OPEN ACCESS
IOP Publishing | ISECAEP
Journal of Reliability Science and Engineering
J. Reliab. Sci. Eng. **1** (2025) 015003 (22pp)
https://doi.org/10.1088/3050-2454/ad9f62

Probabilistic calibration of model parameters with approximate Bayesian quadrature and active machine learning

Pengfei Wei[1,2,*], **Masaru Kitahara**[3], **Matthias G R Faes**[4,7] **and Michael Beer**[5,6,7]

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

E-mail: pengfeiwei@nwpu.edu.cn

Received 21 October 2024, revised 19 November 2024
Accepted for publication 9 December 2024
Published 22 January 2025

Abstract

The calibration of computational models using experimental or operational data to achieve accurate predictions is widely recognized as a crucial challenge in reliability engineering. Bayesian model updating (BMU) has been developed as an appealing methodological framework to achieve this goal, but existing methods range from very approximate but cheap (e.g. Laplace approximation and conjugate priors), less approximate and a bit cheaper (e.g. approximate Bayesian computation), to quite expensive and highly informative techniques such as full Bayesian computation. The goal of this work is to achieve full Bayesian accuracy at a low cost. The approximate Bayesian quadrature has emerged as 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 the desired level of accuracy. The proposed method leverages information revealed by both the mean predictions and the posterior covariance of the probabilistic regression model trained for approximating the likelihood function. It thus provides a better trade-off between exploration and exploitation. Results from both numerical and engineering examples show that the proposed method is applicable to multimodel problems, achieving high accuracy and efficiency.

* Author to whom any correspondence should be addressed.

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 beome standard practices in reliability engineering due to the increasing complexity of products and the high cost or even unavailability of implementing physical reliability tests. To this end, the credibility of physical models has been of special concern for the theoretical development of reliability methods driven by the physics of failure. However, due to the complexity of service environments and failure mechanisms, it is difficult or even impossible to capture all the physical details using mathematical models. Furthermore, numerical solution of the mathematical models may introduce extra prediction uncertainty resulting from the numerical computation. For the above reasons, the established computational simulators, although requiring significant computational effort, may provide untrustworthy predictions. This has motivated the development of methods to calibrate computational simulators, with the proper fusion of measurements generated from well-designed physical tests and the real-life operation of the structure under consideration.

Several model calibration methodological frameworks have been developed, which can generally be divided into two groups, i.e. deterministic and non-deterministic. Deterministic methods aim at calibrating the model parameters by minimizing the cost function and are realized by using effective sensitivity information computed by, e.g. adjoint methods [1]. Deterministic methods seek to find a point estimate of the parameters which best interprets the measurements; however, they are commonly criticized for their inability to treat multimodal inference problems and their instability when processing problems with measurement noise. On the other hand, non-deterministic frameworks aim to model alternative types of root uncertainties, resulting in the representation of a simulator's prediction uncertainty using probabilistic or non-probabilistic models [2]. This framework has received tremendous attention in the past two decades since the milestone work in [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 resulting probabilistic calibration problem is termed 'stochastic model updating' and the numerical solution to the problem can be more challenging [6, 7] as it involves several loops of numerical computation. In this study, we follow the approach commonly used in most probabilistic model updating literature [8, 9] and treat the model parameters as deterministic-but-unknown variables with their epistemic uncertainty described by a subjective probability model.

Methods for probabilistic parameter calibration, also called Bayesian model updating (BMU), have been studied extensively over 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] and the covariance-based MCMC [9], were developed to improve performance for specific types of challenging BMU problems. Generally, MCMC sampling schemes are computationally demanding for expensive-to-evaluate simulators as each state requires at least one call of the simulator, and can be less effective for multimodal 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 were originally developed for probabilistic prediction and updating of the state of a physical system, and rely on state-space models. This group of methods includes Kalman filters (including its advanced versions), particle filters, quadrature filters and functional Bayesian filter. One can refer to [15] for a review of selected Bayesian filter algorithms. These algorithms have also been preliminarily extended to the 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 are flexible as they can sequentially fuse the measurements batch by batch, but the high computational cost and lower suitability for multimodal problems needs to be alleviated.

The third group of methods is usually named Bayesian updating with structural reliability (BUS) and originate from [18]. The basic idea is to reformulate the BMU problem as a structural reliability analysis problem, thus enabling it to be solved using extensive well-developed structural reliability methods, such as sampling (e.g. subset simulation and line sampling) and moment-based methods (first-order and second-order reliability methods). It should be noted that the equivalent reliability problems usually have highly nonlinear or even ill-behaved features, and the associated probability of failure can be extremely small given the many observations. Moreover, one factor, denoted as α, needs to be determined as a prior for reliability analysis, which is non-trivial [19]. To alleviate extra challenges, many active learning schemes

based on neural networks or Gaussian process (GP) regression have been combined with the BUS scheme, see e.g. [19–21]. Despite the above improvements, the performance of tackling multimodal problems still needs to be improved.

The last group, which has received attention in the community of machine learning, is called Bayesian quadrature, which was originally developed for estimating multidimensional integral equations [22]. This scheme has been extended to estimate the 'evidence' term, defined by the integral of the likelihood over prior density and then also the posterior density with the resultant surrogate. Two key issues need to be addressed. The first concerns the approximation of the likelihood with a probabilistic regression model to capture the prediction uncertainty [23], while the second aims at devising acquisition functions to acquire a faster convergence rate [24]. For the latter issue, most of the current developments are based on modification of the uncertainty sampling (US) acquisition function, which does not make full use of the information of the probabilistic regression model for approximating the likelihood. However, these methods have been shown to be superior for addressing multimodal problems in terms of both accuracy and efficiency, owing to the flexibility and interpretability of the GP 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 do not provide more detail as it is not the focus of this work.

Following the technical road-map of Bayesian quadrature, this work introduces a new family of acquisition functions to offer greater flexibility in balancing exploration and exploitation. This is realized by approximating a proxy function, defined by the square root of the likelihood, with a GP model and then formulating the acquisition function by leveraging both posterior mean and covariance information from the GP model. A closed-form expression of the new acquisition function is also presented, allowing for efficient and effective querying 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 multimodal posterior density, which is challenging but common in BMU.

The remainder of the paper is organized as follows. Section 2 briefly reviews some of the research, including the formulation of the BMU problem and the exact Bayesian quadrature for the active learning of multidimensional integrals. Section 3 presents the main developments in this work, including the approximate Bayesian quadrature scheme, the new acquisition function, the closed-form expressions and the details of the algorithm, followed by benchmark studies in section 4. Section 5 concludes this work.

2. Theoretical foundations

2.1. BMU

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

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

where $\epsilon(\boldsymbol{x}_{\text{obs}}^{(k)})$ 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 Gaussian white noise, i.e. $\epsilon(\boldsymbol{x})=\epsilon\sim\mathcal{N}(0,\sigma_n^2)$, with σ_n^2 being the variance of noise. The goal 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 considering the model bias, denoted as $\delta(\boldsymbol{x})$, both the deterministic-but-unknown parameters and the model bias need to be modeled using probabilistic frameworks. This 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 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 the Bayesian rule then turns out to be a compatible way to treat such kind of paradox. Given the assumption of the Gaussian white noise of ϵ, the likelihood function of the observations $\mathcal{D}_{\text{obs}}$ can be formulated as:

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

where $\phi\left(\cdot|\sigma_n^2\right)$ 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 $p(\boldsymbol{\theta})$, the posterior density of $\boldsymbol{\theta}$ is expressed as:

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

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

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

and is termed as 'model evidence', or simply as 'evidence', as its value measures the strength of evidence of the model

after being calibrated. The numerical analysis task for BMU can then be simply described as estimating the posterior density $p(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}})$, which is generally impossible to solve analytically unless the prior and posterior are a pair of conjugate distributions. Moreover, due to the necessity of simulating multiphysical and/or multiscale processes, the computational cost for each call of the simulator $\mathcal{M}(\boldsymbol{x},\boldsymbol{\theta})$, and then also the likelihood function $p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta})$, can be extremely expensive. Therefore, estimating $p(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}})$ with the desired accuracy and as few simulator calls as possible has long been a ersistent and important topic. Multiple branches of algorithms have been developed for this numerical task, which include sampling methods [8], transport map theory [25], Bayesian quadrature [26] and structural reliability method [18], as reviewed in the introduction. Bayesian quadrature will be used in this work due to its simplicity and potential for addressing our current 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 consider 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 $\boldsymbol{\Theta} = \left[\boldsymbol{\theta}^{(l)}\right]_{l=1}^{M}$ (an $M \times d$-dimensional matrix) and weights $\mathcal{W} = \left[w^{(l)}\right]_{l=1}^{M}$ (an M-dimensional column vector) and is presented as $Z = \sum_{l=1}^{M} y^{(l)} w^{(l)}$, with $y^{(l)} = g(\boldsymbol{\theta}^{(l)})$. However, before the integrand is computed, we may have no prior information on its behavior or we may know some information, such as the smoothness of the integrand. The Bayesian quadrature is then initialized by assuming a prior GP model for g, then inferring a posterior GP model for approximating the integrand conditioned on the quadrature nodes $\mathcal{S} = \{(\boldsymbol{\theta}^{(j)}, y^{(j)})\}_{j=1}^{M}$ 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 resulting from the prediction uncertainty of the GP model. Thus, we first briefly review the content of the 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 their 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 reproducing kernel Hilbert space (RKHS) and is denoted as $\mathbb{H}$. In this work, we use the squared exponential kernel defined as:

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

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

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

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

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

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

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

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

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

$$\begin{aligned}\sigma_{Z,M}^2 = \Pi_p\Pi_p'\left[c_{g,M}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right] &= \Pi_p\Pi_p'\left[\kappa\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right)\right] \\ &- \Pi_p[\kappa(\boldsymbol{\theta},\boldsymbol{\Theta})]\mathcal{K}^{-1}\Pi_p'\left[\kappa\left(\boldsymbol{\Theta},\boldsymbol{\theta}'\right)\right],\end{aligned} \tag{9b}$$

where $\Pi_p'[\cdot] = \int_{\mathbb{T}} \cdot p\left(\boldsymbol{\theta}'\right)\mathrm{d}\boldsymbol{\theta}'$ and $\Pi_p\Pi_p'[\cdot] = \int_{\mathbb{T}}\int_{\mathbb{T}} \cdot p\left(\boldsymbol{\theta}'\right) p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}'\mathrm{d}\boldsymbol{\theta}$. Similarly, the posterior mean $\mu_{Z,M}$ provides a mean prediction of the integral Z, while the posterior variance $\sigma_{Z,M}^2$ 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 exactly equals the square of the worst-case error [30], i.e. $\sigma^2_{Z,M} = (\sup_{\|g\|_{\mathbb{H}}\leqslant 1} |\Pi_p[g] - \mu_{Z,M}|)^2$, 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 proven for both random and adaptive design of quadrature points [31, 32]. Due to the richness of the RKHS, these theoretical results present sufficient confidence to extract the potential of using Bayesian quadrature for solving BMU problems and beyond, by simply addressing the numerical error with the posterior variance $\sigma^2_{Z,M}$.

Furthermore, by the assumption of a zero prior mean $m(\boldsymbol{\theta}) \equiv 0$, the posterior mean quadrature rule of equation (9*a*) can be written in a classical quadrature form as $\mu_{Z,M} = \sum_{l=1}^{M} w^{(l)} y^{(l)}$ with the weight vector being computed by $\mathcal{W}^{\top} = \Pi_p[\kappa(\boldsymbol{\theta}, \boldsymbol{\Theta})]\mathcal{K}^{-1}$. 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. Another 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, so-called active learning, is usually driven by an acquisition function which measures the reward of a design. Two popular acquisition functions are the US function [24] and the posterior variance contribution (PVC) function [29] which are formulated as:

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

and

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

respectively. The US function is indeed the prediction uncertainty (weighted by the squared density) of the GP model. Since 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 $\int_{\mathbb{T}} \mathcal{A}_{\mathrm{PVC}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} = \sigma^2_{Z,M}$. 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 the highest amount. As PVC leverages the posterior covariance information of the GP model [29], it is shown to be more effective than the US in most cases.

Due to the multimodal behavior of the US and PVC functions, heuristic algorithms such as particle swarm optimization and genetic algorithms are commonly suggested for searching their global peak points, which brings the requirement that the acquisition function can be computed with trivial cost and admits closed-form expressions. For the 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 the squared exponential kernel and the Gaussian form of $p(\boldsymbol{\theta})$. Assuming that $p(\boldsymbol{\theta})$ is of standard Gaussian form with zero mean and unit variance, the closed-form expressions for the 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 reported in appendix A (see also [22, 28, 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 Bayesian 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, thereby enabling higher flexibility in balancing the exploration and exploitation of the algorithm.

3. The proposed method

It has been widely recognized that, see e.g. [23, 26, 31], the exact Bayesian quadrature rule reviewed in the last section may cause large numerical bias if it is directly used for estimating the evidence in equation (4). This is due to the fact that the integrand in equation (4), which is exactly the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$, 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. Several approximate tricks have been developed to accommodate 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 require an approximate description of the 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 [26] and thus in what follows we present an introduction to this trick and also some further mathematical developments for extending it.

3.1. Approximate Bayesian quadrature: basic rationale

Instead of approximating the likelihood $p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta})$ with a GP, [26] proposed to approximate the proxy function defined as the square root $g(\boldsymbol{\theta}) = \sqrt{2(p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}) - \alpha)}$ with a GP such that $p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}) = \alpha + 0.5g^2(\boldsymbol{\theta})$, where α is a small positive scalar, suggested to evolve with the training sample as $\alpha = 0.8\min_{\boldsymbol{\theta}^{(l)}\in\boldsymbol{\Theta}} p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}^{(l)})$. 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 $g(\boldsymbol{\theta})$ with the GP $\hat{g}_M(\boldsymbol{\theta}) \sim \mathcal{GP}\left(\mu_{g,M}(\boldsymbol{\theta}), c_{g,M}(\boldsymbol{\theta},\boldsymbol{\theta}')\right)$, the induced stochastic process $\hat{p}_M(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta}) = \alpha + 0.5\hat{g}_M^2(\boldsymbol{\theta})$ 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]. In this work, the linearization trick is utilized and introduced below.

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

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

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

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

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

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 equations (9*a*) and (9*b*), the posterior mean and variance of the model evidence Z can be obtained as:

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

and

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

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

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

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

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

In equations (15) and (16), the notation $\kappa(\cdot,\cdot|\boldsymbol{\Sigma}^*)$ 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:

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

The above closed-form expressions, adapted from the supplementary material of [37], are applicable for the case with a Gaussian prior and a squared exponential kernel. The details of the mathematical deductions of these expressions are also reported in appendix B. Although these closed-form expressions appear to be complex, when used correctly, they make the algorithm numerically efficient and robust. Unfortunately, for most of the other pairs of kernels and prior densities, e.g. the Matérn kernel, the closed-form expressions are intractable and one should resort to numerical methods, e.g. Monte Carlo simulation, for estimating these quantities.

3.2. Generalized PVC (GPVC) function for active learning

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

$$\mathcal{A}_{\text{PVC}}^{\mathcal{L}}(\boldsymbol{\theta}) = \rho_M(\boldsymbol{\theta})\, p(\boldsymbol{\theta}), \tag{18}$$

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

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

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

the highest reduction in 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:

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

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 equation (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 the US (GUS) function presented in [38], a family of 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 in balancing the exploration and exploitation significantly influences the performance of the convergence speed, where 'exploration' means the ability to explore 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 equation (21), the values of α and γ provide flexibility for balancing exploration and exploitation. Given fixed values of β and γ, increasing α will result in more focus on exploration, which is desired when the number of quadrature points becomes larger. Thus, it is also suggested to specify α as iteration-dependent. Specifically, given $\beta,\gamma=1$, α can be 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 defined by equation (18). As a case study, the effects of the options on the three parameters α, β and γ are investigated in this work. In these investigations, the particle swarm algorithm is utilized to search for global optima of the acquisition functions. Tto ensure numerical stability, the target function is set as the logarithm of the acquisition functions, in case the values of an acquisition function cover several orders of magnitude. This means searching 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, $\ln p(\boldsymbol{\theta})\propto-\boldsymbol{\theta}^\top\boldsymbol{\theta}/2$. Then, by substituting equations (14) and (20) into equation (22), one can obtain the closed-form expression of $\ln\mathcal{A}^{\mathcal{L}}_{\mathrm{GPVC}}(\boldsymbol{\theta})$, allowing for efficient 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 to transfer the formulation of the evidence as an integral over the 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 $\boldsymbol{\theta}_{1:i}=(\theta_1,\theta_2,\ldots,\theta_i)^\top$. Let $\Phi(\vartheta_i)$ be the univariate CDF of a standard Gaussian variable ϑ_i. The ith 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 $\boldsymbol{\theta}_{1:i-1}$. Owing to the reversibility, the inverse Rosenblatt transformation can be formulated as:

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

Substituting equation (24) into equation (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 equation (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:

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

To summarize, first we need 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 equation (25) using the standard Bayesian quadrature scheme and then apply the forward Rosenblatt transformation using equation (27) to obtain a probabilistic estimate for the targeted posterior density $p(\boldsymbol{\theta}|\mathcal{D}_{\mathrm{obs}})$.

3.3. Summary of the algorithm

Based on the developments above, the pseudocode of the adaptive Bayesian quadrature with GPVC as an acquisition function is then summarized in algorithm 1. The inputs to the algorithm include the prior density $p(\boldsymbol{\theta})$, the likelihood function $p(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta})$, the initial training sample size M_0 and the stopping threshold ϵ. M_0 can be set as 6–15, 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%–5%. The outputs of

Algorithm 1. Adaptive Bayesian quadrature with GPVC as an acquisition function.

Input: $p(\boldsymbol{\theta})$, $p(\mathcal{D}_{\text{obs}}|\boldsymbol{\theta})$, M_0, ϵ.
Output: $\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})$, $c^{\mathcal{L}}_{p,M}(\boldsymbol{\theta},\boldsymbol{\theta})$, $\mu_{Z,M}$, $\sigma^2_{Z,M}$, 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 $\hat{g}_M(\boldsymbol{\theta})$ based on $\mathcal{S}$;
  Induce the posteriro mean $\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})$ and covariance $c^{\mathcal{L}}_{p,M}(\boldsymbol{\theta},\boldsymbol{\theta}')$ of the GP model $\hat{p}^{\mathcal{L}}_M(\mathcal{D} \mid \boldsymbol{\theta})$ with equations (13*a*) and (13*b*) respectively;
  Compute posterior mean $\mu_{Z,M}$ and variance $\sigma^2_{Z,M}$ of evidence based on equations (14) and (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 $\mathcal{A}^{\mathcal{L}}_{\text{GPVC}}(\boldsymbol{\theta})$ by equation (21);
    Calculate the next training sample $\boldsymbol{\theta}^+$ by maximizing equation (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

the algorithm then include a posterior Gaussian distribution with mean $\mu_{Z,M}$ and variance $\sigma^2_{Z,M}$ for the evidence term Z, a posterior GP model $\hat{p}(\boldsymbol{\theta}|\mathcal{D}_{\text{obs}})$ with explicit posterior mean $\mu^{\mathcal{L}}_{p,M}(\boldsymbol{\theta})$ and variance $c^{\mathcal{L}}_{p,M}(\boldsymbol{\theta},\boldsymbol{\theta})$, 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 multiple times (e.g. two times) in succession. This will help to avoid false convergence when may appear in the early training stage.

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. Case studies

In this section, we use one numerical example without physical meaning 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. For fairness, 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. A higher value of α tends to place more emphasis on exploration, while a large value of γ leads to more emphasis on exploitation and a large value of β leads to a preference for exploring areas with large values of prior density. At the current stage, it is still impossible for us to make theoretical and quantitative conclusions about 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 to a value between 0.5 and 2; and γ should be less than 2. Some of the five settings in table 1 are borrowed from good practice of using the GUS acquisition functions (see e.g. [26, 31, 38]). Moreover, to avoid fake convergence, the algorithm is only stopped when the stopping condition $\text{COV}_{Z,M} < \Delta^{\text{COV}}$ is satisfied for two successive iterations, where the threshold Δ^{COV} is set to be 2% for all cases.

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, $x_{\text{obs}} = (-0.5, 1, 0, 0.6, 0.5)$, measurement is conducted and the corresponding values are $y_{\text{obs}} = (0.0077, 0.5654, -0.4518, 0.2921, 0.3593)$. The measurement noise is assumed to be white noise with the standard deviation (STD) being 0.2. Thus, the likelihood function is defined as $p(\mathcal{D}_{obs}|\boldsymbol{\theta}) = \prod_{k=1}^{5} \phi\left(y^{(k)}_{obs} - \mathcal{M}\left(\boldsymbol{\theta}, x^{(k)}\right)|0.2^2\right)$,

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

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 GUS as acquisition functions for each case, and, for comparison fairness, the algorithm with either acquisition function is implemented repeatedly 10 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 it is 0.0801 with the corresponding COV being 0.0098, and thus sufficiently accurate. It can be seen that, with either GPVC or GUS under any of the five cases, the evidence suggests that 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 demonstrates the high accuracy of the Bayesian quadrature for estimating the evidence. It is also shown that from the last column of table 2, for all five cases, the algorithms require far fewer simulator calls on average. The distribution of the number of simulator calls across the 10 replications is then reported in figure 1. It is obvious that the algorithm with GPVC requires fewer simulator calls on average, and the dispersion is obviously smaller than that required by GUS. To further illustrate 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 shown again that to reach the same level of numerical accuracy, the GPVC function requires a smaller number of model calls and it is also empirically shown that the GPVC function has a higher convergence rate. It is shown again that the proposed GPVC function results in faster convergence than the GUS function.

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

We then discuss the results for the posterior density $p(\theta_1,\theta_2|\mathcal{D})$, which are generated under the 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 setting 3 with GPVC and GUS as acquisition functions are shown in the two lower panels. It is shown that the true posterior density displays a bimodal feature, which is caused by the non-monotonic behavior of the model response with respect to the model parameters. This is on purpose as it demonstrates the ability of the proposed method to capture the multimodal feature. It is shown that, although the posterior density shows multimodal behavior, both GUS and GPVC produce accurate estimates of it, and the samples produced by TMCMC match well with the reference density. The absolute errors and the posterior STDs 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 sufficient, proving the high robustness of the estimates. It is also shown in figure 3 that the probability density covers the true values 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 the multimodal 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 requires approximately 100 simulator calls, which is significantly smaller than that of TMCMC. Thus, these results sufficiently demonstrate the high efficiency and accuracy of the proposed method in estimating both the evidence and the posterior density.

4.2. A two degrees of freedom structural dynamic model

A dynamic model of a two-storied shear building modified from [43] is then adopted and is schematically shown in figure 5. The masses of the first and seconds floor are given as $m_1=16\,531$ kg and $m_2=16\,131$ kg, respectively. The value of stiffness of the interstory is 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 $k=29.7\times10^6\,\mathrm{N\,m^{-1}}$. The problem is then formulated as an update 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 distributions of θ_1 and θ_2 are assumed to be log normal, 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 $\left(f_{\mathrm{obs},1},f_{\mathrm{obs},2}\right)^{\mathrm{T}}=(3.13,9.83)$. By integrating the prior information and measurements, the posterior PDF of parameters $\boldsymbol{\theta}$ is formulated as

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

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

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

088

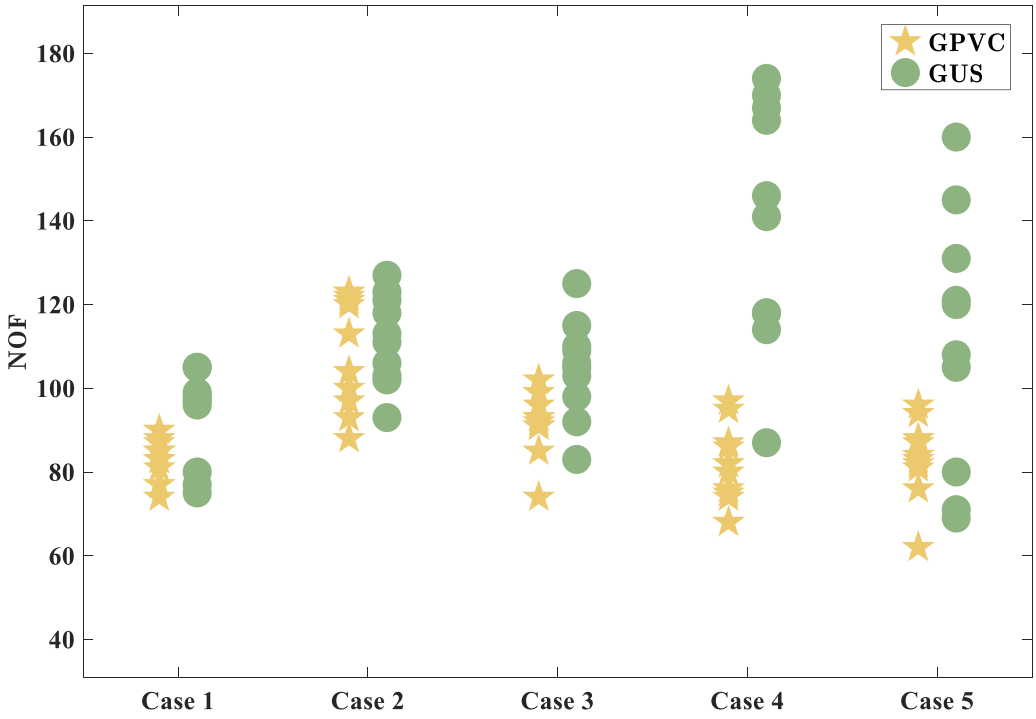


Figure 1. Distribution of the number of simulator calls for all cases of the first example across 10 replications.

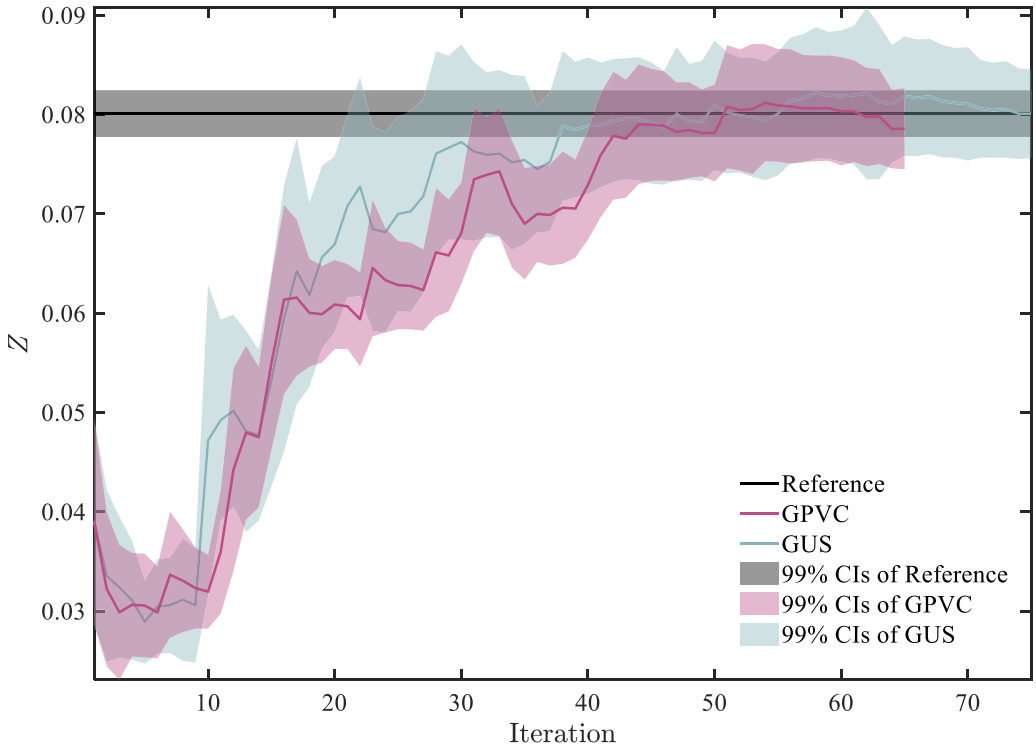


Figure 2. Evolution of posterior confidence intervals of Z with respect to the iteration for case 3 of example 1.

089

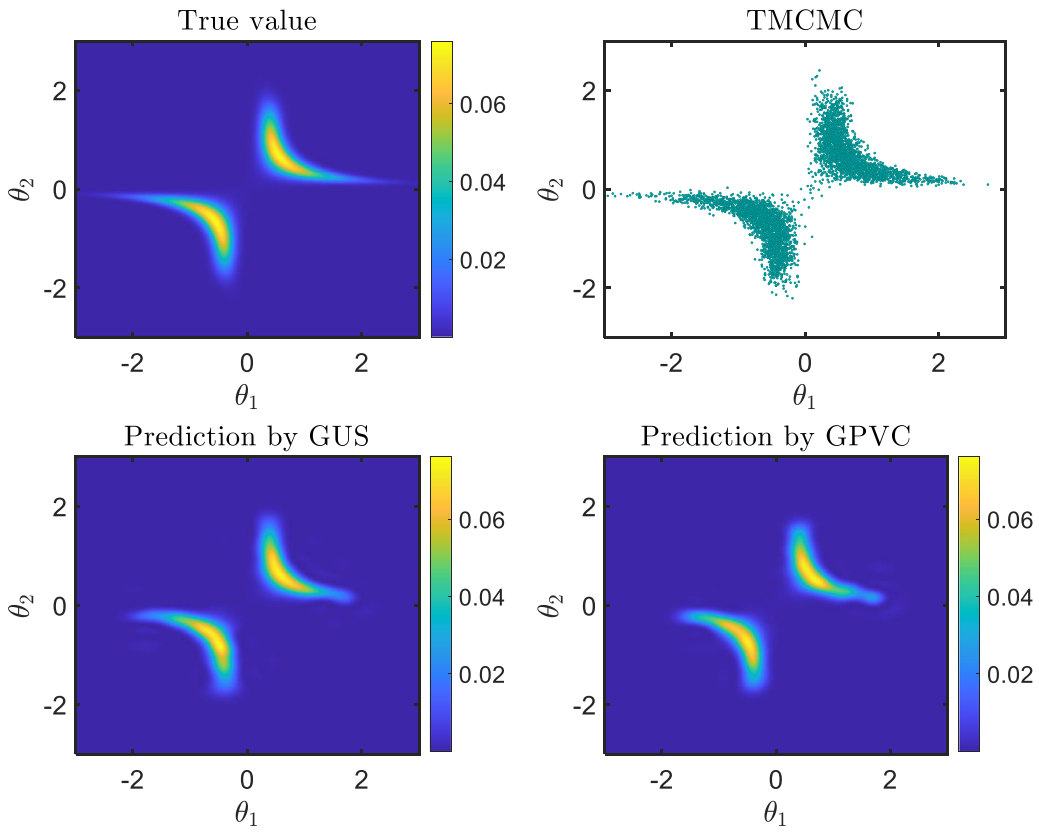


Figure 3. Comparison of results for posterior density of the first example, where PVC and VUS are used under setting of case 3. The corresponding numbers of simulator calls are $N_{\mathrm{PVC}} = 104$ and $N_{\mathrm{VUS}} = 114$.

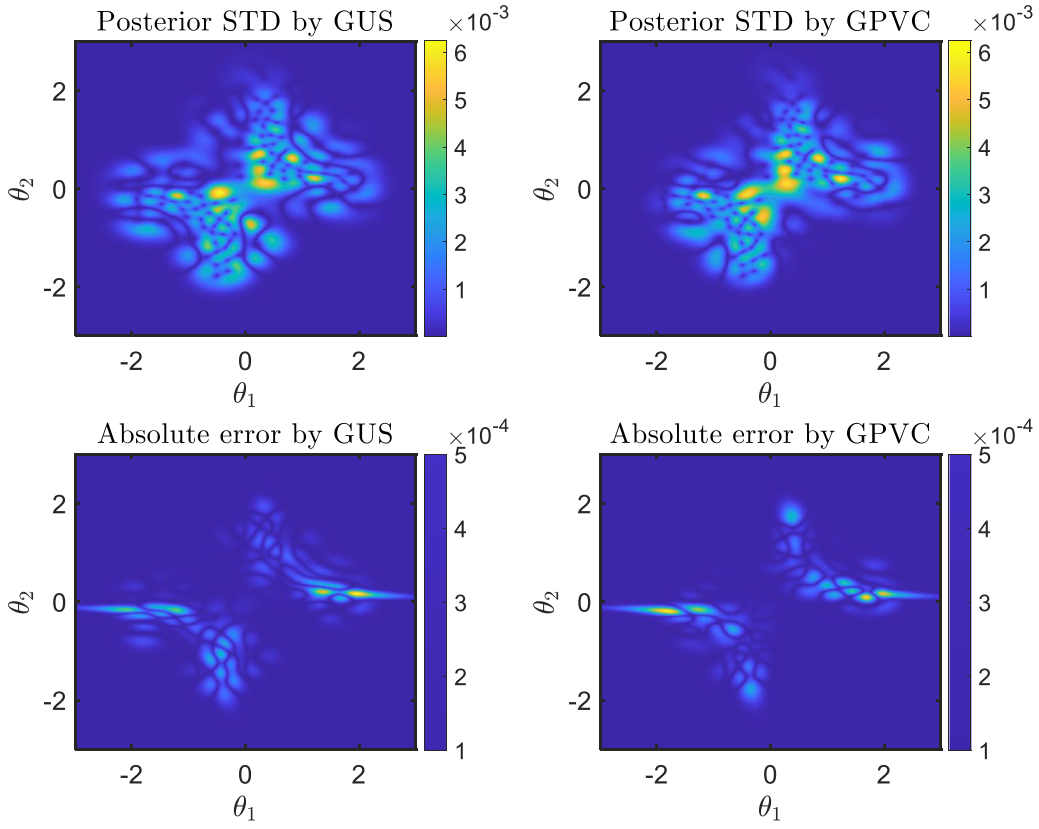


Figure 4. Results of PVC for case 3 ($N_{\mathrm{PVC}} = 104$, $N_{\mathrm{VUS}} = 114$).

090

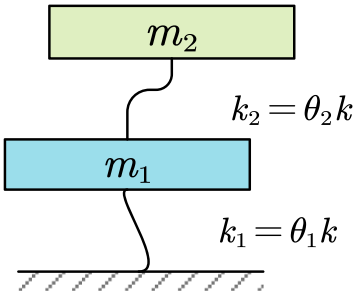


Figure 5. A two-storied shear building model.

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

$$f_{\text{ana},1} = \frac{1}{2\pi}\sqrt{\frac{k_1 m_2 + k_2 m_1 + k_2 m_2 - c}{2m_1 m_2}} \\ f_{\text{ana},2} = \frac{1}{2\pi}\sqrt{\frac{k_1 m_2 + k_2 m_1 + k_2 m_2 + c}{2m_1 m_2}}, \tag{31}$$

where

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

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 five cases of parameter setting. For each setting, the algorithm is repeated 10 times and the results for 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 using MCS with 5×10^4 samples and the corresponding COV is 1.57%. Again, it is found that the algorithm with either GUS or GPVC as acquisition function under any of the five settings produces 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 required for each implementation is reported in figure 6. It is shown that, except for the setting of case 2, the GPVC function requires fewer model calls, and even for case 2, GPVC requires almost the same number of model calls as GUS on average. This sufficiently demonstrates the superiority of GPVC over GUS.

Results for the 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 required by the GPVC and GUS is 45 and 54, respectively. The absolute error and posterior STDs of the estimates by GUS and PVC are displayed in the comparison of GUS and GPVC for case 3 ($N_{\text{PVC}} = 45$, $N_{\text{VUS}} = 54$). The absolute errors as well as the posterior STDs of these results are reported in figure 8. It is shown that, while both GPVC and GUS produce accurate and robust estimates of the posterior density, the GPVC function requires 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. It is shown that, for the two implementations, the GPVC function results in a higher convergence rate than the GUS function. All these results demonstrate the high efficiency and accuracy of the proposed GPVC function for estimation of both the evidence Z and the posterior density.

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

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

Journal bearings are one of the most important components of an aero-engine gear fuel pump as they provide support for the extreme loads excited by the internal flow. Wear failure is one of the main failure modes, and thus lubrication performance is one of the main concerns in the design stage. To predict the lubrication performance under different operating modes, and following [44], we have developed a complex simulator. In this case study, we use a simplified version of this model to illustrate the proposed method for calibrating the parameters of the 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, 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 by a set of partial differential equations, which are the integration of the generalized Reynolds equation, the energy equation [44] and the viscosity–temperature relationship. They 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

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

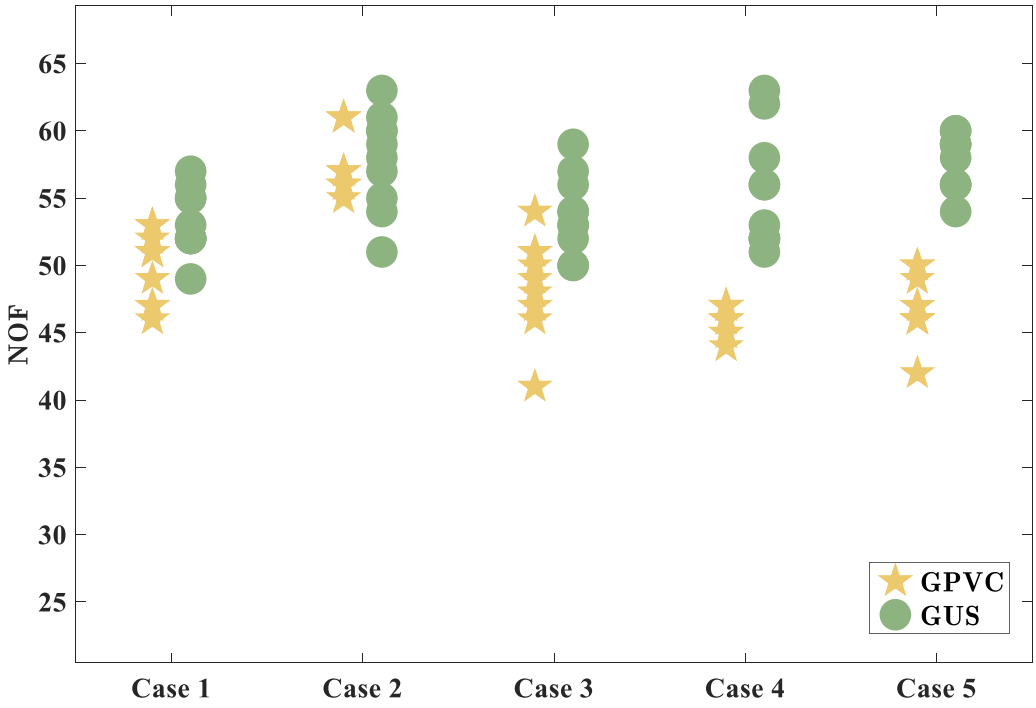


Figure 6. Distribution of the number of model calls for the 2-DOF dynamic model, where for each of the five settings, the GPVC and GUS functions are utilized and implemented for ten times.

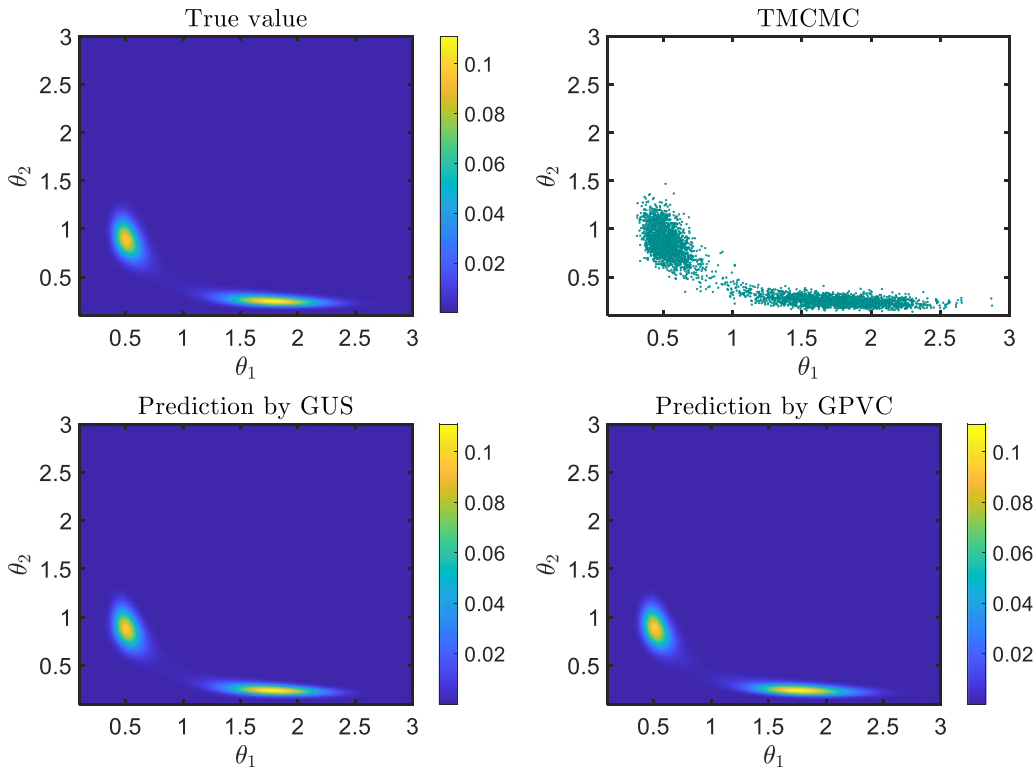


Figure 7. Results of posterior density for the 2-DOF dynamic model.

092

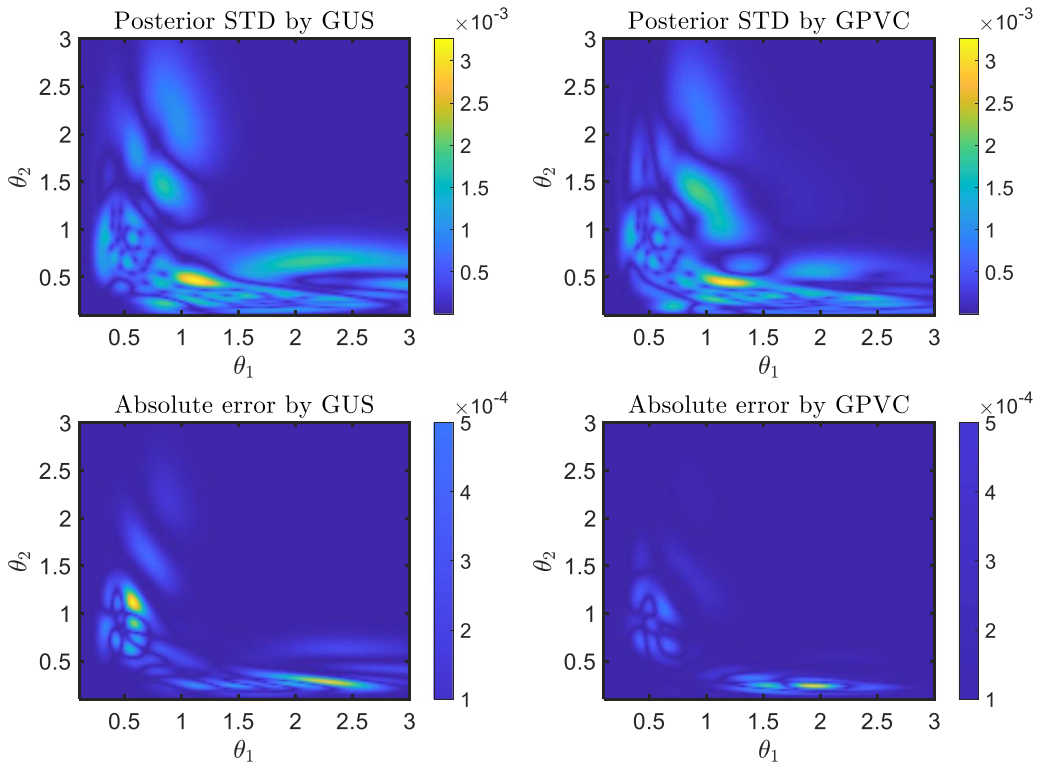


Figure 8. Absolute errors and posterior STDs of the posterior density estimated with approximate Bayesian quadrature.

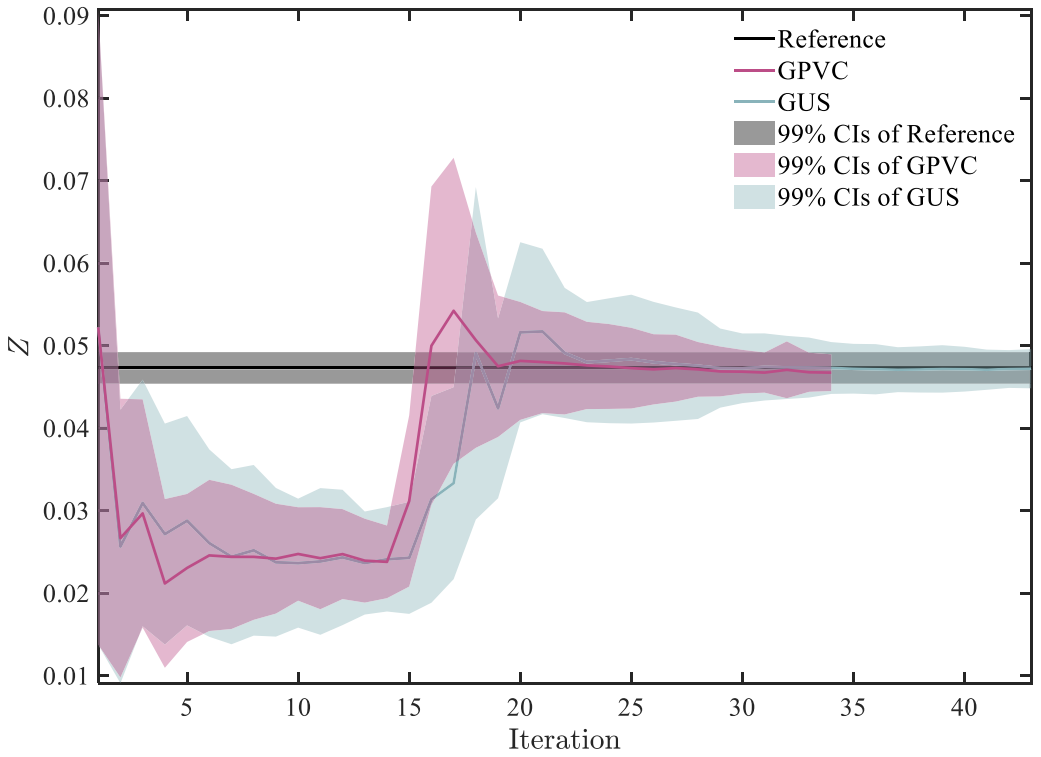
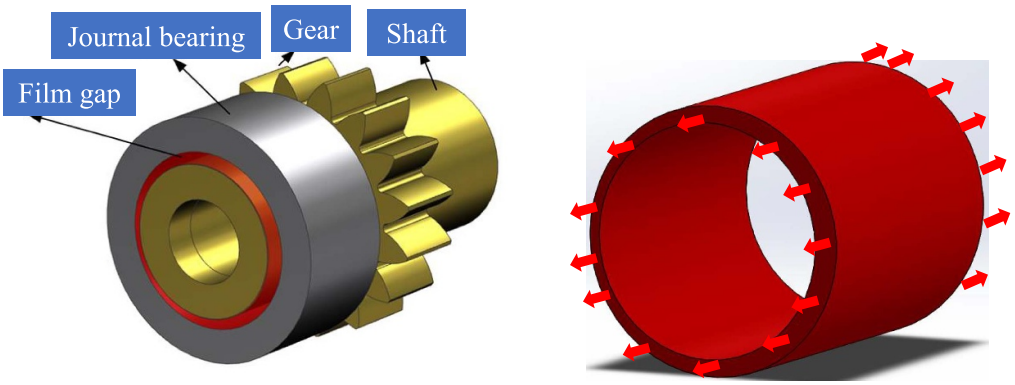


Figure 9. Evolution of evidence with respect to steps of iteration for the 2-DOF dynamic model.

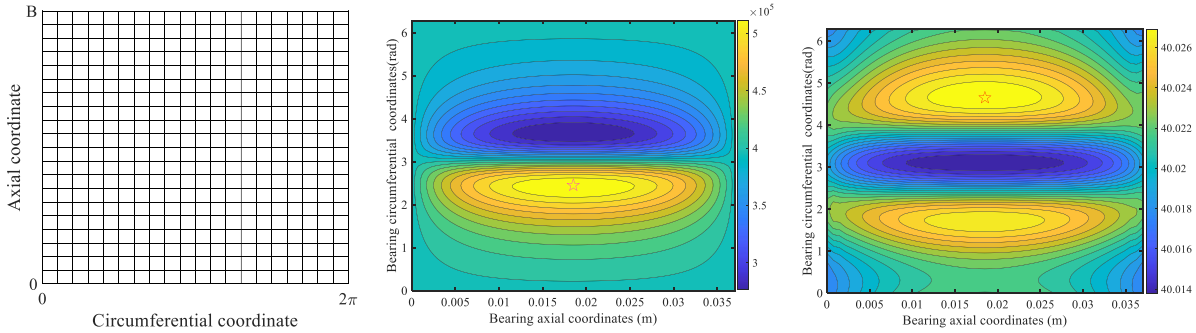


(a) Bearing structure (b) Fluid domain

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

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 $(\mathrm{kg\ m^{-3}})$	779
Velocity U $(\mathrm{m\ s^{-1}})$	$U = \pi R n_r/30$



(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.

where $F_0 = \int_0^h \frac{1}{\eta}\mathrm{d}y$, $F_1 = \int_0^h \frac{y}{\eta}\mathrm{d}y$, and $F_2 = \int_0^h \frac{y^2}{\eta}\mathrm{d}y - \frac{\int_0^h \frac{y}{\eta}\mathrm{d}y \int_0^h \frac{y}{\eta}\mathrm{d}y}{\int_0^h \frac{1}{\eta}\mathrm{d}y}$. 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

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

The model parameters are displayed in table 4.

The finite difference method and the successive over-relaxation method are used to solve the above equations, where the discrete computational area of the film is given in figure 11, together with the solved field of pressure, temperature and their maxima with eccentricity ratio, attitude angle, rotation speed and initial viscosity being set to 0.6, 0.1, 6200 and 9.78×10^{-4}, respectively. Next, the observation data $(527\,499.1407\ \mathrm{Pa}, 40.0417\ \mathrm{C}^\circ)$ 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 is assumed to be $N(0.4, 0.1^2)$, $N(0, 0.2^2)$, $N(6000, 200^2)$ and $N(9.5 \times 10^{-4}, (5 \times 10^{-5})^2)$, respectively.

Using the same parameter settings for the algorithm, the results of the evidence Z are reported in table 5, where the last column reports the average number of model calls across 10

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

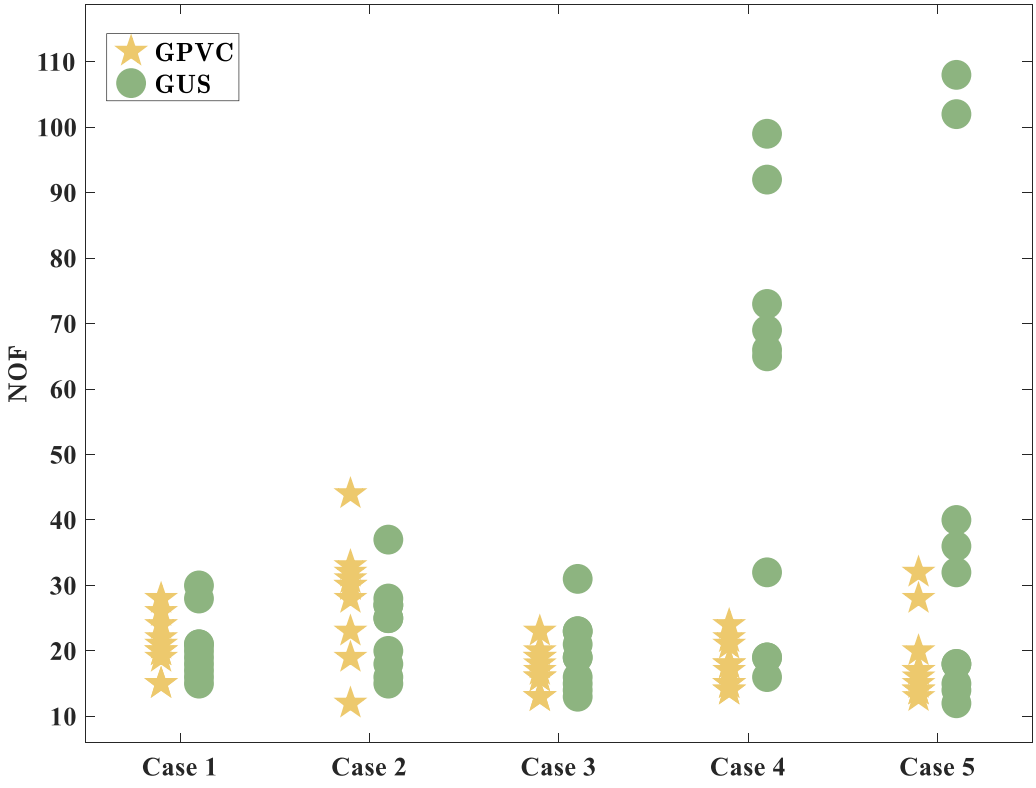


Figure 12. Distribution of the number of model calls required by each setting across 10 replications for the lubrication model.

replications. Due to the high computational cost of each model call being expensive, the reference results are not computed by MCS and the results from the two acquisition functions are cross-validated. As shown, the mean estimates from all the replications match well with one another and are thus believed to be accurate. It is also shown 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 number of model calls required 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 requires fewer simulator calls than GUS, although the superiority is not as significant as the last two examples. We find that the GPVC usually shows higher superiority for models with higher nonlinear behaviors.

The results for 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

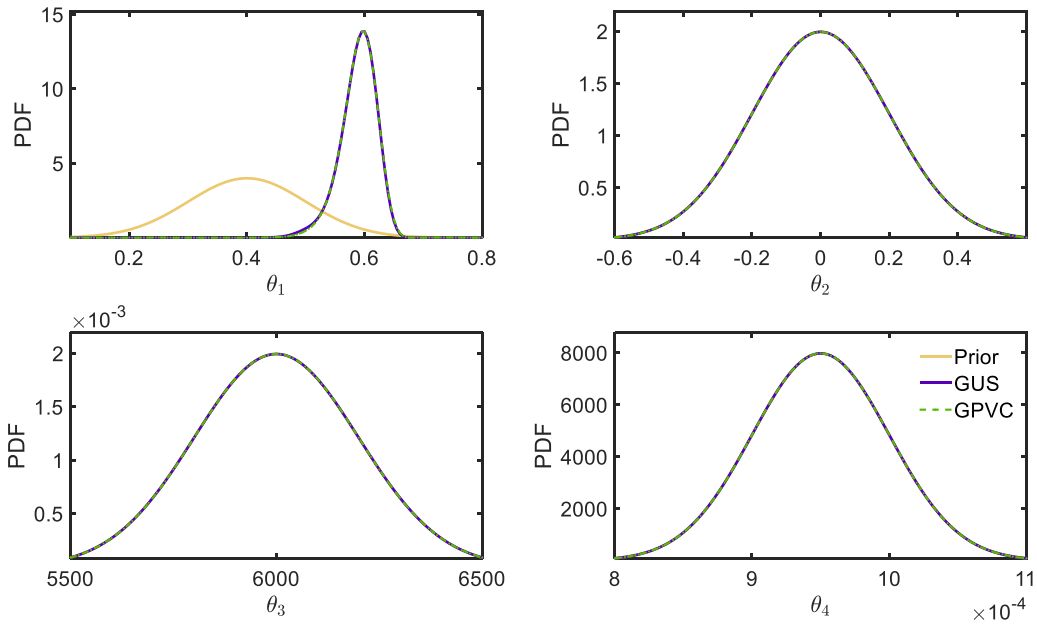


Figure 13. Results of the marginal posterior probability density function (PDF) of the lubrication models.

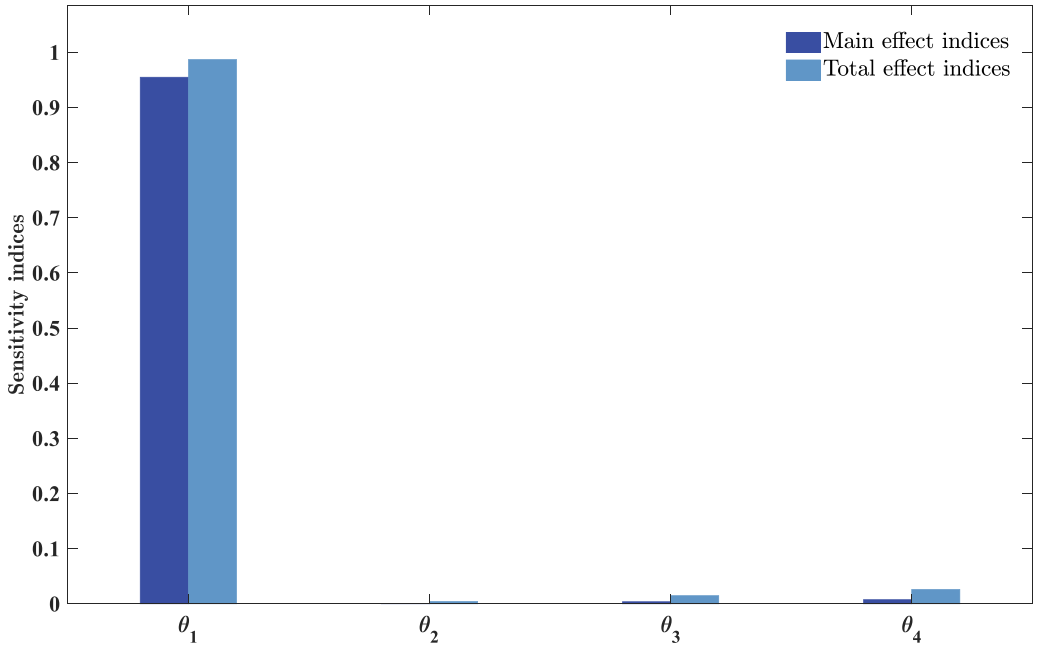


Figure 14. Results of variance-based sensitivity indices for the lubrication models.

096

the marginal density are reported together with their priors. It should be noted that this marginal posterior density of each $\boldsymbol{\theta}$, and the closed-form expression of the mean are reported in appendix D. As can be seen from figure 13, the posterior density estimated by GUS and GPVC match well, indicating 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 proven in our long-term test. To demonstrate this, the total effect variance-based sensitivity indices are estimated and are reported in figure 14 (see [45] for details). It is shown that the sensitivity indices of $\theta_2 \sim \theta_4$ are all close to zero, indicating that they are non-influential. This is in good agreement with the above conclusions. This may motivate us to first perform sensitivity analysis before implementing Bayesian updating of high-dimensional parameters. This can 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 a reduction of the prediction uncertainty of the lubrication model.

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 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 to BMU, a generalized version of it has been proposed to provide sufficient flexibility and potential for balancing the exploration and exploitation of the algorithm, and to accelerate convergence. Another appealing feature of the active learning procedure is the high efficiency of each query, benefiting from the closed-form expressions of the GPVC acquisition function.

Results of three examples under extensive different settings have demonstrated that, given the same level of accuracy tolerance, the proposed GPVC function requires a much smaller number of model calls and thus is more efficient, even for the challenging task of estimating multimodal 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 is required. Examples include 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 an extension to the joint calibration of deterministic-but-unknown model parameters and model bias. All this will be conducted in future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant number 52475164, the CIRP 2023 project granted by the Commercial Aircraft Corporation of China, Ltd. (COMAC), and 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 $\Pi_p[\kappa(\boldsymbol{\theta},\boldsymbol{\Theta})]$, $\Pi_p'[\kappa(\boldsymbol{\theta},\boldsymbol{\theta}')]$ and $\Pi_p\Pi_p'[\kappa(\boldsymbol{\theta},\boldsymbol{\theta}')]$ are given by:

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

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

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

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

Appendix B. Closed-form expressions for the 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 $\hat{Z}_M\sim\mathcal{N}\left(\mu_{Z,M},\sigma_{Z,M}^2\right)$. Before inferring the posterior mean and variance of integral $\hat{Z}_M=\Pi_p\left[\hat{p}_M^{\mathcal{L}}(\mathcal{D}_{\mathrm{obs}}|\boldsymbol{\theta})\right]$, 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 is formulated as:

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

Based on equation (B.1), equation (B.4) can be further derived as:

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

098

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 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 \times \mathcal{Y}^\top \mathcal{K}^{-1} \left\{ \begin{array}{c} \kappa(\Theta,\Theta|3\boldsymbol{\Sigma}) \times \kappa\left(\frac{4\Theta}{3}, -\frac{2\Theta}{3} \left| 2\boldsymbol{\Sigma} + 4\mathbf{I}_d + \frac{2}{3}\boldsymbol{\Sigma}\right.\right) \\ \times\kappa \left(\begin{array}{c} \left[\frac{\Theta}{3}\left(\frac{2}{3}\boldsymbol{\Sigma}\right)^{-1} - \Theta(2\boldsymbol{\Sigma} + 4\mathbf{I}_d)^{-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 \right. \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(\Theta,\Theta|2\boldsymbol{\Sigma}) \times \kappa(\Theta,-\Theta|2\boldsymbol{\Sigma} + 4\mathbf{I}_d)\right] \mathcal{K}^{-1} \left[\kappa(\Theta,\Theta|2\boldsymbol{\Sigma}) \times \kappa(\Theta,-\Theta|2\boldsymbol{\Sigma} + 4\mathbf{I}_d)\right] \mathcal{K}^{-1}\mathcal{Y}.
\end{aligned} \tag{B.4}$$

Appendix C. Analytical results for the GPVC function

Similar to appendix B, the $\rho(\boldsymbol{\theta})$ involved in the GPVC can be deduced as:

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

Appendix D. Analytical results for posterior marginal density

Similar to appendices B and C, the marginal PDF of model parameter $\boldsymbol{\theta}$ can be inferred as:

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

References

[1] Mottershead J E, Link M and Friswell M I 2011 The sensitivity method in finite element model updating: a tutorial *Mech. Syst. Signal Process.* **25** 2275–96

[2] Faes M, Broggi M, Patelli E, Govers Y, Mottershead J, Beer M and Moens D 2019 A multivariate interval approach for inverse uncertainty quantification with limited experimental data *Mech. Syst. Signal Process.* **118** 534–48

[3] Kennedy M C and O'Hagan A 2001 Bayesian calibration of computer models *J. R. Stat. Soc.* B **63** 425–64

[4] Wu X, Kozlowski T, Meidani H and Shirvan K 2018 Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, part 1: theory *Nucl. Eng. Des.* **335** 339–55

[5] Wu X, Xie Z, Alsafadi F and Kozlowski T 2021 A comprehensive survey of inverse uncertainty quantification of physical model parameters in nuclear system thermal–hydraulics codes *Nucl. Eng. Des.* **384** 111460

[6] Bingham D, Butler T and Estep D 2024 Inverse problems for physics-based process models *Annu. Rev. Stat. Appl.* **11** 461–82

[7] Bi S, Beer M, Cogan S and Mottershead J 2023 Stochastic model updating with uncertainty quantification: an overview and tutorial *Mech. Syst. Signal Process.* **204** 110784

[8] Lye A, Cicirello A and Patelli E 2021 Sampling methods for solving Bayesian model updating problems: a tutorial *Mech. Syst. Signal Process.* **159** 107760

[9] Carrera B and Papaioannou I 2024 Covariance-based MCMC for high-dimensional Bayesian updating with sequential Monte Carlo *Probab. Eng. Mech.* **77** 103667

[10] Tierney L 1994 Markov chains for exploring posterior distributions *Ann. Stat.* **22** 1701–28

[11] Ching J and Chen Y-C 2007 Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection and model averaging *J. Eng. Mech.* **133** 816–32

[12] Betz W, Papaioannou I and Straub D 2016 Transitional Markov chain Monte Carlo: observations and improvements *J. Eng. Mech.* **142** 04016016

[13] Radivojević T and Akhmatskaya E 2020 Modified Hamiltonian Monte Carlo for Bayesian inference *Stat. Comput.* **30** 377–404

[14] Bardsley J M, Solonen A, Haario H and Laine M 2014 Randomize-then-optimize: a method for sampling from posterior distributions in nonlinear inverse problems *SIAM J. Sci. Comput.* **36** A1895–910

[15] Fang H, Tian N, Wang Y, Zhou M and Haile M A 2018 Nonlinear Bayesian estimation: from Kalman filtering to a broader horizon *IEEE/CAA J. Autom. Sin.* **5** 401–17

[16] Song J, Wei P, Valdebenito M and Beer M 2020 Adaptive reliability analysis for rare events evaluation with global imprecise line sampling *Comput. Methods Appl. Mech. Eng.* **372** 113344

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

[18] Straub D and Papaioannou I 2015 Bayesian updating with structural reliability methods *J. Eng. Mech.* **141** 04014134

[19] Kitahara M, Dang C and Beer M 2023 Bayesian updating with two-step parallel Bayesian optimization and quadrature *Comput. Methods Appl. Mech. Eng.* **403** 115735

[20] Song C, Wang Z, Shafieezadeh A and Xiao R 2022 BUAK-AIS: efficient Bayesian updating with active learning Kriging-based adaptive importance sampling *Comput. Methods Appl. Mech. Eng.* **391** 114578

[21] Giovanis D G, Papaioannou I, Straub D and Papadopoulos V 2017 Bayesian updating with subset simulation using artificial neural networks *Comput. Methods Appl. Mech. Eng.* **319** 124–45

[22] O'Hagan A 1991 Bayes–hermite quadrature *J. Stat. Plan. Inference* **29** 245–60

[23] Osborne M, Garnett R, Ghahramani Z, Duvenaud D K, Roberts S J and Rasmussen C 2012 Active learning of model evidence using Bayesian quadrature *Advances in Neural Information Processing Systems* vol 25

[24] Hennig P, Osborne M A and Kersting H P 2022 *Probabilistic Numerics: Computation as Machine Learning* (Cambridge University Press)

[25] Baptista R, Marzouk Y and Zahm O 2023 On the representation and learning of monotone triangular transport maps *Found. Comput. Math.* **24** 2063–108

[26] Gunter T, Osborne M A, Garnett R, Hennig P and Roberts S J 2014 Sampling for inference in probabilistic models with fast Bayesian quadrature *Advances in Neural Information Processing Systems* vol 27

[27] Kanagawa M, Hennig P, Sejdinovic D and Sriperumbudur B K 2018 Gaussian processes and kernel methods: a review on connections and equivalences (arXiv:1807.02582)

[28] Rasmussen C E and Ghahramani Z 2003 Bayesian Monte Carlo *Advances in Neural Information Processing Systems* pp 505–12

[29] Wei P, Zhang X and Beer M 2020 Adaptive experiment design for probabilistic integration *Comput. Methods Appl. Mech. Eng.* **365** 113035

[30] Huszár F and Duvenaud D 2012 Optimally-weighted herding is Bayesian quadrature *Int. Conf. on Uncertainty in Artificial Intelligence (UAI)*

[31] Kanagawa M and Hennig P 2019 Convergence guarantees for adaptive Bayesian quadrature methods *Advances in Neural Information Processing Systems* vol 32

[32] Briol F-X, Oates C J, Girolami M, Osborne M A and Sejdinovic D 2019 Probabilistic integration: a role in statistical computation? *Stat. Sci.* **34** 1–22

[33] Karvonen T and Särkkä S 2017 Classical quadrature rules via Gaussian processes *2017 IEEE 27th Int. Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE) pp 1–6

[34] Kitahara M and Kitahara T 2025 Sequential and adaptive probabilistic integration for Bayesian model updating *Mech. Syst. Signal Process.* **223** 111825

[35] Song J, Liang Z, Wei P and Beer M 2025 Sampling-based adaptive Bayesian quadrature for probabilistic model updating *Comput. Methods Appl. Mech. Eng.* **433** 117467

[36] Hong F, Wei P, Bi S and Beer M 2025 Efficient variational Bayesian model updating by Bayesian active learning *Mech. Syst. Signal Process.* **224** 112113

[37] Adachi M, Hayakawa S, Jørgensen M, Oberhauser H and Osborne M A 2022 Fast Bayesian inference with batch Bayesian quadrature via kernel recombination *Advances in Neural Information Processing Systems* vol 35 pp 16533–47

[38] Acerbi L 2019 An exploration of acquisition and mean functions in variational Bayesian Monte Carlo *Symp. on Advances in Approximate Bayesian Inference* (PMLR) pp 1–10

[39] Frazier P I 2018 A tutorial on Bayesian optimization (arXiv:1807.02811)

[40] Song J, Cui Y, Wei P, Valdebenito M A and Zhang W 2024 Constrained Bayesian optimization algorithms for

estimating design points in structural reliability analysis *Reliab. Eng. Syst. Saf.* **241** 109613
[41] Wei P, Zheng Y, Fu J, Xu Y and Gao W 2023 An expected integrated error reduction function for accelerating Bayesian active learning of failure probability *Reliab. Eng. Syst. Saf.* **231** 108971
[42] Rosenblatt M 1952 Remarks on a multivariate transformation *Ann. Math. Stat.* **23** 470–2
[43] Beck J L and Au S-K 2002 Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation *J. Eng. Mech.* **128** 380–91
[44] Li B, Sun J, Zhu S, Fu Y, Zhao X, Wang H, Teng Q, Ren Y, Li Y and Zhu G 2019 Thermohydrodynamic lubrication analysis of misaligned journal bearing considering the axial movement of journal *Tribol. Int.* **135** 397–407
[45] Song J, Wei P, Valdebenito M A, Faes M and Beer M 2022 Data-driven and active learning of variance-based sensitivity indices with Bayesian probabilistic integration *Mech. Syst. Signal Process.* **163** 108106

Assoc. Prof. Pengfei Wei is an experienced principal investigator, in the directions of Bayesian intelligent computing, model VV&UQ, and reliability engineering, and is affiliated to the school of power and energy of Northwestern Polytechnical University, China.

Dr. Masaru Kitahara is Assistant Professor at Department of Civil Engineering of the University of Tokyo and works in the Bridge and Wind Engineering Laboratory.

Prof. Dr. Matthias Faes is head of the Chair for Reliability Engineering and full professor at the mechanical engineering department of TU Dortmund University.

Prof. Dr. Michael Beer is the head of the Institute for Risk and Reliability at Leibniz University Hannover as well as part-time Professor of Uncertainty in Engineering in the University of Liverpool, UK and a Guest Professor at Tongji University, China.