z by including the location with the highest PVC value. The closed-form expression of $\rho_M(\boldsymbol{\theta})$ can be generated for squared kernel and Gaussian prior density as:

$$\begin{aligned}\rho_M(\boldsymbol{\theta}) =& b\mathcal{Y}^\top\mathcal{K}^{-1}\left[\kappa(\boldsymbol{\theta},-\Theta\,|2\boldsymbol{\Sigma}+4\mathbf{I}_d)\times\kappa\left(\boldsymbol{\theta}-\frac{2\Theta}{3},\frac{\Theta}{3}\middle|\frac{2}{3}\boldsymbol{\Sigma}\right)\times\kappa(\Theta,\Theta\,|3\boldsymbol{\Sigma})\right]\mathcal{K}^{-1}\mathcal{Y}\\ &-b\mathcal{Y}^\top\mathcal{K}^{-1}\left[\kappa(\Theta,\boldsymbol{\theta})\times\kappa(\boldsymbol{\theta},\Theta)\right]\mathcal{K}^{-1}\left[\kappa(\Theta,\Theta\,|2\boldsymbol{\Sigma})\times\kappa(\Theta,-\Theta\,|2\Sigma+4\mathbf{I}_d)\right]\mathcal{K}^{-1}\mathcal{Y}\end{aligned} \tag{20}$$

, where b is a constant computed by $b=\left|2\boldsymbol{\Sigma}^{-1}+\mathbf{I}_d\right|^{-1/2}$. One can refer to Appendix C for the mathematical details of deducing Eq. (20).

Despite the above appealing advantage, the PVC function can be further improved by making a better trade-off between exploration and exploitation, and this can be realized by properly incorporating the information reflected by the posterior mean $\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})$. To achieve this, following the basic idea of devising the generalized uncertainty sampling function presented in [38], a family of generalized PVC (GPVC) functions are proposed as:

$$\mathcal{A}^{\mathcal{L}}_{\mathrm{GPVC}}(\boldsymbol{\theta})=\rho^{\alpha}_M(\boldsymbol{\theta})\,p^{\beta}(\boldsymbol{\theta})\exp\left(\gamma\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})\right) \tag{21}$$

, with $\alpha,\beta,\gamma\geqslant 0$, where the extra term $\exp\left(\gamma\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})\right)$ is introduced to make a trade-off between exploration and exploitation. For all Bayesian numerical methods, which also includes Bayesian optimization [39, 40] and Bayesian reliability analysis [41], the performance of an acquisition function on balancing the exploration and exploitation is highly responsible for performance of the convergence speed, where 'exploration' means the ability of exploring the regions with high uncertainty (measured by $\rho_M(\boldsymbol{\theta})$), and 'exploitation' refers to the ability of exploiting the regions with high posterior probability. Thus, with the definition of Eq. (21), the values of α and γ provide a flexibility for balancing exploration and exploitation. Given fixed values of β and γ, increasing α will results in more focus on exploration, which is desired when the number of quadrature points get larger. Thus, it is also suggested to specify α as iteration-dependent. Specifically, given $\beta,\gamma=1$, α can set as $\alpha(M)=\max(1,\ln(M))$ or $\alpha(M)=\sqrt{M}$. It is noted that, setting $\alpha,\beta=1$ and $\gamma=0$ results in the classical PVC function of defined by Eq. (18). In the case study of this work, the effects of the options of the three parameters α, β and γ will be investigated. In these investigations, the particle swarm algorithm will be utilized for searching the global optima of the acquisition functions, and for ensuring numerical stability, the target function will be set as the logarithm of the acquisition functions, in case the values of an acquisition function may cover several orders of magnitude. This means to search the location $\boldsymbol{x}^+$ of the next training point by maximizing the following acquisition function:

$$\ln\mathcal{A}^{\mathcal{L}}_{\mathrm{GPVC}}(\boldsymbol{\theta})=\alpha\ln\rho_M(\boldsymbol{\theta})+\beta\ln p(\boldsymbol{\theta})+\gamma\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta}) \tag{22}$$

For standard Gaussian prior, $\ln p(\boldsymbol{\theta})\propto-\boldsymbol{\theta}^\top\boldsymbol{\theta}/2$, and then substituting Eq.(20) and (14) into Eq.(22), one

can obtain the closed-form expression of $\ln \mathcal{A}^{\mathcal{L}}_{\mathrm{GPVC}}(\boldsymbol{\theta})$, which is cheap for optimization with any heuristic optimization algorithm. In this work, the particle swarm algorithm is utilized.

For non-Gaussian but explicit prior density, the Rosenblatt transformation can be utilized for transferring the formulation of the evidence as an integral over standard Gaussian density. Denote by $P(\boldsymbol{\theta})$ and $P_{1:i}(\boldsymbol{\theta}_{1:i})$ the joint cumulative distribution functions (CDFs) of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{1:i}$ respectively, with $\boldsymbol{\theta}_{1:i} = (\theta_1, \theta_2, \cdots, \theta_i)^\top$. Let $\Phi(\vartheta_i)$ be the univariate CDF of a standard Gaussian variable ϑ_i. The i-th component $\mathcal{R}_i(\boldsymbol{\theta})$ of the Rosenblatt transformation, denoted as $\boldsymbol{\vartheta} = \mathcal{R}(\boldsymbol{\theta})$, is formulated as [42]:

$$\mathcal{R}_i(\boldsymbol{\theta}) = \Phi^{-1}\left[P_{i|1:i-1}(\theta_i|\boldsymbol{\theta}_{1:i-1})\right] \tag{23}$$

, where $P_{i|1:i-1}(\theta_i|\boldsymbol{\theta}_{1:i-1}) = P_{1:i}(\boldsymbol{\theta}_{1:i}) / P_{1:i-1}(\boldsymbol{\theta}_{1:i-1})$ refers to the CDF of θ_i conditional on the former $i-1$ parameters $\boldsymbol{\theta}_{1:i-1}$. Owning to the reversibility, the inverse Rosenblatt transformation can be formulated as:

$$\boldsymbol{\theta} = \mathcal{R}^{-1}(\boldsymbol{\vartheta}). \tag{24}$$

Substituting Eq. (24) into Eq. (4) yields:

$$Z = \int_{\mathbb{T}} p\left(\mathcal{D}_{\mathrm{obs}}|\mathcal{R}^{-1}(\boldsymbol{\vartheta})\right)\phi(\boldsymbol{\vartheta})\,\mathrm{d}\boldsymbol{\vartheta} \tag{25}$$

, where $\phi(\boldsymbol{\vartheta})$ refers to the density of standard Gaussian vector $\boldsymbol{\vartheta}$. Solving Eq. (25) with the developed adaptive Bayesian quadrature, a posterior density of $\boldsymbol{\vartheta}$ can be formulated as:

$$q(\boldsymbol{\vartheta}|\mathcal{D}_{\mathrm{obs}}) = p\left(\mathcal{D}_{\mathrm{obs}}|\mathcal{R}^{-1}(\boldsymbol{\vartheta})\right)\phi(\boldsymbol{\vartheta}) \tag{26}$$

, and is approximated by a GP model with prediction uncertainty refined to desired accuracy. Then the posterior density $p(\boldsymbol{\theta}|\mathcal{D}_{\mathrm{obs}})$ can be ultimately formulated as:

$$p(\boldsymbol{\theta}|\mathcal{D}_{\mathrm{obs}}) = q\left(\mathcal{R}(\boldsymbol{\theta})|\mathcal{D}_{\mathrm{obs}}\right). \tag{27}$$

To summary, we need first to apply the inverse Rosenblatt transformation to estimate the evidence Z and the posterior density $q(\boldsymbol{\vartheta}|\mathcal{D}_{\mathrm{obs}})$, with desired accuracy, by solving the integral of Eq. (25) using the standard Bayesian quadrature scheme, and then apply the forward Rosenblatt transformation using Eq. (27) to obtain a probabilistic estimate for the targeted posterior density $p(\boldsymbol{\theta}|\mathcal{D}_{\mathrm{obs}})$.

3.3. Summary of Algorithm

Based on the developments above, the pseudo-code of the adaptive Bayesian quadrature with GPVC as acquisition function is then summarized in Algorithm 1. The inputs to the algorithm include the prior density

$p(\boldsymbol{\theta})$, the likelihood function $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)$, the initial training sample size M_0, and the stopping threshold ϵ. M_0 can be set as $6 \sim 15$, depending on the dimension of $\boldsymbol{\theta}$, and the initial samples can be generated by e.g., Latin-hypercube sampling. The threshold ϵ can be determined based on the users' tolerance to numerical error, and generally, it is suggested to be $2 \sim 5\%$. The outputs of the algorithm then includes, a posterior Gaussian distribution with mean $\mu_{Z,M}$ and variance $\sigma^2_{Z,M}$ for the evidence term Z, a posterior GP model $\hat{p}\left(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}}\right)$ with explicit posterior mean $\mu^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta}\right)$ and variance $c^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}\right)$, and the total number M of simulator calls. For practical implementation, it is suggested to use the delayed stopping judging strategy, that is to break the algorithm when the stopping condition is satisfied for multiple times (e.g., 2 times) in succession. This will helps to avoid the fake convergence when may appear in the early training stage.

Algorithm 1: Adaptive Bayesian Quadrature with GPVC as acquisition function

Input: $p\left(\boldsymbol{\theta}\right)$, $p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)$, M_0, ϵ.

Output: $\mu^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta}\right)$, $c^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}\right)$, $\mu_{Z,M}$, $\sigma^2_{Z,M}$, M

Generate initial training sample data $\mathcal{S} = \left\{\left(\boldsymbol{\theta}^{(j)}, y^{(j)}\right)\right\}_{j=1}^{M}$ of size M_0, using, e.g., Latin-hypercube sampling, where $y^{(j)} = \sqrt{2\left(p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}^{(j)}\right) - \alpha\right)}$;

while $1 == 1$ **do**

  Train a GP model $\hat{g}_M\left(\boldsymbol{\theta}\right)$ based on $\mathcal{S}$;

  Induce the posteriro mean $\mu^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta}\right)$ and covariance $c^{\mathcal{L}}_{p,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)$ of the GP model $\hat{p}^{\mathcal{L}}_M\left(\mathcal{D} \mid \boldsymbol{\theta}\right)$ with Eq. (13a) and Eq. (13b) respectively;

  Compute posterior mean $\mu_{Z,M}$ and variance $\sigma^2_{Z,M}$ of evidence based on Eq. (14) and Eq. (15);

  Evaluate the coefficient of variation $\text{COV}_{Z,M} = \sigma_{Z,M}/\mu_{Z,M}$;

  if $\text{COV}_{Z,M} < \epsilon$ **then**

    break while;

  else

    Infer the GPVC function $\mathcal{A}^{\mathcal{L}}_{\text{GPVC}}\left(\boldsymbol{\theta}\right)$ by Eq. (21);

    Calculate the next training sample $\boldsymbol{\theta}^{+}$ by maximizing Eq. (22) using, e.g., particle swarm algorithm;

    Evaluate the likelihood function $p\left(\mathcal{D}_{\text{obs}} \mid \boldsymbol{\theta}^{+}\right)$ at $\boldsymbol{\theta}^{+}$;

    Compute $y^{+} = g\left(\boldsymbol{\theta}^{+}\right) = \sqrt{2\left(p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}^{+}\right) - \alpha\right)}$;

    Add the training sample $\left\{\left(\boldsymbol{\theta}^{+}, y^{+}\right)\right\}$ to the training data set $\mathcal{S}$, and let $M = M + 1$;

  end

end

4. Case studies

In this section, we use one numerical examples without physical meanings and two engineering examples with physical backgrounds to illustrate the proposed method, and to prove the superiority of the proposed GPVC function over the GUS function. To be fair, we set five cases for both acquisition functions with different values for the three parameters α, β, and γ, see Table 1 for details. The three parameters have distinct effects on the performance of the algorithm. Higher value of α tends to put more emphasis on exploration, while large value of γ lead to more emphasis on exploitation, and large value of β leads to a preference of exploring the area with large value of prior density. At the current stage, it is still impossible for use to have a theoretical and quantitative conclusions of their exact effect on the convergence rate. Based on our vast numerical investigation, α is suggested to be higher than 0.5, and can also increase with the sample size; β can be set a value between 0.5 and 2; and γ should better be less than 2. Some of the five settings in Table 1 are borrowed from good practices of using the GUS acquisition functions (see e.g., [26, 31, 38]). Besides, for avoiding fake convergence, the algorithm is only stopped when the stopping condition $\mathrm{COV}_{Z,M} < \Delta^{\mathrm{COV}}$ is satisfied for two successive iterations, where the threshold Δ^{COV} is set to be 2% for all cases.

Table 1: Settings of parameters for the two acquisition functions used in the case studies.

Case ID	α	β	γ
Case 1	1	1	0
Case 2	1	2	0
Case 3	1	1	1
Case 4	$\sqrt{M}$	1	1
Case 5	$\ln(M)$	1	1

4.1. A Numerical Illustrative Example

For illustration and comparison, we consider a computational model expressed as:

$$\mathcal{M}(\boldsymbol{\theta}, x) = \theta_1 \sin(\theta_2 x) \tag{28}$$

, where θ_1 and θ_2 are two model parameters to be calibrated, whose true values are 1 and 0.2 respectively, and x is the controllable input. It is assumed that at five locations $x_{\mathrm{obs}} = (-0.5,\ 1,\ 0,\ 0.6,\ 0.5)$, measurement is conducted and the corresponding values are $y_{\mathrm{obs}} = (0.0077,\ 0.5654,\ -0.4518,\ 0.0.2921,\ 0.3593)$. The measurement noise is assumed to be white noise with standard deviation being 0.2. Thus, the likelihood function is defined as $p(\mathcal{D}_{obs}|\boldsymbol{\theta}) = \prod_{k=1}^{5} \phi\left(y_{obs}^{(k)} - \mathcal{M}\left(\boldsymbol{\theta}, x^{(k)}\right)|0.2^2\right)$, where $\phi\left(\cdot|0.2^2\right)$ refers to the Gaussian

density with zero mean and variance 0.2^2, and '$\cdot$' means the argument. The prior density of $\boldsymbol{\theta}$ is assumed to be standard Gaussian.

The Bayesian quadrature is implemented with GPVC and generalized uncertainty sampling (GUS) as acquisition function for each case, and to be fair for comparison, the algorithm with either acquisition function is implemented repeatedly for ten times and the average results for the evidence Z are then reported in Table 2. The reference value of Z is estimated by MCS using 5×10^4 samples, and is 0.0801 with the corresponding COV being 0.0098, and thus is sufficiently accurate. It can be seen that, with either GPVC or GUS under any of the five cases, the evidence Z is accurately estimated as all the mean estimates are very close to the reference results and the posterior COVs of all estimates are less than 2%. This demonstrated the high accuracy of the Bayesian quadrature for estimating the evidence. It is also seen that from the last column of Table 2, for all the five cases, the algorithms consumes much less number of simulator calls on average. The distribution of the number of simulator calls across the ten replications are then reported in Figure 1, and it is obvious that the algorithm with GPVC consumes smaller number of the simulator calls on average, and also the dispersion is obviously smaller than that consumed by GUS. For further illustrating the convergence process of the algorithm, the evolution of the confidence interval of Z generated with GPVC and GUS under case 3 is compared in Figure 2. It is seen again that, to reach the same level of numerical accuracy, the GPVC function consumes smaller number of model calls, and also it is empirically show that the GPVC function has a higher convergence rate. It is seen again that the proposed GPVC function results in faster convergence than the GUS function.

Table 2: Results for the first test example.

Case	Setting	Method	Means	COVs	M
Case 1	$\alpha = 1,\ \beta = 1,\ \gamma = 0$	GUS	0.0746	0.0192	92.9
		GPVC	0.0747	0.0185	83.3
Case 2	$\alpha = 1,\ \beta = 2,\ \gamma = 1$	GUS	0.0775	0.0194	111.7
		GPVC	0.0761	0.0185	108.1
Case 3	$\alpha = 1,\ \beta = 1,\ \gamma = 1$	GUS	0.0750	0.0190	104.6
		GPVC	0.0769	0.0188	91.6
Case 4	$\alpha = \sqrt{M},\ \beta = 1,\ \gamma = 1$	GUS	0.0799	0.0190	139.9
		GPVC	0.0764	0.0187	82
Case 5	$\alpha = \log(M),\ \beta = 1,\ \gamma = 1$	GUS	0.0768	0.0187	111
		GPVC	0.0759	0.0182	83.3
Reference		MCS	0.0801	0.0098	5×10^4

From Table 2, it is seen again that, given the same level of error tolerance, the proposed GPVC under parameter settings of case 1, case 4 and case 5 consumes the least simulator calls. Compared with GUS acquisition functions, the proposed GPVC function under setting of any cases is more efficient than the classical GUS functions.

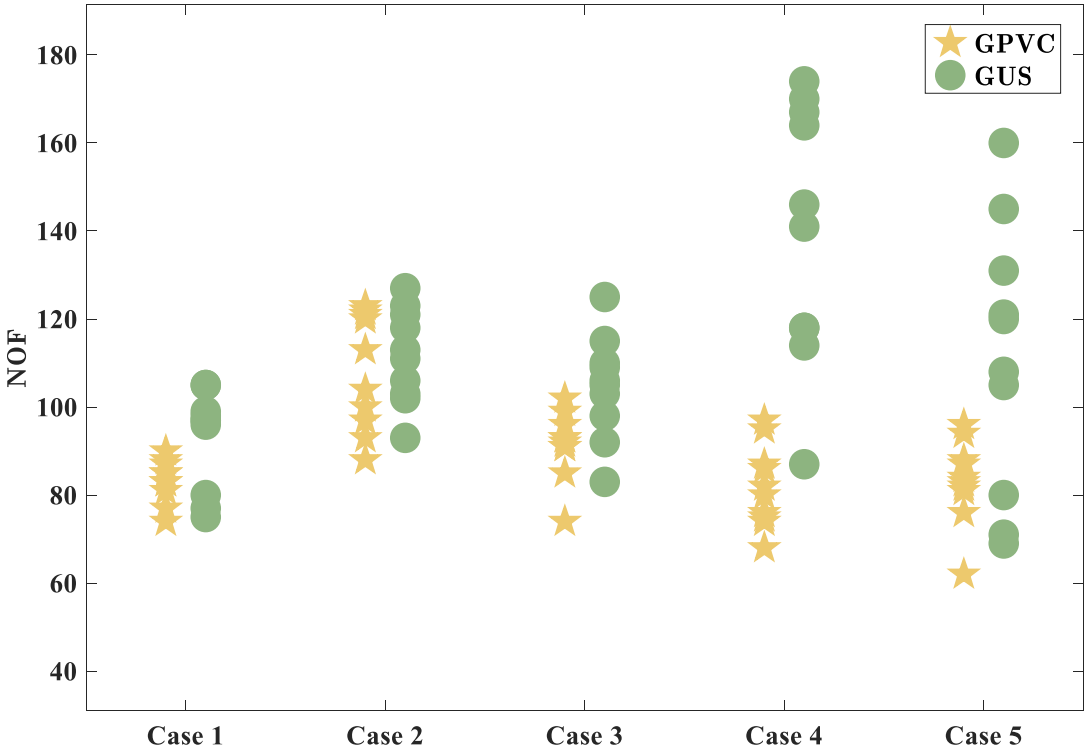


Figure 1: Distribution of the number of simulator calls for all cases of the first example across ten replications.

We then discuss the results for the posterior density $p(\theta_1, \theta_2|\mathcal{D})$, which are generated under setting of case 3, and are reported in Figure 3. The true posterior density is also shown in the upper left panel for illustrating the accuracy, and the posterior samples generated with the TMCMC are reported in the upper right panel for comparison. The results generated under the setting 3 with GPVC and GUS as acquisition functions are shown in the two lower panels. It is seen that the true posterior density shows bimodal feature, which is caused by the non-monotonic behavior of the model response with respect to the model parameters. This example is designed on purpose as this to demonstrate the ability of the proposed method for capturing the multi-modal feature. It is shown that, although the posterior density show multi-modal behavior, both GUS and GPVC produces accurate estimate of it, and also the samples produced by TMCMC match well with the reference density. The absolute errors and the posterior STDs (STandard Deviations) of the estimates reported in Figure 3 are then shown in Figure 4. As can be seen, even for the posterior density, both types of errors are sufficiently, proving the high robustness of the estimates. It is also seen from

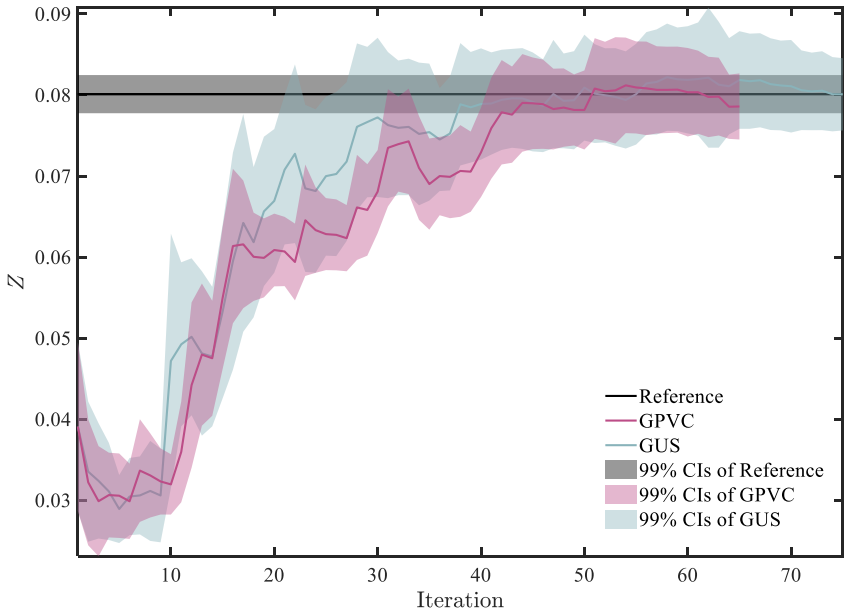


Figure 2: Evolution of posterior confidence intervals of Z with respect to the iteration for case 3 of example 1.

Figure 3 that the probability density covers the true value of θ_1 and θ_2, which are 1 and 0.2 respectively with high probability. The uncertainty related to the posterior distribution is caused by the measurement noise and multi-modal behavior of the model functions, which are both properly accommodated by the posterior distribution. In terms of computational cost, the classical TMCMC commonly requires at least several thousands of simulator calls. Comparably, the Bayesian quadrature with either GUS and GPVC consumes approximately one hundred simulator calls, which is significantly smaller than that of TMCMC. Thus, these results sufficiently demonstrated the high efficiency and accuracy of the proposed method estimating both the evidence and the posterior density.

4.2. A two degrees of freedom structural dynamic model

The dynamic model of a two-storied shear building modified from Ref. [43] is then adopted, and is schematically shown in Figure 5. The masses of the first and seconds floor are given as $m_1 = 16531$ kg and $m_2 = 16131$ kg, respectively. The values of stiffness of the inter-story are assumed to be deterministic-but-unknown and depends on two dimensionless parameters θ_1 and θ_2, i.e., $k_1 = \theta_1 k$ and $k_2 = \theta_2 k$, with $k = 29.7 \times 10^6$ N/m. The problem is then formulated as updating of the density of the two deterministic-but-unknown parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$, given the measurements of the natural frequencies of the two floors. The prior distribution of θ_1 and θ_2 are assumed to be of lognormal, i.e., $\theta_1 \sim \mathrm{LN}(1.3, 1)$ and $\theta_2 \sim \mathrm{LN}(0.8, 1)$, respectively, and their joint prior density is denoted by $p(\boldsymbol{\theta})$. The values of the frequencies are measured to be $(f_{\mathrm{obs},1}, f_{\mathrm{obs},2})^{\mathrm{T}} = (3.13, 9.83)$. By integrating the prior information and measurements, the posterior

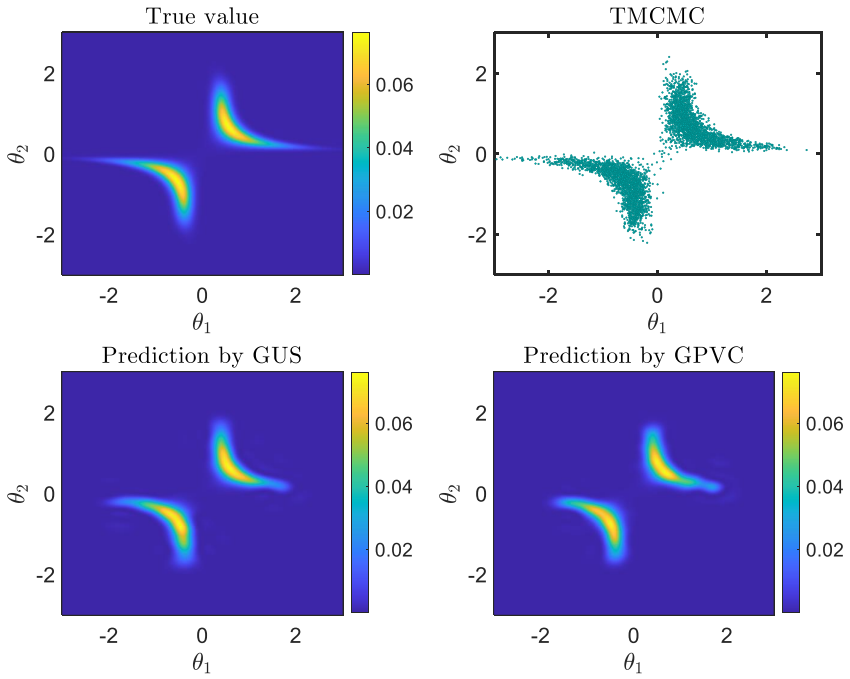


Figure 3: Comparison of results for posterior density of the first example, where PVC and VUS are used under setting of case 3, and the corresponding number of simulator calls are $N_{\mathrm{PVC}} = 104$ and N_{VUS}=114.

PDF of parameters $\boldsymbol{\theta}$ is formulated as

$$p\left(\boldsymbol{\theta} \mid \mathcal{Y}\right) \propto \exp\left[-J\left(\boldsymbol{\theta}\right)\right] p\left(\boldsymbol{\theta}\right) \tag{29}$$

, where the term $J\left(\boldsymbol{\theta}\right)$ denotes function of the fitness between the measurements and the analytical values, and is expressed as

$$J\left(\boldsymbol{\theta}\right) = \frac{1}{2} \sum_{j=1}^{2} \left[\frac{f_{\mathrm{ana},j} - f_{\mathrm{obs},j}}{f_{\mathrm{obs},j} \sigma_R} \right]^2 \tag{30}$$

, with $\sigma_R = 1/16$ representing the STD of the measurement noise. The two natural frequencies are calculated from the dynamic model as:

$$\begin{aligned} f_{\mathrm{ana},1} &= \frac{1}{2\pi} \sqrt{\frac{k_1 m_2 + k_2 m_1 + k_2 m_2 - c}{2 m_1 m_2}} \\ f_{\mathrm{ana},2} &= \frac{1}{2\pi} \sqrt{\frac{k_1 m_2 + k_2 m_1 + k_2 m_2 + c}{2 m_1 m_2}} \end{aligned} \tag{31}$$

, where

$$c = \sqrt{k_1^2 m_2^2 - 2 k_1 k_2 m_1 m_2 + 2 k_1 k_2 m_2^2 + k_2^2 m_1^2 + 2 k_2^2 m_1 m_2 + k_2^2 m_2^2}. \tag{32}$$

Under the above setting, we use the proposed method to numerically infer the posterior density of $\boldsymbol{\theta}$.

Similarly, the algorithm is implemented with both GPVC and GUS under the five cases of parameter

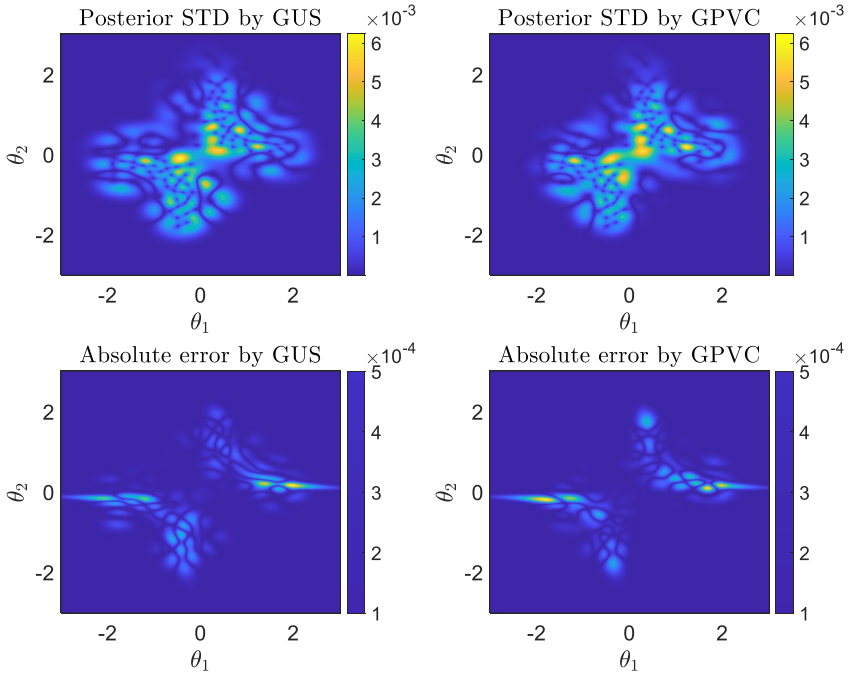


Figure 4: Results of PVC for case 3 ($N_{\text{PVC}} = 104$, N_{VUS}=114).

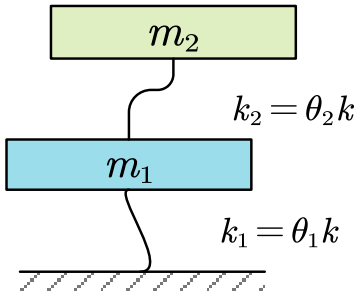


Figure 5: A two-storied shear building model.

setting, and for each setting, the algorithm is repeated for ten times, and the results of posterior means and COVs of Z as well as the average number of model calls are reported in Table 3. The reference results are also computed by MCS with 5×10^4 samples, and the corresponding COV is 1.57%. It is found again that the algorithm with either GUS or GPVC as acquisition function under any of the five settings produce accurate estimate of Z with posterior COVs being less than 2%, indicating the high accuracy of the proposed method. The distribution of the model calls consumed for each implementation is reported in Figure 6. It is seen that, except for the setting of case 2, the GPVC function consumes less model calls, and even for case 2, GPVC consumes almost the same number of model calls as GUS on average. This sufficiently demonstrates the superiority of GPVC over GUS.

Table 3: Results of evidence Z for the 2-DOF dynamic model.

Case	Setting	Method	Means	COVs	M
Case 1	$\alpha = 1,\ \beta = 1,\ \gamma = 0$	GUS	0.0473	0.0176	53.3
		GPVC	0.0470	0.0173	50
Case 2	$\alpha = 1,\ \beta = 2,\ \gamma = 1$	GUS	0.0473	0.0182	57.8
		GPVC	0.0469	0.0185	57.7
Case 3	$\alpha = 1,\ \beta = 1,\ \gamma = 1$	GUS	0.0470	0.0175	53.8
		GPVC	0.0469	0.0181	48.4
Case 4	$\alpha = \sqrt{M},\ \beta = 1,\ \gamma = 1$	GUS	0.0472	0.0172	55.5
		GPVC	0.0471	0.0177	46
Case 5	$\alpha = \log(M),\ \beta = 1,\ \gamma = 1$	GUS	0.0472	0.0169	57.7
		GPVC	0.0471	0.0178	46.9
Reference		MCS	0.0471	0.0157	5×10^4

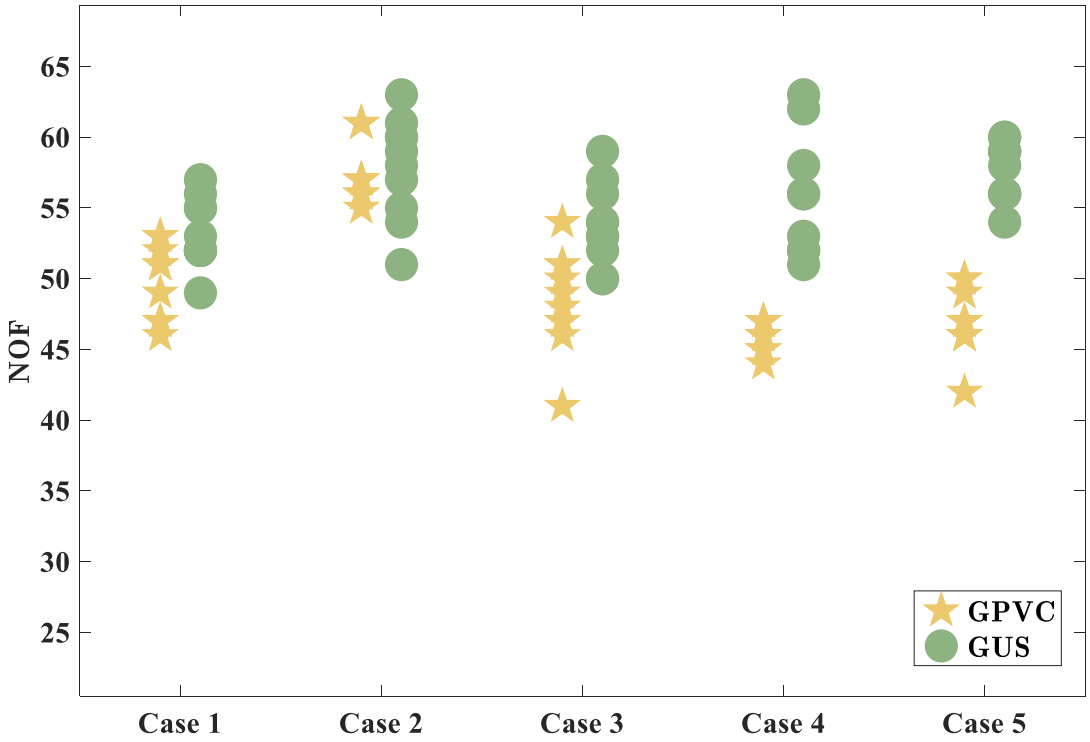


Figure 6: Distribution of the number of model calls for the 2-DOF dynamic model, where for each of the five settings, the GPVC and GUS functions are utilized and implemented for ten times

The results of posterior density computed under the setting of case 3 are reported in the two lower panels

of Figure 7, together with the reference solution generated analytically and by TMCMC in the two upper panels. The number of model calls consumed by the GPVC and GUS are 45 and 54 respectively. The absolute error and posterior STDs of the estimates by GUS and PVC are displayed in Comparison of GUS and GPVC for case 3 ($N_{\mathrm{PVC}} = 45$, N_{VUS}=54). The absolute errors as well as the posterior STDs of these results are reported in Figure 8. It is seen that, both GPVC and GUS produce accurate and robust estimate of the posterior density, but the GPVC function consumes the least model calls. The evolution of the posterior confidence intervals of Z against the iteration steps for these two implementations is schematically compared in Figure 9, and it is shown apparently that, for the two implementations, the GPVC function results in higher convergence rate that the GUS function. All these results demonstrates the high efficiency and accuracy of the proposed GPVC function for estimation of both the evidence Z and the posterior density.

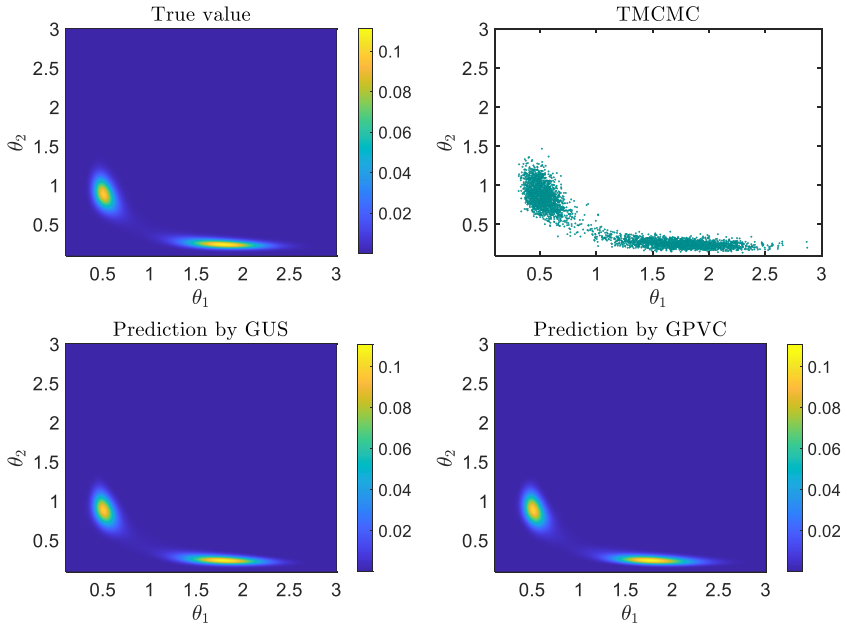


Figure 7: Results of posterior density for the 2-DOF dynamic model, where the GPVC .

4.3. Lubrication model of journal bearing of aero-engine gear pump

The journal bearing is one of the most important components of a aero-engine gear fuel pump as it provides support for the extreme loads excited by the internal flow. The wear failure is one of the main failure modes, and thus the lubrication performance is one of the most concern in the design stage. To predict the lubrication performance under different operating modes, and following Ref.[44], we have developed a complex simulator. In this case study, we use a simplified version of this mode to illustrate the proposed method for calibrating the parameters of this model. The geometric model of the gear pump and the flow domain of the lubrication film are schematically shown in Figure 10. It is assumed that, while operating,

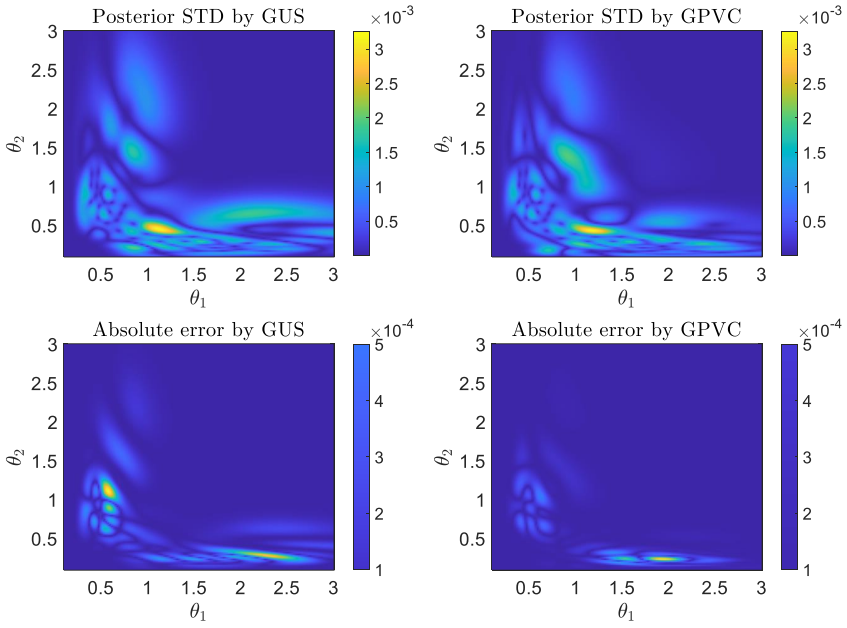


Figure 8: Absolute errors and posterior STDs of the posterior density estimated with approximate Bayesian quadrature.

the pressure and temperature can be measured, and used for model calibration. Practically, the maximum pressure $P_{\max}$ and the maximum temperature $T_{\max}$ are measured.

The pressure and temperature fields of the lubrication film are governed a set of PDE, which are the integration of the generalized Reynolds equation, energy equation [44], viscosity-temperature relationship, and are formulated as:

$$\frac{\partial}{R^2\partial\theta}\left(\frac{\partial p}{\partial\theta}F_2\right)+\frac{\partial}{\partial z}\left(\frac{\partial p}{\partial z}F_2\right)=\frac{\partial}{R\partial\theta}\left(U\frac{F_1}{F_0}\right), \tag{33}$$

$$\rho c_f\left(\frac{v_x}{R}\frac{\partial T_f}{\partial\theta}+v_y\frac{\partial T_f}{\partial y}+v_z\frac{\partial T_f}{\partial z}\right)=k_f\frac{\partial^2 T_f}{\partial y^2}+\eta\left[\left(\frac{\partial v_x}{\partial y}\right)^2+\left(\frac{\partial v_z}{\partial y}\right)^2\right] \tag{34}$$

, and

$$\eta=\eta_0\exp\left[-0.0298\left(T-T_0\right)\right] \tag{35}$$

, where $F_0=\int_0^h\frac{1}{\eta}dy$, $F_1=\int_0^h\frac{y}{\eta}dy$, and $F_2=\int_0^h\frac{y^2}{\eta}dy-\frac{\int_0^h\frac{y}{\eta}dy\int_0^h\frac{y}{\eta}dy}{\int_0^h\frac{1}{\eta}dy}$. The film thickness h can be computed as

$$h=C\left(1+\varepsilon\cos\left(\theta-\phi\right)\right), \tag{36}$$

and the velocity v_x and v_z can be computed by

$$\begin{aligned}v_x&=\frac{\partial p}{R\partial\theta}\left[\int_0^y\frac{y}{\eta}dy-\int_0^y\frac{1}{\eta}dy\frac{F_1}{F_0}\right]+\frac{U_s}{F_0}\int_0^y\frac{1}{\mu}dy\\ v_z&=\frac{\partial p}{\partial z}\left[\int_0^y\frac{y}{\eta}dy-\int_0^y\frac{1}{\eta}dy\frac{F_1}{F_0}\right]+W_b\left(1-\frac{\int_0^y\frac{1}{\mu}dy}{F_0}\right).\end{aligned} \tag{37}$$

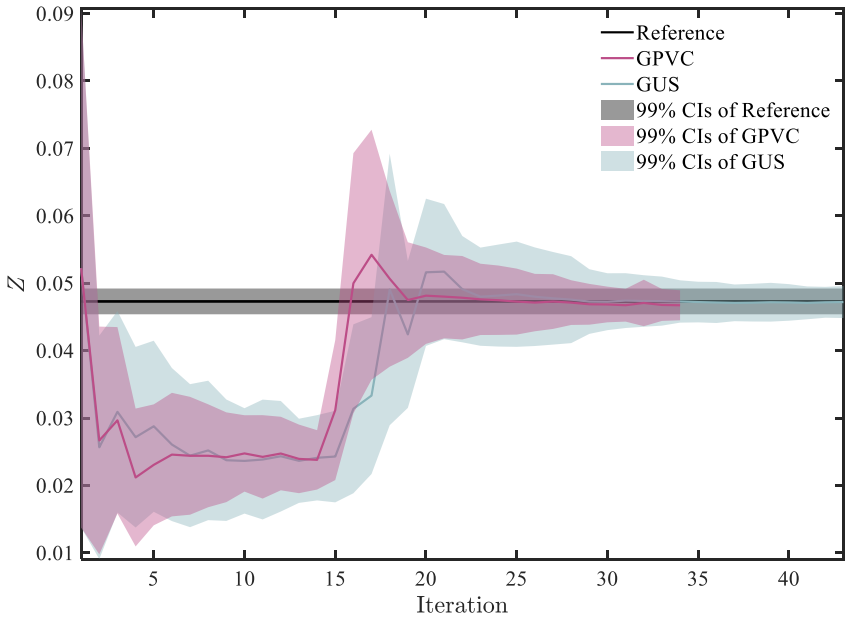


Figure 9: Evolution of evidence with respect to steps of iteration for the 2-DOF dynamic model.

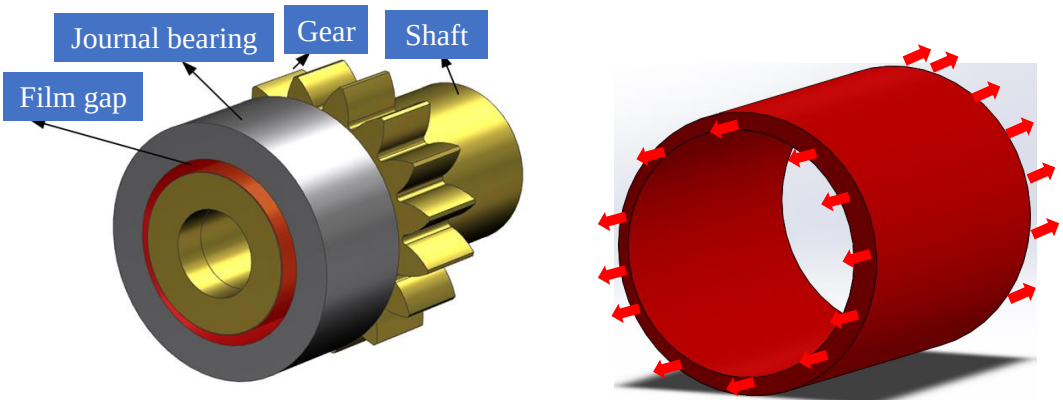
The model parameters are displayed in Table 4.

Table 4: Setting of parameters for the journal bearing.

Parameter	Value
Bearing's width B (mm)	37
Bearing's radius R (mm)	16
Radial gap C (mm)	0.06
Bulk modulus of film β (Pa)	5×10^8
Density of lubrication film ρ $\left(\mathrm{kg/m^3}\right)$	779
Velocity U (m/s)	$U = \pi R n_r/30$

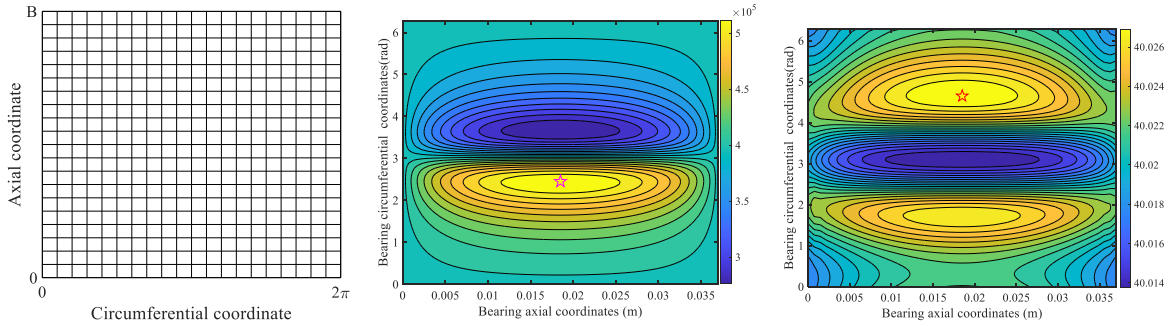
The finite difference method and the successive overrelaxation (SOR) method are used to solve the above equations, where the discrete computational area of film are given in Figure 11, together with the solved field of pressure, temperature and their maxima with eccentricity ratio, attitude angle, rotate speed and initial viscosity being set to 0.6, 0.1, 6200 and 9.78×10^{-4} respectively. Next, the observation data (527499.1407Pa, 40.0417C$^\circ$) is used to infer the unknown eccentricity ratio, attitude angle, rotate speed and initial viscosity angle of the real bearing structure by using the proposed method, where the prior distribution are assumed to be $N\left(0.4, 0.1^2\right)$, $N\left(0, 0.2^2\right)$, $N\left(6000, 200^2\right)$ and $N\left(9.5 \times 10^{-4}, \left(5 \times 10^{-5}\right)^2\right)$, respectively.

With the same parameter settings for algorithm, the results of the evidence Z are reported in Table 5, where the last column reports the average number of model calls across ten replications. Due to the high computational cost of each model call is expensive, the reference results are not computed by MCS, and the results by the two acquisition functions are crossly validated. As can be seen, the mean estimates by



(a) Bearing structure (b) Fluid domain

Figure 10: The structure of bearing and its fluid domain.



(a) Mesh (b) Pressure distribution (c) Temperature distribution

Figure 11: Illustration of mesh and the solved fields of pressure and temperature of the lubrication film.

all the replications match well with each other, thus are believed to accurate. Besides, it is also seen that the posterior COVs of all estimates are lower than 2%, indicating the robustness and high accuracy of the mean estimates again. The distribution of the numbers of model calls consumed by the algorithm under each setting case is schematically shown in Figure 12. From both the last column of Table 5 and Figure 12, it can be seen again, that GPVC consumes less simulator calls than GUS, although the superiority is not as significant as the last two examples. We found that, the GPVC usually show higher superiority for models with higher nonlinear behaviors.

Table 5: Results of Z for the lubrication model of journal bearing.

Case	Setting	Method	Means	COVs	M
Case 1	$\alpha = 1,\ \beta = 1,\ \gamma = 0$	GUS	0.0353	0.0083	20.5
		GPVC	0.0354	0.0113	21
Case 2	$\alpha = 1,\ \beta = 2,\ \gamma = 1$	GUS	0.0352	0.0134	23.8
		GPVC	0.0418	0.0132	28.2
Case 3	$\alpha = 1,\ \beta = 1,\ \gamma = 1$	GUS	0.0357	0.0128	19.4
		GPVC	0.0354	0.0106	17.4
Case 4	$\alpha = \sqrt{M},\ \beta = 1,\ \gamma = 1$	GUS	0.0354	0.0143	55
		GPVC	0.0344	0.0109	17.8
Case 5	$\alpha = \log(M),\ \beta = 1,\ \gamma = 1$	GUS	0.0347	0.0133	39.5
		GPVC	0.0356	0.0112	18.6

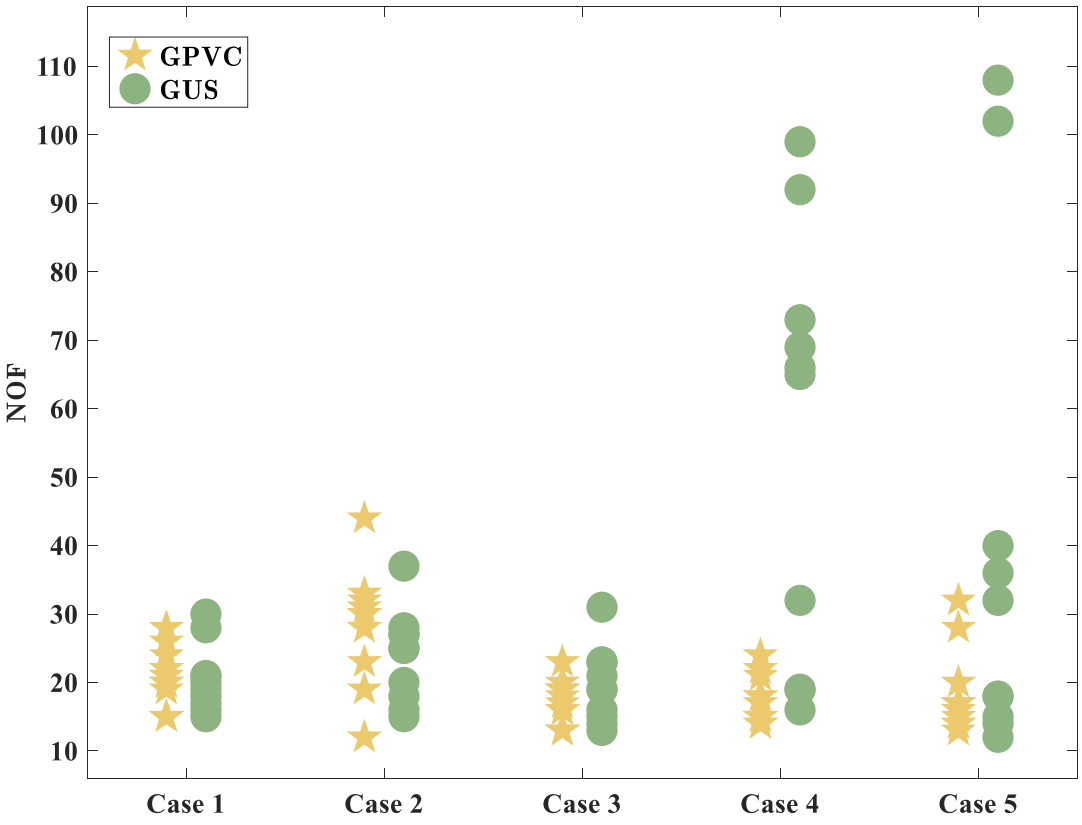


Figure 12: Distribution of the number of model calls consumed by each setting across ten replications for the lubrication model.

The results of the posterior density are then reported in Figure 13. As there are four parameters being

calibrated, it is impossible to show the joint density, and thus the results of the marginal density are reported together with their priors. It should be noted that these marginal posterior density of each $\boldsymbol{\theta}$, and the closed-form expression of the mean is reported in Appendix D. As can be seen form Figure 13, the posterior density estimated by GUS and GPVC match well, indicting the high accuracy. It is noted that, for each of the three parameters $\theta_2 \sim \theta_4$, the prior and posterior densities are almost the same. This is due to the fact that the model response (pressure and temperature of the film) are insensitive to these parameters, this has also been proved in our long-term test. For demonstrating this, the total effect variance-based sensitivity indices are estimated are reported in Figure 14 (see Ref. [45] for details). It is seen that the sensitivity indices of $\theta_2 \sim \theta_4$ are all close to zero, indicating they are non-influential. This is in good agreement with the above conclusions. This may motivates us to first perform sensitivity analysis before implementing the Bayesian updating of high-dimensional parameters. This can be substantially alleviate the challenge caused by the 'curse of dimensionality'. For the important parameter θ_1, the density has been significantly modified and the uncertainty has been substantially reduced. This, in turn, results in the reduction of the prediction uncertainty of the lubrication model.

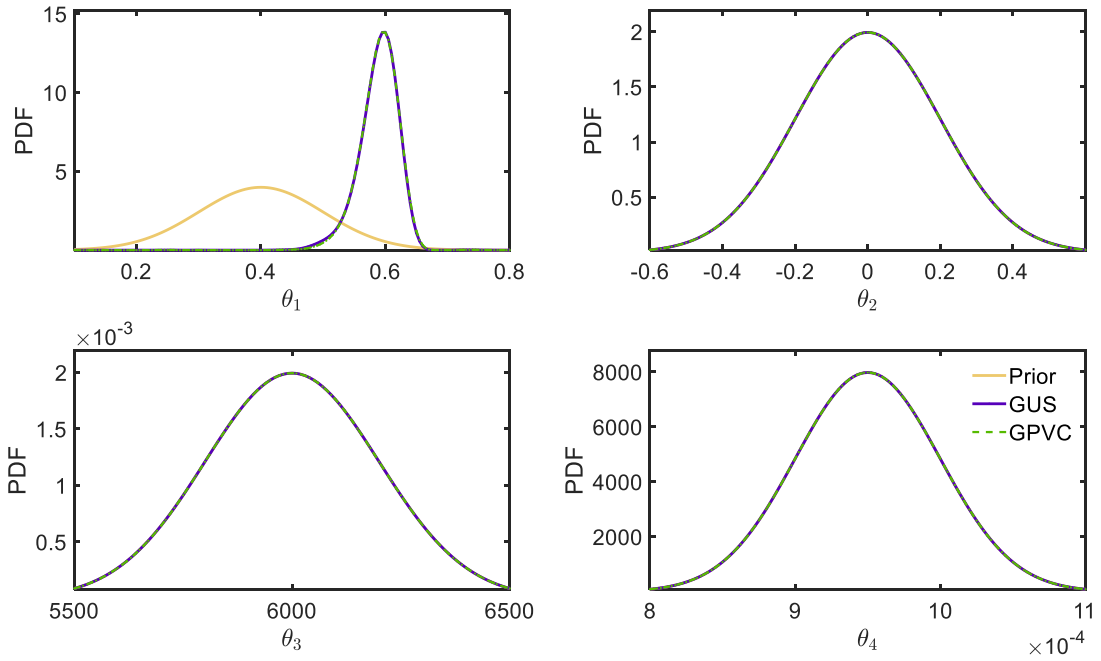


Figure 13: Results of the marginal posterior probability density function (PDF) of the lubrication models.

5. Conclusions

This work has developed an effective approximate Bayesian quadrature algorithm, driven by a well-designed acquisition function called GPVC, for adaptively inferring the evidence and posterior density for

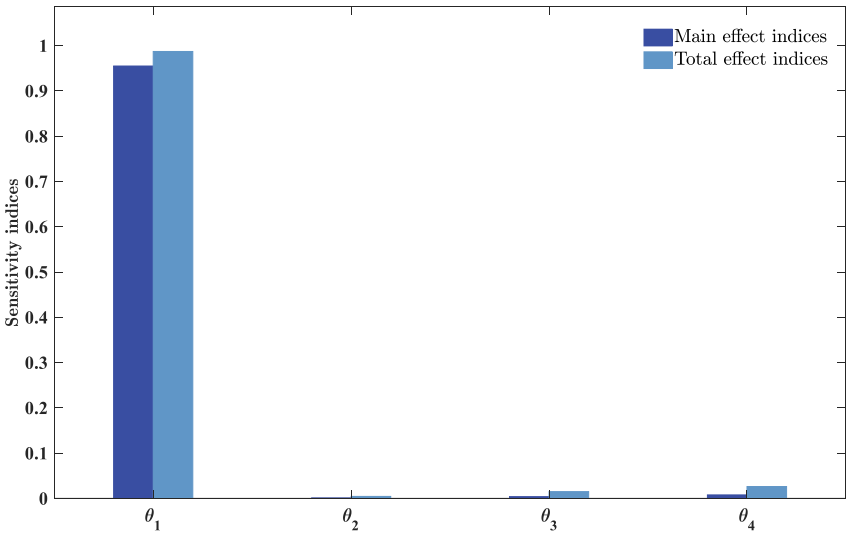


Figure 14: Results of variance-based sensitivity indices for the lubrication models.

BMU with desired accuracy. The PVC function was originally developed for exact Bayesian quadrature, and has not been investigated for BMU. Instead of simply extending it for BMU, a generalized version of it has been proposed for provide sufficient flexibility and potential for balancing the exploration and exploitation of the algorithm, and to accelerate the convergence. Another appealing feature of the active learning procedure is the high efficiency for each query, benefiting from the closed-form expressions of the GPVC acquisition function.

Results of three examples under extensive different settings has demonstrated that, given the same level of accuracy tolerance, the proposed GPVC function requires much less number of model calls, and thus is more efficient, even for the challenging task of estimating multi-modal posterior density. This conclusion has encouraged the utilization of the GPVC function combined with the approximate Bayesian quadrature for solving the BMU problem. Some further work needs to be extended. For example, a theoretical investigation of the convergence rate of the GPVC function, an extension of the method to cases with extremely small model evidence (typically, $Z < 10^{-4}$), and also an extension to the joint calibration of deterministic-but-unknown model parameters and model bias. All these will be conducted in future work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under grant number 52475164 and 72171194, as well as the Sino-German Mobility Program under grant number M-0175 (2021-2025).

Appendix A. Closed-form expressions for kernel means

We assume that the weight density $p(\boldsymbol{\theta})$ is of standard normal form with zero mean and diagonal unit covariance matrix. The closed-form expressions for the three kernel means $\Pi_p[\kappa(\boldsymbol{\theta}, \boldsymbol{\Theta})]$, $\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right]$ and $\Pi_p\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right]$ are given by:

$$\Pi_p[\kappa(\boldsymbol{\theta}, \boldsymbol{\Theta})] = \sigma_0^2 \left|\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \exp\left[-\frac{1}{2}\mathrm{vec}\left\{\mathrm{diag}\left[\boldsymbol{\Theta}\left(\boldsymbol{\Sigma} + \mathbf{I}_d\right)^{-1}\boldsymbol{\Theta}^\top\right]\right\}\right] \tag{A.1}$$

$$\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right] = \sigma_0^2 \left|\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \exp\left[-\frac{1}{2}\boldsymbol{\theta}\left(\boldsymbol{\Sigma} + \mathbf{I}_d\right)^{-1}\boldsymbol{\theta}^\top\right] \tag{A.2}$$

$$\Pi_p\Pi'_p\left[\kappa\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right] = \sigma_0^2 \left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \tag{A.3}$$

, where $\mathbf{I}_d$ refers to the M-dimensional identity matrix, $\boldsymbol{\Sigma} = \mathrm{diag}\left(\sigma_1^2, \sigma_2^2, \cdots, \sigma_d^2\right)$, and $\mathrm{vec}\{\mathrm{diag}[\cdot]\}$ is the operator of vectorizing the diagonal elements of a matrix.

Appendix B. Closed-form expressions for Approximate Bayesian Quadrature

For simplicity, we assume a zero prior mean for $\hat{g}$, i.e., $m(\boldsymbol{\theta}) \equiv 0$. When the linearization trick is utilized, the induced Gaussian distribution of the integral is denoted as $\hat{Z}_M \sim \mathcal{N}\left(\mu_{Z,M}, \sigma^2_{Z,M}\right)$. Before inferring the posterior mean and variance of integral $\hat{Z}_M = \Pi_p\left[\hat{p}^{\mathcal{L}}_M\left(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta}\right)\right]$, we derive the formula for multiplying two covariance functions, i.e.,

$$\kappa(\boldsymbol{\theta}, \boldsymbol{\xi}\,|\Sigma_1)\,\kappa(\boldsymbol{\theta}, \boldsymbol{\zeta}\,|\Sigma_2) = \kappa\left(\boldsymbol{\theta}, \left(\boldsymbol{\xi}\Sigma_1^{-1} + \boldsymbol{\zeta}\Sigma_2^{-1}\right)\left(\Sigma_1^{-\mathbf{1}} + \Sigma_2^{-\mathbf{1}}\right)^{-1}\middle|\left(\Sigma_1^{-\mathbf{1}} + \Sigma_2^{-\mathbf{1}}\right)^{-1}\right)\kappa(\boldsymbol{\xi}, \boldsymbol{\zeta}\,|\Sigma_1 + \Sigma_2), \tag{B.1}$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ denote any vector with size d, and $\kappa(\cdot, \cdot\,|*)$ represents the covariance of any two vector with covariance matrix being $*$. The posterior mean $\mu_{Z,M}$ is the expectation of $\mu_{p,M}(\boldsymbol{\theta})$, and thus is formulated as:

$$\begin{aligned}
\mu_{Z,M} &= \Pi_p\left[\mu_{p,M}(\boldsymbol{\theta})\right] \\
&= \alpha + 0.5\Pi_p\left[\mu^2_{g,M}(\boldsymbol{\theta})\right] \\
&= \alpha + 0.5\mathcal{Y}^\top\mathcal{K}^{-1}\Pi_p\left[\kappa(\boldsymbol{\Theta}, \boldsymbol{\theta})\,\kappa(\boldsymbol{\theta}, \boldsymbol{\Theta})\right]\mathcal{K}^{-1}\mathcal{Y} \\
&= \alpha + 0.5\mathcal{Y}^\top\mathcal{K}^{-1}\sum_{k,l}\Pi_p\left[\kappa\left(\Theta^{(k)}, \boldsymbol{\theta}\right)\kappa\left(\boldsymbol{\theta}, \Theta^{(l)}\right)\right]\mathcal{K}^{-1}\mathcal{Y}
\end{aligned} \tag{B.2}$$

Based on Eq. (B.1), and thus Eq. (B.4) can be further derived as:

$$\begin{aligned}
\mu_{Z,M} &= \alpha + 0.5\mathcal{Y}^\top\mathcal{K}^{-1}\left\{\sum_{k,l}\kappa\left(\Theta^{(k)}, \Theta^{(l)}\,|2\Sigma\right)\Pi_p\left[\kappa\left(\boldsymbol{\theta}, \frac{\Theta^{(k)} + \Theta^{(l)}}{2}\middle|\frac{1}{2}\Sigma\right)\right]\right\}\mathcal{K}^{-1}\mathcal{Y} \\
&= \alpha + 0.5\left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2}\mathcal{Y}^\top\mathcal{K}^{-1}\left[\kappa(\Theta, \Theta\,|2\Sigma) \times \kappa(\Theta, -\Theta\,|2\Sigma + 4\mathbf{I}_d)\right]\mathcal{Y}^\top\mathcal{K}^{-1}
\end{aligned} \tag{B.3}$$

The posterior variance $\sigma^2_{Z,M}$ is the expectation of $\mu_{g,M}(\boldsymbol{\theta})\, c_{g,M}(\boldsymbol{\theta},\boldsymbol{\theta}')\, \mu_{g,M}(\boldsymbol{\theta}')$, and thus is formulated as:

$$
\begin{aligned}
\sigma^2_{Z,M} &= \Pi_p \Pi_p' \left[\mu_{g,M}(\boldsymbol{\theta})\, c_{g,M}(\boldsymbol{\theta},\boldsymbol{\theta}')\, \mu_{g,M}(\boldsymbol{\theta}')\right] \\
&= \mathcal{Y}^\top \mathcal{K}^{-1} \Pi_p \Pi_p' \left[\kappa(\Theta,\boldsymbol{\theta})\, \kappa(\boldsymbol{\theta},\boldsymbol{\theta}')\, \kappa(\boldsymbol{\theta}',\Theta)\right] \mathcal{K}^{-1}\mathcal{Y} \\
&\quad - \mathcal{Y}^\top \mathcal{K}^{-1} \Pi_p \Pi_p' \left[\kappa(\Theta,\boldsymbol{\theta})\, \kappa(\boldsymbol{\theta},\Theta)\, \mathcal{K}^{-1} \kappa(\Theta,\boldsymbol{\theta}')\, \kappa(\boldsymbol{\theta}',\Theta)\right] \mathcal{K}^{-1}\mathcal{Y} \\
&= \mathcal{Y}^\top \mathcal{K}^{-1} \left\{ \sum_{k,l} \Pi_p \Pi_p' \left[\kappa\left(\Theta^{(k)},\boldsymbol{\theta}\right) \kappa(\boldsymbol{\theta},\boldsymbol{\theta}')\, \kappa\left(\boldsymbol{\theta}',\Theta^{(l)}\right)\right] \right\} \mathcal{K}^{-1}\mathcal{Y} \\
&\quad - \mathcal{Y}^\top \mathcal{K}^{-1} \left\{ \sum_{k,l} \sum_{s,t} \mathcal{K}^{-1(l,s)} \Pi_p \Pi_p' \left[\kappa\left(\Theta^{(k)},\boldsymbol{\theta}\right) \kappa\left(\boldsymbol{\theta},\Theta^{(l)}\right) \kappa\left(\Theta^{(s)},\boldsymbol{\theta}'\right) \kappa\left(\boldsymbol{\theta}',\Theta^{(t)}\right)\right] \right\} \mathcal{K}^{-1}\mathcal{Y} \\
&= \left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \left| (2\boldsymbol{\Sigma} + 4\mathbf{I}_d)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1} + \mathbf{I}_d \right|^{-1/2} \\
&\quad \mathcal{Y}^\top \mathcal{K}^{-1} \left\{ \begin{array}{c} \kappa\left(\Theta,\Theta\,|3\boldsymbol{\Sigma}\right) \times \kappa\left(\frac{4\Theta}{3}, -\frac{2\Theta}{3}\,\middle|2\boldsymbol{\Sigma} + 4\mathbf{I}_d + \frac{2}{3}\boldsymbol{\Sigma}\right) \\ \times\kappa \left(\begin{array}{c} \left[\frac{\Theta}{3}\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1} - \Theta\left(2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)^{-1}\right] \left[(2\boldsymbol{\Sigma} + 4\mathbf{I}_d)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1}, \\ -\frac{2\Theta}{3}\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1} \left[(2\boldsymbol{\Sigma} + 4\mathbf{I}_d)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1} \\ \left|\left[(2\boldsymbol{\Sigma} + 4\mathbf{I}_d)^{-1} + \left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1}\right]^{-1} + \mathbf{I}_d \end{array} \right) \end{array} \right\} \mathcal{K}^{-1}\mathcal{Y} \\
&\quad - \left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1} \mathcal{Y}^\top \mathcal{K}^{-1} \left[\kappa\left(\Theta,\Theta\,|2\boldsymbol{\Sigma}\right) \times \kappa\left(\Theta,-\Theta\,|2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)\right] \mathcal{K}^{-1} \left[\kappa\left(\Theta,\Theta\,|2\boldsymbol{\Sigma}\right) \times \kappa\left(\Theta,-\Theta\,|2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right)\right] \mathcal{K}^{-1}\mathcal{Y}
\end{aligned}
\tag{B.4}
$$

Appendix C. Analytical results for generalized Posterior Variance Contribution function

Similarly with Appendix Appendix B, the $\rho(\boldsymbol{\theta})$ involved in generalized PVC can be deduced as

$$
\begin{aligned}
\rho(\boldsymbol{\theta}) &= \Pi_p' \left[\mu_{g,M}(\boldsymbol{\theta})\, c_{g,M}(\boldsymbol{\theta},\boldsymbol{\theta}')\, \mu_{g,M}(\boldsymbol{\theta}')\right] \\
&= \mathcal{Y}^\top \mathcal{K}^{-1} \sum_{k,l} \Pi_p' \left[\kappa\left(\Theta^{(k)},\boldsymbol{\theta}\right) \kappa(\boldsymbol{\theta},\boldsymbol{\theta}')\, \kappa\left(\boldsymbol{\theta}',\Theta^{(l)}\right)\right] \mathcal{K}^{-1}\mathcal{Y} \\
&\quad - \mathcal{Y}^\top \mathcal{K}^{-1} \sum_{k,l} \sum_{s,t} \mathcal{K}^{-1(s,t)} \kappa\left(\Theta^{(k)},\boldsymbol{\theta}\right) \kappa\left(\boldsymbol{\theta},\Theta^{(l)}\right) \Pi_p' \left[\kappa\left(\Theta^{(s)},\boldsymbol{\theta}'\right) \kappa\left(\boldsymbol{\theta}',\Theta^{(t)}\right)\right] \mathcal{K}^{-1}\mathcal{Y} \\
&= \left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \mathcal{Y}^\top \mathcal{K}^{-1} \left[\kappa\left(\boldsymbol{\theta},-\Theta\,|2\boldsymbol{\Sigma} + 4\mathbf{I}_d\right) \times \kappa\left(\boldsymbol{\theta} - \frac{2\Theta}{3}, \frac{\Theta}{3}\,\middle|\frac{2}{3}\boldsymbol{\Sigma}\right) \times \kappa\left(\Theta,\Theta\,|3\boldsymbol{\Sigma}\right)\right] \mathcal{K}^{-1}\mathcal{Y} \\
&\quad - \left|2\boldsymbol{\Sigma}^{-1} + \mathbf{I}_d\right|^{-1/2} \mathcal{Y}^\top \mathcal{K}^{-1} \left[\kappa(\Theta,\boldsymbol{\theta}) \times \kappa(\boldsymbol{\theta},\Theta)\right] \mathcal{K}^{-1} \left[\kappa\left(\Theta,\Theta\,|2\boldsymbol{\Sigma}\right) \times \kappa\left(\Theta,-\Theta\,|2\Sigma + 4\mathbf{I}_d\right)\right] \mathcal{K}^{-1}\mathcal{Y}
\end{aligned}
\tag{C.1}
$$

Appendix D. Analytical results for posterior marginal density

Similarly with Appendix Appendix B and Appendix C, the marginal PDF of model parameter $\boldsymbol{\theta}$ can be inferred as

$$
\begin{aligned}
p^{\text{posterior}}(\boldsymbol{\theta}) &= \Pi_{-i}\left[p\left(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}}\right)\right] \\
&= Z^{-1}\Pi_{p_{-i}}\left[p\left(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}\right)\right]p\left(\theta_i\right) \\
&= Z^{-1}\left\{\alpha + 0.5\mathcal{Y}^{\top}\mathcal{K}^{-1}\Pi_{p_{-i}}\left[\kappa\left(\Theta,\boldsymbol{\theta}\right)\kappa\left(\boldsymbol{\theta},\Theta\right)\right]\mathcal{K}^{-1}\mathcal{Y}\right\}p\left(\theta_i\right) \\
&= \alpha + 0.5\left|2\boldsymbol{\Sigma}_{-i}^{-1} + \mathrm{I}_{-iM}\right|^{-1/2}\mathcal{Y}^{\top}\mathcal{K}^{-1}\left[\begin{array}{c}\kappa\left(\Theta,\Theta\,|2\Sigma\right)\times\kappa\left(\theta_i - \frac{\Theta_i}{2},\frac{\Theta_i}{2}\,\middle|\,\frac{1}{2}\Sigma_i\right) \\ \times\kappa\left(\Theta_{-i},-\Theta_{-i}\,|2\Sigma_{-i} + 4\mathrm{I}_{-iM}\right)\end{array}\right]\mathcal{Y}^{\top}\mathcal{K}^{-1}
\end{aligned}
\tag{D.1}
$$

References

[1] J. E. Mottershead, M. Link, M. I. Friswell, The sensitivity method in finite element model updating: A tutorial, Mechanical Systems and Signal Processing 25 (7) (2011) 2275–2296.

[2] M. Faes, M. Broggi, E. Patelli, Y. Govers, J. Mottershead, M. Beer, D. Moens, A multivariate interval approach for inverse uncertainty quantification with limited experimental data, Mechanical Systems and Signal Processing 118 (2019) 534–548.

[3] M. C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (3) (2001) 425–464.

[4] X. Wu, T. Kozlowski, H. Meidani, K. Shirvan, Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, part 1: Theory, Nuclear Engineering and Design 335 (2018) 339–355.

[5] X. Wu, Z. Xie, F. Alsafadi, T. Kozlowski, A comprehensive survey of inverse uncertainty quantification of physical model parameters in nuclear system thermal–hydraulics codes, Nuclear Engineering and Design 384 (2021) 111460.

[6] D. Bingham, T. Butler, D. Estep, Inverse problems for physics-based process models, Annual Review of Statistics and Its Application 11 (2024).

[7] S. Bi, M. Beer, S. Cogan, J. Mottershead, Stochastic model updating with uncertainty quantification: an overview and tutorial, Mechanical Systems and Signal Processing 204 (2023) 110784.

[8] A. Lye, A. Cicirello, E. Patelli, Sampling methods for solving Bayesian model updating problems: A tutorial, Mechanical Systems and Signal Processing 159 (2021) 107760.

[9] B. Carrera, I. Papaioannou, Covariance-based MCMC for high-dimensional Bayesian updating with sequential Monte Carlo, Probabilistic Engineering Mechanics 77 (2024) 103667.

[10] L. Tierney, Markov chains for exploring posterior distributions, The Annals of Statistics (1994) 1701–1728.

[11] J. Ching, Y.-C. Chen, Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging, Journal of Engineering Mechanics 133 (7) (2007) 816–832.

[12] W. Betz, I. Papaioannou, D. Straub, Transitional Markov chain Monte Carlo: observations and improvements, Journal of Engineering Mechanics 142 (5) (2016) 04016016.

[13] T. Radivojević, E. Akhmatskaya, Modified Hamiltonian Monte Carlo for Bayesian inference, Statistics and Computing 30 (2) (2020) 377–404.

[14] J. M. Bardsley, A. Solonen, H. Haario, M. Laine, Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems, SIAM Journal on Scientific Computing 36 (4) (2014) A1895–A1910.

[15] H. Fang, N. Tian, Y. Wang, M. Zhou, M. A. Haile, Nonlinear bayesian estimation: From Kalman filtering to a broader horizon, IEEE/CAA Journal of Automatica Sinica 5 (2) (2018) 401–417.

[16] J. Song, P. Wei, M. Valdebenito, M. Beer, Adaptive reliability analysis for rare events evaluation with global imprecise line sampling, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113344.

[17] I. Yoshida, T. Nakamura, S.-K. Au, Bayesian updating of model parameters using adaptive Gaussian process regression and particle filter, Structural Safety 102 (2023) 102328.

[18] D. Straub, I. Papaioannou, Bayesian updating with structural reliability methods, Journal of Engineering Mechanics 141 (3) (2015) 04014134.

[19] M. Kitahara, C. Dang, M. Beer, Bayesian updating with two-step parallel Bayesian optimization and quadrature, Computer Methods in Applied Mechanics and Engineering 403 (2023) 115735.

[20] C. Song, Z. Wang, A. Shafieezadeh, R. Xiao, BUAK-AIS: Efficient Bayesian updating with active learning Kriging-based adaptive importance sampling, Computer Methods in Applied Mechanics and Engineering 391 (2022) 114578.

[21] D. G. Giovanis, I. Papaioannou, D. Straub, V. Papadopoulos, Bayesian updating with subset simulation using artificial neural networks, Computer Methods in Applied Mechanics and Engineering 319 (2017) 124–145.
[22] A. O'Hagan, Bayes–hermite quadrature, Journal of Statistical Planning and Inference 29 (3) (1991) 245–260.
[23] M. Osborne, R. Garnett, Z. Ghahramani, D. K. Duvenaud, S. J. Roberts, C. Rasmussen, Active learning of model evidence using Bayesian quadrature, Advances in neural information processing systems 25 (2012).
[24] P. Hennig, M. A. Osborne, H. P. Kersting, Probabilistic Numerics: Computation as Machine Learning, Cambridge University Press, 2022.
[25] R. Baptista, Y. Marzouk, O. Zahm, On the representation and learning of monotone triangular transport maps, Foundations of Computational Mathematics (2023) 1–46.
[26] T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, S. J. Roberts, Sampling for inference in probabilistic models with fast bayesian quadrature, Advances in neural information processing systems 27 (2014).
[27] M. Kanagawa, P. Hennig, D. Sejdinovic, B. K. Sriperumbudur, Gaussian processes and kernel methods: A review on connections and equivalences, arXiv preprint arXiv:1807.02582 (2018).
[28] C. E. Rasmussen, Z. Ghahramani, Bayesian Monte Carlo, Advances in neural information processing systems (2003) 505–512.
[29] P. Wei, X. Zhang, M. Beer, Adaptive experiment design for probabilistic integration, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113035.
[30] F. Huszár, D. Duvenaud, Optimally-weighted herding is bayesian quadrature, International Conference on Uncertainty in Artificial Intelligence (UAI) (2012).
[31] M. Kanagawa, P. Hennig, Convergence guarantees for adaptive Bayesian quadrature methods, Advances in neural information processing systems 32 (2019).
[32] F.-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, D. Sejdinovic, et al., Probabilistic integration: A role in statistical computation?, Statistical Science 34 (1) (2019) 1–22.
[33] T. Karvonen, S. Särkkä, Classical quadrature rules via Gaussian processes, in: 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2017, pp. 1–6.
[34] M. Kitahara, T. Kitahara, Sequential and adaptive probabilistic integration for bayesian model updating, Mechanical Systems and Signal Processing 223 (2025) 111825.
[35] J. Song, Z. Liang, P. Wei, M. Beer, Sampling-based adaptive Bayesian quadrature for probabilistic model updating, Computer Methods in Applied Mechanics and Engineering (2025) 117467.
[36] F. Hong, P. Wei, S. Bi, M. Beer, Efficient variational bayesian model updating by bayesian active learning, Mechanical Systems and Signal Processing 224 (2025) 112113.
[37] M. Adachi, S. Hayakawa, M. Jørgensen, H. Oberhauser, M. A. Osborne, Fast Bayesian inference with batch Bayesian quadrature via kernel recombination, Advances in Neural Information Processing Systems 35 (2022) 16533–16547.
[38] L. Acerbi, An exploration of acquisition and mean functions in variational Bayesian Monte Carlo, in: Symposium on Advances in Approximate Bayesian Inference, PMLR, 2019, pp. 1–10.
[39] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811 (2018).
[40] J. Song, Y. Cui, P. Wei, M. A. Valdebenito, W. Zhang, Constrained Bayesian optimization algorithms for estimating design points in structural reliability analysis, Reliability Engineering & System Safety 241 (2024) 109613.
[41] P. Wei, Y. Zheng, J. Fu, Y. Xu, W. Gao, An expected integrated error reduction function for accelerating Bayesian active learning of failure probability, Reliability Engineering & System Safety 231 (2023) 108971.
[42] M. Rosenblatt, Remarks on a multivariate transformation, The Annals of Mathematical Statistics 23 (3) (1952) 470–472.
[43] J. L. Beck, S.-K. Au, Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation, Journal of Engineering Mechanics 128 (4) (2002) 380–391.
[44] B. Li, J. Sun, S. Zhu, Y. Fu, X. Zhao, H. Wang, Q. Teng, Y. Ren, Y. Li, G. Zhu, Thermohydrodynamic lubrication analysis of misaligned journal bearing considering the axial movement of journal, Tribology International 135 (2019) 397–407.
[45] J. Song, P. Wei, M. A. Valdebenito, M. Faes, M. Beer, Data-driven and active learning of variance-based sensitivity indices with Bayesian probabilistic integration, Mechanical Systems and Signal Processing 163 (2022) 108106.