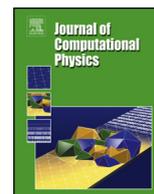


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Journal of Computational Physics

www.elsevier.com/locate/jcp

Fixed inducing points online Bayesian calibration for computer models with an application to a scale-resolving CFD simulation



Yu Duan*, Matthew D. Eaton, Michael J. Bluck

Nuclear Engineering Group/Rolls-Royce Modelling and Analytics (M&A) Group, Department of Mechanical Engineering, Imperial College London, SW7 2BX, UK

ARTICLE INFO

Article history:

Available online 1 March 2021

Dataset link:

<https://doi.org/10.5281/zenodo.4562382>

Keywords:

Online learning

Bayesian calibration

Stochastic variational inference

Gaussian processes

ABSTRACT

This paper proposes a novel fixed inducing points online Bayesian calibration (FIPO-BC) algorithm to efficiently learn the model parameters using a benchmark database. The standard Bayesian calibration (STD-BC) algorithm provides a statistical method to calibrate the parameters of computationally expensive models. However, the STD-BC algorithm does not scale well with regard to the number of data points and also it lacks an online learning capability. The proposed FIPO-BC algorithm greatly improves the computational efficiency of the algorithm and, in addition, enables online calibration to be performed by executing the calibration on a set of predefined inducing points.

To demonstrate the procedure of the FIPO-BC algorithm, two tests are performed, finding the optimal value and exploring the posterior distribution of 1) the parameter in a simple function, and 2) the high-wave number damping factor in a scale-resolving turbulence model (scale adaptive simulation shear-stress transport model/SAS-SST). The results (such as the calibrated model parameter and its posterior distribution) of FIPO-BC with different inducing points are compared to those of STD-BC. It is found that FIPO-BC and STD-BC can provide very similar results, once the predefined set of inducing points in FIPO-BC is sufficiently fine. Given that fewer datapoints are needed in the proposed FIPO-BC algorithm, compared to the STD-BC algorithm, it will be a more computational efficient algorithm. In our demonstration test cases, the proposed FIPO-BC algorithm is at least ten times faster than the STD-BC algorithm. Meanwhile, the online feature of the FIPO-BC allows continuous updating of the calibration outputs and potentially reduces the workload on generating the database.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Computational modelling is now an integral tool to predict and understand complex physical systems in many branches of science and engineering. Computational modelling enables engineers and scientists to analyse physical phenomena which cannot be easily measured by experiments and reduces the time and financial cost of analysing complex systems. However, a computational model is normally subject to various types of errors, such as discretisation error, parametric uncertainty error, and model error. The discretisation error, parametric uncertainty error and model error all contribute to the total

* Corresponding author.

E-mail address: y.duan@imperial.ac.uk (Y. Duan).

error in a mathematical or computational model. It is only by assessing and reducing each of the components to the total error that one can assess the accuracy of a given mathematical or computational model. The discretisation and parametric uncertainty errors are relatively easy to reduce, depending upon the computational resources and experimental data available. However, the model error is normally linked to the underlying mathematical model not representing the exact physics of the system being analysed. Therefore, model error is particularly challenging to both rigorously analyse and reduce in magnitude. In many applications, such as turbulence modelling, the model parameter is determined based upon approximations or simplified conditions. Therefore, model parameter calibration with respect to (w.r.t.) the benchmark data becomes important to improve the overall performance and accuracy of the model against experimental data and results.

The determination of the model parameters through calibration, data adjustment or data assimilation enables the model to more accurately represent the underlying physics of a system being analysed. By this definition, it is natural to calibrate the model parameters using the best-fit principle. However, the best-fit principle does not consider the uncertainties that the data may be subjected to, such as the observational/numerical discretisation errors, which should be considered in determining the agreement between two databases [1–4]. Kennedy and O’Hagan [5] proposed a Bayesian approach to calibrating the computational model by representing model bias and computer model outputs as the Gaussian processes. The method automatically integrates the uncertainties into the calibration process by treating them as Gaussian noise. For simplicity, we will refer to Kennedy and O’Hagan’s approach as standard Bayesian Calibration (STD-BC).

Since the development of STD-BC, the concept was adopted and further developed by many researchers in different areas of science and engineering. For example, Bayarri et al. [6–8] applied the method to validate models for resistance spot welding, dynamic stress and vehicle collision modelling. In computational fluid dynamic (CFD) simulation, it was applied to tuning the model parameters and inferring the true physical value [9,10], quantifying the inverse uncertainty [11–18] and informing the design of supercomputer simulations [19]. Recently, Wu et al. [20,21] modularised the STD-BC by dividing the database into two different sets; a calibration set and an inverse uncertainty quantification set. This framework has been successfully applied in Liu et al. [22] to carry out the inverse uncertainty quantification of the correlation in two-fluid model based multiphase-CFD simulations. Wu et al. [23] furtherly discussed the “lack of identifiability” issue of inverse uncertainty quantification using the Bayesian calibration. Karagiannis and Lin [24] proposed a mixture Bayesian Calibration framework, which can represent the real system by the mixture of the outputs of available computer models with different fidelity level.

Despite the successful applications listed above, the STD-BC is impractical for science and engineering problems involving very large data sets [25]. This is due to the poor scaling of inverting the covariance matrix with respect to (w.r.t.) the size of the data sets involved. Assuming we have a data set of size N (including numerical and benchmark data), there will be an N -by- N covariance matrix to invert. The computational complexity to invert an N -by- N matrix is N^3 . To solve this problem, Higdon et al. [26,27] followed the definition of Kennedy and O’Hagan but suggested to decompose the model output and model discrepancy using singular value decomposition (SVD). The weights generated in the SVD of the numerical output are treated as a function of the model parameters and spatial coordinate. The concept of the STD-BC is then applied to the weight functions. Zhang and Fu [16,17] improved the efficiency of the original Bayesian UQ framework by using the adaptive HDMR technique to decompose the original high dimensional problems into several lower-dimensional sub-problems. These treatments sub-sequentially improve efficiency, however, it does not invoke the online learning capability. Once more benchmark data becomes available, users have to execute, from scratch, the whole procedure with a larger dataset. The work presented in this paper aims to provide a solution to improve the efficiency of the STD-BC and enable the online capability using methods developed within the research literature on Gaussian processes (GP).

GP is a naïve Bayesian supervised machine learning algorithm. By assuming the data follows an adjoint multivariate Gaussian distribution, it offers the potential to determine the mean and variances of the unknown. As is the case with the STD-BC, the poor scalability of the inversion of the covariance matrix has limited the usage of the GP. In recent years, many efforts had been made to develop a sparse Gaussian processes for very large data sets. A detailed review can be found in [28]. In general, there are several ways to increase the sparsity of a GP, such as the Nyström GP [29,30], deterministic training conditional (DTC) GP [31], and kernel interpolation for scalable structured GP (KISS-GP) [32]. In this paper, we propose to adapt the stochastic variational inference GP (SVI-GP) method [33] to improve the efficiency of the Bayesian calibration and enable the online learning capability. We name the new method ‘fixed inducing points online Bayesian calibration’ (FIPO-BC).

The rest of this paper is organised as follows: A brief introduction to the SVI-GP and STD-BC is given in section 2 which is followed by the description of FIPO-BC in section 3. In section 4, we summarise the results of a toy case, while section 5 is contributed to calibrate a key model parameter in a scale resolving turbulence model (SAS-SST model). This paper is closed by a summary and discussion of future work in Section 5.

2. Background

According to Kennedy and O’Hagan [5], the physical process $f(\cdot)$, noisy observation $y_i \in \mathbb{R}$, and model outputs $h(\cdot, \cdot)$ at $\mathbf{x}_i \in \mathbb{R}^d$ has the following form:

$$y_i = f(\mathbf{x}_i) + \epsilon_i = h(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon_i, \quad (1)$$

where ϵ_i is the observation error and is subject to the normal distribution ($\mathcal{N}(0, \beta_{exp}^{-1})$) with zero mean and standard deviation $\sqrt{\beta_{exp}^{-1}}$, $\delta(\cdot)$ is the numerical model inadequacy and $\boldsymbol{\vartheta} \in \mathbb{R}^\theta$ is the vector of model parameters. The Bayesian calibration assumes the numerical dataset and observations is a joint GP and then proceeds to find the most plausible $\boldsymbol{\vartheta}$ via maximise the marginal likelihood.

2.1. Stochastic variational inference Gaussian processes (SVI-GP)

Before further description of the Bayesian calibration method, a concise description of Gaussian processes [34,35] and stochastic variational inference Gaussian processes (SVI-GP) [33] are given in this section. This allows us to introduce notation and expressions which will be used in the description of STD-BC and FIPO-BC methods.

The standard GP assumes that any finite subset of $\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d\}$ follows an adjoint Gaussian distribution. Considering the vector of noisy observations $\mathbf{y} = \{y_i \in \mathbb{R}\}_{i=1}^n$ taken at n locations $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$. In the standard GP, we have

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \beta_{exp}^{-1} \mathbf{I}) \tag{2}$$

$$p(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}_{ee}) \tag{3}$$

Using Eq. (2 & 3), we define the prior of the real-valued function $f(\mathbf{x})$ as an adjoint multivariate Gaussian distribution with zero means ($\mathbf{0}$) and covariance matrix (\mathbf{K}_{ee}). \mathbf{K}_{ee} is an $n \times n$ matrix, whose elements are calculated using a predefined covariance function which is typically dependent upon a set of hyperparameters. Hyperparameters can be selected by maximising the marginal likelihood $p(\mathbf{y} \mid \mathbf{X}) = \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \mathbf{K}_{ee} + \beta_{exp}^{-1} \mathbf{I})$. The posterior GP of unknown y^* at \mathbf{x}^* is $\mathcal{N}(y^* \mid \mathbf{K}^* \mathbf{K}_{ee}^{-1} \mathbf{y}, \mathbf{K}^{**} - \mathbf{K}^* \mathbf{K}_{ee}^{-1} (\mathbf{K}^*)^T + \beta_{exp}^{-1})$, where \mathbf{K}^* is a $1 \times n$ covariance vector and \mathbf{K}^{**} is calculated at \mathbf{x}^* using the covariance function. The need to invert the $n \times n$ covariance matrix \mathbf{K}_{ee} means a GP has a computational complexity of $\mathcal{O}(n^3)$.

The key idea of the SVI-GP method (or similar sparse GP approaches) is to expand the probability space of the GP with inducing points (or auxiliaries) [36]. Assuming, we can find inducing variables \mathbf{u} that are conditioned on m inducing points $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$ where $m \ll n$. We augment the GP prior as

$$p(\mathbf{f} \mid \mathbf{X}) = p(\mathbf{f} \mid \mathbf{u}, \mathbf{Z}, \mathbf{X}) p(\mathbf{u} \mid \mathbf{Z}) \tag{4}$$

Similar to $p(\mathbf{f} \mid \mathbf{X})$,

$$p(\mathbf{u} \mid \mathbf{Z}) = \mathcal{N}(\mathbf{u} \mid \mathbf{0}, \mathbf{K}_{uu}), \tag{5}$$

where \mathbf{K}_{uu} is a $m \times m$ covariance matrix evaluated between all inducing points. Let \mathbf{K}_{eu} (an $n \times m$ matrix) be the covariance matrix between \mathbf{X} and \mathbf{Z} . We will then have

$$p(\mathbf{f} \mid \mathbf{u}, \mathbf{Z}, \mathbf{X}) = \mathcal{N}(\mathbf{f} \mid \mathbf{K}_{eu} \mathbf{K}_{uu}^{-1} \mathbf{u}, \tilde{\mathbf{K}}), \tag{6}$$

where $\tilde{\mathbf{K}} = \mathbf{K}_{ee} - \mathbf{K}_{eu} \mathbf{K}_{uu}^{-1} \mathbf{K}_{eu}$ and \mathbf{K}_{ue} is the transpose of \mathbf{K}_{eu} . The task then becomes finding the proper \mathbf{u} and hyperparameters with knowledge of \mathbf{y} . Parameters in $q(\mathbf{u})$, including $\boldsymbol{\mu}$, \mathbf{S} , and hyperparameters of the kernel function, are optimised using the lower boundary of $p(\mathbf{y} \mid \mathbf{X})$. In the SVI-GP, the objection function can be written as

$$L_{SVI-GP} = \sum_{i=1}^n \left\{ \log N(y_i \mid \mathbf{k}_i^T \mathbf{K}_{uu}^{-1} \boldsymbol{\mu}, \beta_{exp}^{-1}) - \frac{\tilde{k}_{ii} + \mathbf{k}_i \mathbf{K}_{uu}^{-1} \mathbf{S} \mathbf{K}_{uu}^{-1} \mathbf{k}_i^T}{2\beta_{exp}^{-1}} \right\} - KL(q(\mathbf{u}) \mid p(\mathbf{u})) \tag{7}$$

with \mathbf{k}_i being the i^{th} row of \mathbf{K}_{eu} and the \tilde{k}_{ii} is the diagonal element of $\tilde{\mathbf{K}}$.

It is given in [33] that the gradients of L_{SVI-GP} w.r.t. the parameters of $q(\mathbf{u})$ are:

$$\begin{cases} \frac{\partial L_{SVI-GP}}{\partial \boldsymbol{\mu}} = \beta_{exp} \mathbf{K}_{uu}^{-1} \mathbf{K}_{ue} \mathbf{y} - \boldsymbol{\Lambda} \boldsymbol{\mu} \\ \frac{\partial L_{SVI-GP}}{\partial \mathbf{S}} = \frac{1}{2} \mathbf{S}^{-1} - \frac{1}{2} \boldsymbol{\Lambda} \end{cases} \tag{8}$$

$\boldsymbol{\Lambda}$ in Eq. (8) is $\beta_{exp} \mathbf{K}_{uu}^{-1} \mathbf{K}_{ue} \mathbf{K}_{eu} \mathbf{K}_{uu}^{-1} + \mathbf{K}_{uu}^{-1}$. When the training data is just one set and the hyperparameters are selected, the optimal parameters of $q(\mathbf{u})$ are $\mathbf{S} = \boldsymbol{\Lambda}$ and $\mathbf{u} = \beta_{exp} \boldsymbol{\Lambda}^{-1} \mathbf{K}_{uu}^{-1} \mathbf{K}_{ue} \mathbf{y}$. The computational complexity to evaluate L_{SVI-GP} is of $\mathcal{O}(nm^2)$.

Once the training data is supplied to the SVI-GP model sequentially, the stochastic variational inference is then used to approximate the parameters in $q(\mathbf{u})$ and enables online training. The SVI algorithm works by taking steps in the direction of the approximate natural gradient, which is defined as the inverse Fisher information scaled usual gradient [33]. To use the natural gradient, it is worth reiterating the canonical parameters of $q(\mathbf{u})$:

$$\begin{cases} \boldsymbol{\theta}_1 = \mathbf{S}^{-1} \boldsymbol{\mu} \\ \boldsymbol{\theta}_2 = \frac{1}{2} \mathbf{S}^{-1} \end{cases} \quad (9)$$

The change of the canonical parameters in the natural gradient direction is then written as

$$\begin{cases} \boldsymbol{\theta}_{2(t+1)} = \frac{1}{2} \mathbf{S}_{(t)}^{-1} + \gamma \left(-\frac{1}{2} \mathbf{A} + \frac{1}{2} \mathbf{S}_{(t)}^{-1} \right) \\ \boldsymbol{\theta}_{1(t+1)} = \mathbf{S}_{(t)}^{-1} \boldsymbol{\mu}_{(t)} + \gamma \left(\beta_{exp} \mathbf{K}_{uu}^{-1} \mathbf{K}_{ue} \mathbf{y} - \mathbf{S}_{(t)}^{-1} \boldsymbol{\mu}_{(t)} \right) \end{cases} \quad (10)$$

In Eq. (10), γ represents the learning rate, while t is the iteration in the approximation of the parameters. For $\gamma = 1$, the optimal parameters obtained by Eq. (10) are the same as those obtained using Eq. (8). In practice for online training, the learning rate should be smaller than 1. In our experience, $\gamma \leq 0.2$ works well. For the online training, the computational complexity to evaluate L_{SVI-GP} is of $\mathcal{O}(n_i m^2)$, where n_i is the number of training data points in the i^{th} data set.

Once $q(\mathbf{u})$ is determined, the posterior GP mean and covariance can be approximated as described in [37]. There is difficulty in determining the optimised inducing inputs (\mathbf{z}) as it will introduce a dependence on the training data if the optimisation procedure is done w.r.t. it. To solve this problem, a proper design of the inducing points can be used [33,36]. To further simplify the procedure, the uniformly distributed inducing point is adopted in this paper. Therefore, it is important to check the suitability of the predefined inducing point by a sensitivity study, which is akin to mesh sensitivity studies in CFD simulations. Once the SVI-GP model is converged on the number of inducing points, further increments of the inducing points will not result in a large difference in the surrogate model accuracy.

2.2. Standard Bayesian calibration (STD-BC) revisited

Consider observations $\mathbf{y} = \{y_i \in \mathbb{R}\}_{i=1}^{m_1}$ at $\mathbf{X}_{exp} = \{\mathbf{x}_{exp,i} \in \mathbb{R}^d\}_{i=1}^{m_1}$, and simulation outputs $\mathbf{h} = \{h_{i,j} = M(\mathbf{x}_{num,i}, \boldsymbol{\vartheta}_j)\}$ at $\mathbf{X}_{num} = \{\mathbf{x}_{num,i} \in \mathbb{R}^d\}_{i=1}^{n_1}$ and a_1 sets of model parameters $\boldsymbol{\theta} = \{\boldsymbol{\vartheta}_i \in \mathbb{R}^{\theta}\}_{i=1}^{a_1}$. It should be noted that d and θ represent the dimension of $\mathbf{x}_{exp,i}$ and $\boldsymbol{\vartheta}_i$, respectively. The number of elements in \mathbf{h} is $N_1 = a_1 n_1$. It is well known that numerical outputs are also subject to numerical error (β_{num}). Therefore, we assume $\mathbf{h} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}_{nn})$, in which $\mathbf{L}_{nn} = \mathbf{K}_{num} + \beta_{num}^{-1} \mathbf{I}$ and \mathbf{K}_{num} is a $N_1 \times N_1$ matrix where the element is evaluated using $k_{num}(\mathbf{x}_{num,i}, \boldsymbol{\vartheta}_k), (\mathbf{x}_{num,j}, \boldsymbol{\vartheta}_l)$. Now, let us assume $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}_{ee})$, where $\mathbf{L}_{ee} = \mathbf{K}_{ee} + \beta_{exp}^{-1} \mathbf{I}$ and $\mathbf{K}_{ee} = \mathbf{K}_{num} + \mathbf{K}_{\delta}$ is the covariance matrix of true values (\mathbf{f}). \mathbf{K}_{δ} is a $m_1 \times m_1$ matrix for the model inadequacy with elements calculated using $k_{\delta}(\mathbf{x}_{exp,i}, \mathbf{x}_{exp,j})$. The hyperparameters in the k_{num} and k_{δ} are $\boldsymbol{\varphi}_{num}$ and $\boldsymbol{\varphi}_{\delta}$, respectively. Also, it should be noted that, for simplicity, we imply that the number of grid/mesh in numerical simulations are the same.

Let us denote $\mathbf{d}^T = [\mathbf{h} \ \mathbf{y}]$. According to the definition of the STD-BC [5], we then have

$$p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}, \boldsymbol{\varphi}_{\delta}) = \mathcal{N}(\mathbf{d} | \mathbf{0}, \mathbf{V}_{BB}) \quad (11)$$

In Eq. (11),

$$\mathbf{V}_{BB} = \begin{bmatrix} \mathbf{L}_{nn} & \mathbf{C}_{ne} \\ \mathbf{C}_{en} & \mathbf{L}_{ee} \end{bmatrix} \quad (12)$$

\mathbf{V}_{BB} is $(m_1 + N_1) \times (m_1 + N_1)$ matrix. \mathbf{C}_{en} is a $m_1 \times N_1$ cross-covariance matrix and is designed to describe the relationship between numerical outputs and observations with $\boldsymbol{\vartheta}$. The elements in \mathbf{C}_{en} are evaluated using $k_{num}(\mathbf{x}_{i,j}, \boldsymbol{\vartheta}_i), (\mathbf{x}_{exp,k}, \boldsymbol{\vartheta})$ and $\mathbf{C}_{ne} = \mathbf{C}_{en}^T$. It should be noted that we eliminated $\mathbf{X} = [\mathbf{X}_{num} \ \mathbf{X}_{exp}]$ in Eq. (11). It is noted in [5] that marginalised the effect of $\boldsymbol{\varphi}_{num}$ and $\boldsymbol{\varphi}_{\delta}$ would be too complex. Therefore, the optimal hyperparameters $\boldsymbol{\varphi}_{num}^*$ and $\boldsymbol{\varphi}_{\delta}^*$ are suggested to be used. It should be noted that β_{num} and β_{exp} are folded in $\boldsymbol{\varphi}_{num}^*$ and $\boldsymbol{\varphi}_{\delta}^*$ here.

With fixed hyperparameters, the plausibility of $\boldsymbol{\vartheta}$ is then expresses as

$$p(\boldsymbol{\vartheta} | \mathbf{d}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*) = \frac{p(\boldsymbol{\vartheta}) p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)}{\int p(\boldsymbol{\vartheta}) p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*) d\boldsymbol{\vartheta}} \quad (13)$$

Then, the objective in the coupled calibration is to obtain $\text{argmax}_{\boldsymbol{\vartheta} \in \mathcal{A}} p(\boldsymbol{\vartheta} | \mathbf{d}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$. The denominator in Eq. (13) is a constant and the prior of $\boldsymbol{\vartheta}$ ($p(\boldsymbol{\vartheta})$) should be as flat as possible to avoid strong bias. As a result, the shape and the maxima of $p(\boldsymbol{\vartheta} | \mathbf{d}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$ are effectively determined by $p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$. Then $\text{argmax}_{\boldsymbol{\vartheta} \in \mathcal{A}} p(\boldsymbol{\vartheta} | \mathbf{d}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$ becomes $\text{argmax}_{\boldsymbol{\vartheta} \in \mathcal{A}} p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$. The logarithm form of $p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*)$ can be written as

$$\log p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*) = -\frac{1}{2} \mathbf{d}^T \mathbf{V}_{BB}^{-1} \mathbf{d} - \frac{1}{2} \log |\mathbf{V}_{BB}| - \frac{m_1 + N_1}{2} \log 2\pi \quad (14)$$

The STD-BC is then to solve $\text{argmax}_{\boldsymbol{\vartheta} \in \mathcal{A}} \log(p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*))$ in which $\mathcal{A} \subseteq \mathbb{R}^{\theta}$. For the sake of simplicity, we will write $\log(p(\mathbf{d} | \boldsymbol{\vartheta}, \boldsymbol{\varphi}_{num}^*, \boldsymbol{\varphi}_{\delta}^*))$ as $\log(p(\mathbf{d} | \boldsymbol{\vartheta}))$ in the following discussion.

The flow chart for STD-BC is illustrated in the Fig. 1. STD-BC contains three major steps:

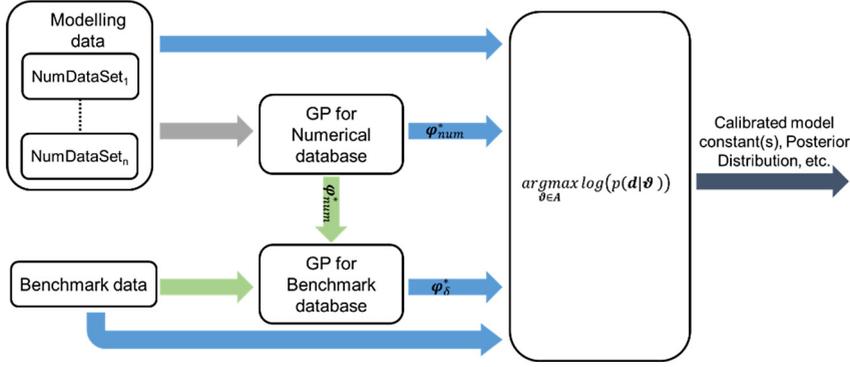


Fig. 1. Flow chart of the standard Bayesian calibration (STD-BC).

1. Determine a set of optimal hyperparameters (φ_{num}^*) for k_{num} w.r.t. the marginal likelihood of $\mathbf{h} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}_{nn})$.
2. Determine a set of optimal hyperparameters (φ_{δ}^*) for k_{δ} with knowledge of φ_{num}^* .
3. Solve $\text{argmax}_{\vartheta \in A} \log(p(\mathbf{d} | \vartheta))$ to obtain calibrated ϑ or to obtain the posterior distribution of ϑ w.r.t. $p(\mathbf{d} | \vartheta)$ for the inverse uncertainty quantification purpose.

3. Fixed inducing points online Bayesian calibration (FIPO-BC)

In the FIPO-BC framework, an extra layer formed by the inducing points is introduced to eliminate the dependence of the size of the covariance matrix on the available data. The form of the likelihood defined in the Bayesian calibration is applied to the variables on the inducing points. The likelihood is then marginalised w.r.t. the variational distribution of the inducing variables which is obtained using the SVI method.

Let $\tilde{\mathbf{u}} = \{u_i = f(\mathbf{z}_{exp,i}), \mathbf{z}_{exp,i} \in \mathbb{R}^d\}$ and $\mathbf{W} = \{w_{i,j} = M(\mathbf{z}_{num,i}, \boldsymbol{\omega}_j), \mathbf{z}_{num,i} \in \mathbb{R}^d \text{ and } \boldsymbol{\omega}_j \in \mathbb{R}^{\theta}\}$ to be the inducing variables for \mathbf{f} and \mathbf{h} . To differentiate the model parameter in the original dataset \mathbf{d} , the $\boldsymbol{\omega}_j$ is used to represent the vector of model parameter here. Let $\mathbf{Z}_{exp} = \{\mathbf{z}_{exp,i} \in \mathbb{R}^d\}_{i=1}^{m_2}$, $\mathbf{Z}_{num} = \{\mathbf{z}_{num,i} \in \mathbb{R}^d\}_{i=1}^{n_2}$, $\boldsymbol{\Omega} = \{\boldsymbol{\omega}_j \in \mathbb{R}^{\theta}\}_{j=1}^{a_2}$, and $\mathbf{Z} = [\mathbf{Z}_{num} \ \mathbf{Z}_{exp}]$. So \mathbf{W} contains $N_2 = a_2 n_2$ elements and $N_2 + m_2 \ll N_1 + m_1$. Let $\tilde{\mathbf{w}}$ be the vectorised \mathbf{W} and $\mathbf{d}_S^T = [\tilde{\mathbf{w}} \ \tilde{\mathbf{u}}]$. Similar to the STD-BC, we now define the $p(\mathbf{d}_S | \vartheta)$ as

$$p(\mathbf{d}_S | \vartheta, \tilde{\boldsymbol{\varphi}}_{num}^*, \tilde{\boldsymbol{\varphi}}_{\delta}^*) = \mathcal{N}(\mathbf{d}_S | \mathbf{0}, \mathbf{V}_{SS}) \tag{15}$$

and

$$\mathbf{V}_{SS} = \begin{bmatrix} \mathbf{L}_{ww} & \mathbf{C}_{wu} \\ \mathbf{C}_{uw} & \mathbf{L}_{uu} \end{bmatrix} \tag{16}$$

In Eq. (16), $\mathbf{L}_{ww} = \mathbf{K}_{ww}$ and $\mathbf{L}_{uu} = \mathbf{K}_{uu}$. \mathbf{K}_{ww} and $\mathbf{K}_{uu} = \mathbf{K}_{ww} + \mathbf{K}_{\delta'}$ are the covariance matrix for $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{u}}$, respectively. $\mathbf{K}_{\delta'}$ is the covariance matrix with element calculated using kernel function k_{δ} for the model inadequacy related to $\tilde{\mathbf{u}}$. \mathbf{C}_{uw} is the cross-covariance matrix for $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{u}}$ with element evaluated using k_{num} and $\mathbf{C}_{uw} = \mathbf{C}_{uw}^T$.

We further write the variational distribution of $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{u}}$ to be

$$q(\tilde{\mathbf{w}}) = \mathcal{N}(\tilde{\mathbf{w}} | \mathbf{M}_{\tilde{\mathbf{w}}}, \mathbf{S}_{ww}) \tag{17}$$

$$q(\tilde{\mathbf{u}}) = \mathcal{N}(\tilde{\mathbf{u}} | \mathbf{M}_{\tilde{\mathbf{u}}}, \mathbf{S}_{uu}) \tag{18}$$

Let $\mathbf{M}_S^T = [\mathbf{M}_{\tilde{\mathbf{w}}} \ \mathbf{M}_{\tilde{\mathbf{u}}}]$ and $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{ww} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{uu} \end{bmatrix}$, then

$$q(\mathbf{d}_S) = \mathcal{N}(\mathbf{d}_S | \mathbf{M}_S, \mathbf{S}) \tag{19}$$

Expectation of $p(\mathbf{d}_S | \vartheta, \tilde{\boldsymbol{\varphi}}_{num}^*, \tilde{\boldsymbol{\varphi}}_{\delta}^*)$ given $q(\mathbf{d}_S)$ is

$$\begin{aligned} \langle p(\mathbf{d}_S | \vartheta, \tilde{\boldsymbol{\varphi}}_{num}^*, \tilde{\boldsymbol{\varphi}}_{\delta}^*) \rangle_{q(\mathbf{d}_S)} &= \frac{1}{\sqrt{(2\pi)^{m_2+N_2} |\mathbf{V}_{SS} + \mathbf{S}|}} \exp\left(-\frac{1}{2} \mathbf{M}_S^T (\mathbf{V}_{SS} + \mathbf{S})^{-1} \mathbf{M}_S\right) \\ &= \mathcal{N}(\mathbf{M}_S | \mathbf{0}, \mathbf{V}_{SS} + \mathbf{S}) \end{aligned} \tag{20}$$

By eliminating the hyperparameters in the expression, we can now comfortably write $\langle p(\mathbf{d}_S | \vartheta, \tilde{\boldsymbol{\varphi}}_{num}^*, \tilde{\boldsymbol{\varphi}}_{\delta}^*) \rangle_{q(\mathbf{d}_S)}$ as $p(\mathbf{M}_S | \vartheta)$ and its logarithm form is

$$\log(p(\mathbf{M}_S | \boldsymbol{\vartheta})) = -\frac{1}{2} \mathbf{M}_S^T (\mathbf{V}_{SS} + \mathbf{S})_{BB}^{-1} \mathbf{M}_S - \frac{1}{2} \log |\mathbf{V}_{SS} + \mathbf{S}| - \frac{m_2 + N_2}{2} \log(2\pi) \quad (21)$$

The general flow chart of the FIPO-BC is illustrated in Fig. 2. The proposed FIPO-BC algorithm contains following steps:

1. Design the inducing points;
2. Determine a robust set of hyperparameters ($\tilde{\boldsymbol{\varphi}}_{num}^*$) and infer the model output ($\mathbf{M}_{\tilde{\mathbf{w}}}$) at the inducing points using the SVI-GP regarding to the whole set/a batch of model outputs;
3. Determine a robust set of hyperparameters ($\tilde{\boldsymbol{\varphi}}_{\delta}^*$) and infer the measurements ($\mathbf{M}_{\tilde{\mathbf{u}}}$) at the inducing points using the SVI-GP regarding to the whole set/a batch of the benchmark database;
4. Obtain the calibrated model parameters by solving $argmax_{\boldsymbol{\vartheta} \in \mathcal{A}} \log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ or obtain the posterior probability distribution of $\boldsymbol{\vartheta}$ w.r.t. $p(\mathbf{M}_S | \boldsymbol{\vartheta})$.
5. Update the calibrated $\boldsymbol{\vartheta}$ and posterior probability distribution of $\boldsymbol{\vartheta}$ with knowledge ($\mathbf{M}_{\tilde{\mathbf{w}}}$, $\mathbf{S}_{w\tilde{w}}$, $\mathbf{M}_{\tilde{\mathbf{u}}}$, and $\mathbf{S}_{u\tilde{u}}$) obtained in the previous FIPO-BC process, once more data is available.

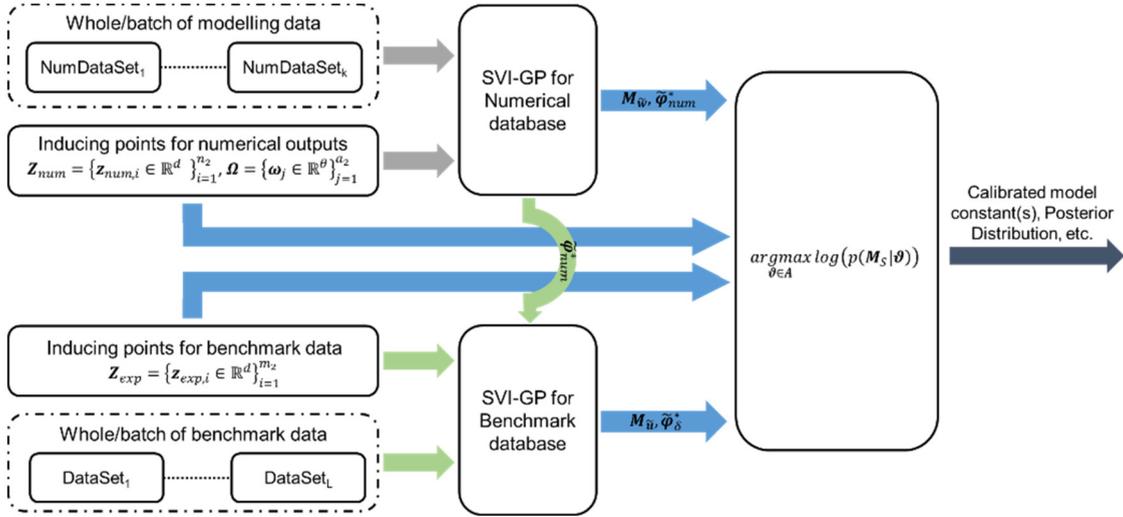


Fig. 2. Flow chart of the fixed inducing points online Bayesian calibration framework.

The computational complexities for training the surrogate models in the STD-BC and FIPO-BC algorithms are $\mathcal{O}(N_1^3 + m_1^3)$ and $\mathcal{O}(N_2^3 + m_2^3)$, respectively. The computational cost of calculating $\log(p(\mathbf{d} | \boldsymbol{\vartheta}))$ or $\log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ is dominated by the inversion of the covariance matrix. The $(N_1 + m_1) \times (N_1 + m_1)$ covariance matrix \mathbf{V}_{BB} in the STD-BC algorithm leads to a computational complexity of $\mathcal{O}((N_1 + m_1)^3)$, while the $\mathbf{V}_{SS} + \mathbf{S}$ in the FIPO-BC algorithm leads to a computational complexity of $\mathcal{O}((N_2 + m_2)^3)$. Since $N_2 \ll N_1$ and $m_2 \ll m_1$, the FIPO-BC algorithm is much more computationally efficient than the STD-BC algorithm. In the FIPO-BC algorithm, the calculation of $\log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ only relies on the parameters in the SVI-GP models. Hyperparameters ($\tilde{\boldsymbol{\varphi}}_{num}^*$ and $\tilde{\boldsymbol{\varphi}}_{\delta}^*$) and $q(\mathbf{d}_S)$ in FIPO-BC can be continuously updated according to the new training data using the SVI method as described in section 2.1. Accordingly, the calibrated $\boldsymbol{\vartheta}$ and/or posterior probability distribution of $\boldsymbol{\vartheta}$ can be updated by repeating step 4. Therefore, the FIPO-BC algorithm can be used in an online manner. Since the size of the covariance matrix in $\log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ is the same in the off-line and on-line FIPO-BC, the computational complexity for calculating $\log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ therefore the same as well. Like the STD-BC algorithm, the accuracy of the FIPO-BC algorithm is dependent on the validity of the surrogate models. Provided the SVI-GP models in the FIPO-BC algorithm can accurately represent the profile of the original data, the FIPO-BC algorithm will produce the similar outputs as the STD-BC algorithm.

4. Application of the FIPO-BC algorithm in a simple test case

This section verifies the implementation as well as demonstrating the practical application of the FIPO-BC algorithm by determining the parameter within a simple mathematical function. Both the STD-BC and FIPO-BC algorithms are used to identify the parameter in the mathematical function $f(x) = x \sin(4.0x)$. To distinguish the true function and model, the latter is expressed as $h(x, C) = x \sin(Cx)$. Moreover, forty measurements are manufactured by adding noise ($\epsilon \sim N(0, 0.05^2)$) to the function outputs. Outputs of $h(x, C)$ are obtained using different values for C but the same mesh. Twenty constants are sampled in the uniform distribution $U(0.0, 8.0)$ using the Latin hypercube sampling (LHS) method. Manufactured measurements and model outputs together with the profile of $f(x)$ are presented in Fig. 3(a). Extra validation data is

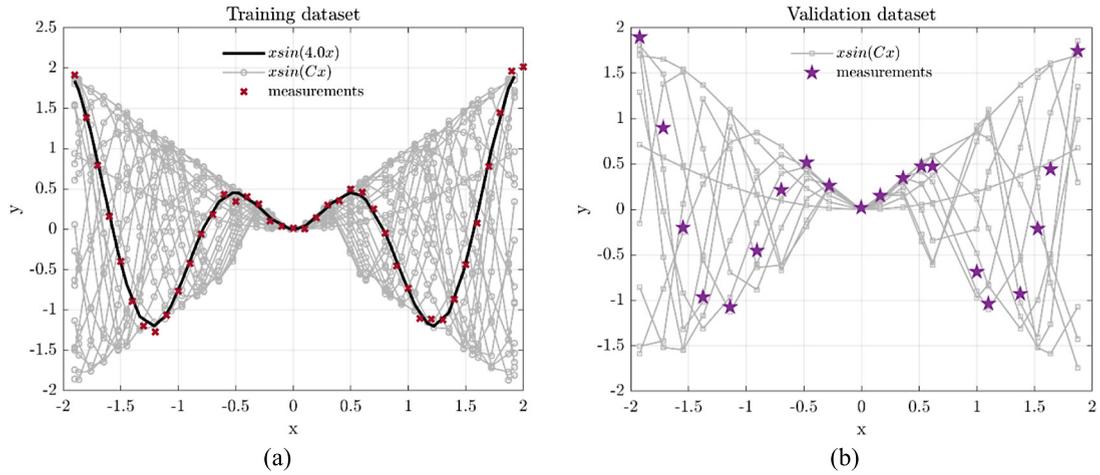


Fig. 3. (a) Training dataset: (model outputs with different constant ($x\sin(Cx)$) and manufactured measurements (benchmark database)); (b) Verification dataset.

Table 1

Number of points in the standard Bayesian calibration and resolution of fixed inducing points online Bayesian calibration.

Cases:	C	x	Measurements	Data points/Inducing points
STD-BC:				
Standard BC	20	40	40	840
FIPO-BC:				
Case 1	6	11	16	82
Case 2	9	16	21	165
Case 3	11	21	26	257
Case 4	17	26	31	473

generated to validate the hyperparameters of GP models in the calibration process. The validation dataset, shown in Fig. 3(b), contains twenty measurements and outputs of $h(x, C)$ with ten C values sampled in the uniform distribution $U(0.0, 8.0)$.

Four FIPO-BC cases with different sets of inducing point are generated. The details of the sets of the inducing points are documented in Table 1. From Case 1 to Case 4, the number of the inducing points gradually increases. Outputs of FIPO-BC algorithm, including the calibrated model parameter, the posterior distribution of the parameter, and the profile of the objective function, are compared to those of STD-BC algorithm. The hyperparameters of the surrogate models are optimised using the adaptive Nelder-Mead algorithm [38]. The validity of the GP/SVI-GP models for $h(x, C)$ and measurements (y) are checked using the validation dataset. For the sake of simplicity, the GP model is expressed as g and the SVI-GP model is expressed as g_s . Several attempts to determine the optimal hyperparameters have been done. In each attempt, the initial searching point of the adaptive Nelder-Mead simplex algorithm is carefully selected. Only the best-performing model is considered in the following discussion.

The quantity-to-quantity (Q-Q) plot and the mean square error (MSE) are illustrated in Fig. 4. Outputs of $g_s(x, C)$ are very similar to those of $h(x, C)$ except for Case 1, seeing Fig. 4(a). $g_s(x, C)$ in Case 1 is visibly different from the validation data with $MSE \cong 1.0E-1$. The MSEs of $g_s(x, C)$ in Cases 2, 3 & 4 are smaller than $1.0E-4$, while the MSE of $g(x, C)$ in standard BC (STD-BC) is just above $1.0E-4$. Interestingly, the MSE in Case 4 ($\sim 1.0E-5$) is larger than that in the Case 3 ($\sim 1.0E-6$). This observation may be explained by the non-marginalised hyperparameter effect, namely the results are not integrated w.r.t. the distribution of the hyperparameters. Since the MSE in both Case 3 and 4 are much smaller than the observational noise, such a difference is deemed to be ignorable. Furthermore, $g_s(x)$ and $g(x)$ for the manufactured measurement almost overlap with each other, referring to Fig. 4(b). The MSEs ($<1.0E-02$) of $g_s(x)$ and $g(x)$ are much smaller than the standard deviation (0.05) of ϵ . It is confident to claim that the surrogate models in the STD-BC case and last three FIPO-BC cases are valid.

The posterior distribution of C is obtained using the Metropolis-Hasting (MH) algorithm, by assuming that the prior distribution of C follows a uniform distribution in $[0, 8]$. 10,000 samples have been taken in each case. The posterior probability distribution functions (PDF) of C (represented in the form of histogram), the profiles of log marginal likelihood defined in both FIPO-BC and STD-BC, as well as maxima of the profiles are plotted in Fig. 5. Because of the deficiency of $g_s(x, C)$ in Case 1, the calibrated C is well away from 4.0. In other FIPO-BC cases and STD-BC, calibrated C is the same as the true value. With more inducing points added in the FIPO-BC, the profile of $\log(p(\mathbf{M}_S | \boldsymbol{\vartheta}))$ becomes more and more akin to the profile of $\log(p(\mathbf{d} | \boldsymbol{\vartheta}))$ until the FIPO-BC is converged on the inducing points, i.e. outputs of Case 4 are similar

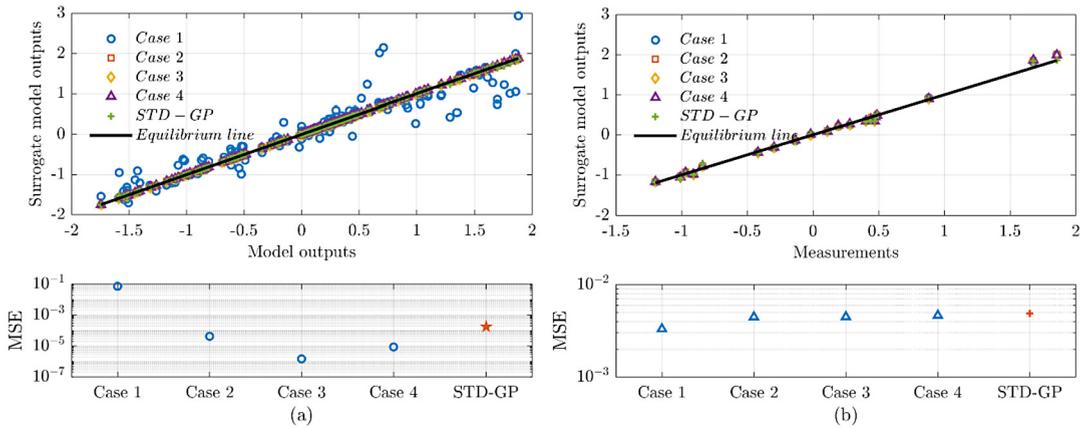


Fig. 4. Validation for (a) surrogate models for $h(x, C)$ (b) surrogate models for measurements in FIPO-BC and STD-BC.

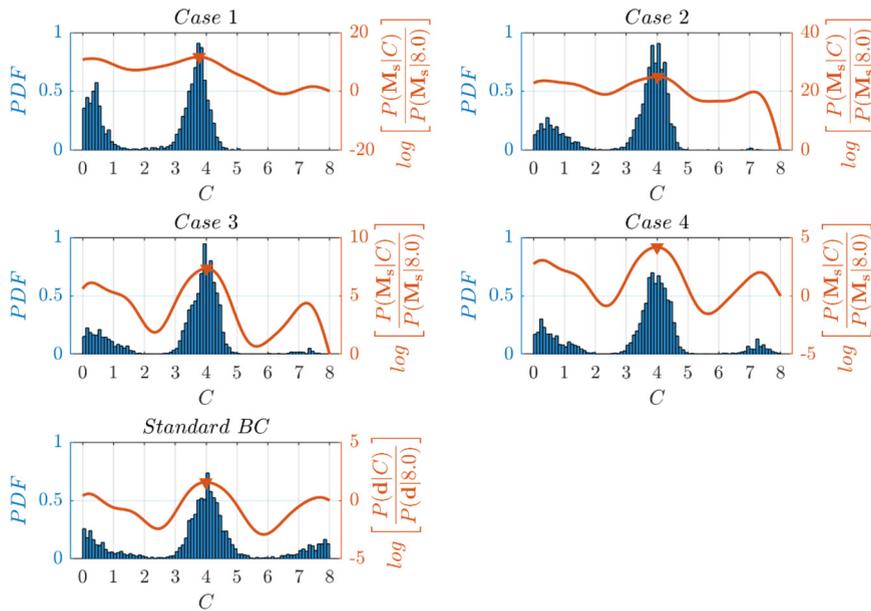


Fig. 5. The posterior probability distribution functions (PDF) of C and the log marginal likelihood for cases of FIPO-BC (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4 and (e) the STD-BC case.

to outputs of Case 3. The posterior probability distribution function of C in Case 4 is very similar to the outcome of STD-BC. Although the effects of hyperparameter in the STD-BC and FIPO-BC are not integrated out (or marginalised), the shape of $\log(p(\mathbf{M}_s | \vartheta))$ is about the same as the $\log(p(\mathbf{d} | \vartheta))$.

The CPU time per 1000 MH sampling in each case is presented in Fig. 6. The in-house Matlab code runs on a laptop with the Intel i7-7600U CPU. The time efficiency of MH sampling is greatly improved when using FIPO-BC algorithm. For instance, the CPU time per 1000 MH sampling in Case 3 is 135.8 s just a tenth of that in the STD-BC algorithm.

In order to demonstrate the online capability of the proposed method, the output of $h(x, C)$ in the training dataset is split into four batches. Each batch is separately fed into the FIPO-BC solver of Case 3. The results are shown in Fig. 7, which includes the value of C used to generate the data in the batches, response surface of true model output approximated using the $g_s(x, C)$, as well as the calibrated value of C at each step. With more data supplied to the SVI-GP solver, the response surface gradually converges, and therefore to the calibrated value of C . The calibrated C with the first set of data ('Batch-1') is 3.92. With the 'Batch-2' supplied to the calibration, while 'Batch-1' is omitted, the calibrated C is 4.0. Although the calibrated C shifted to 4.01 with 'Batch-3' and 'Batch-4', such a small difference does not cause much difference in the function output. Therefore, it is reasonable to claim that the online FIPO-BC converges with dataset 'Batch-2'.

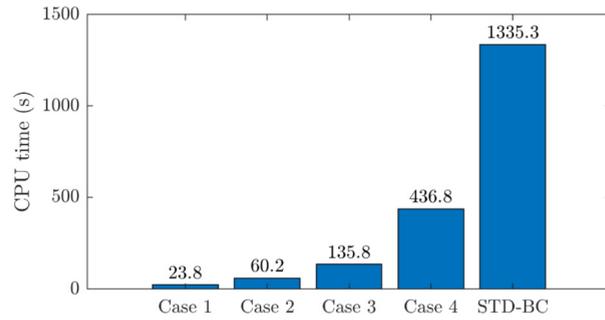


Fig. 6. CPU time(s) per 1000 MH sampling in each case.

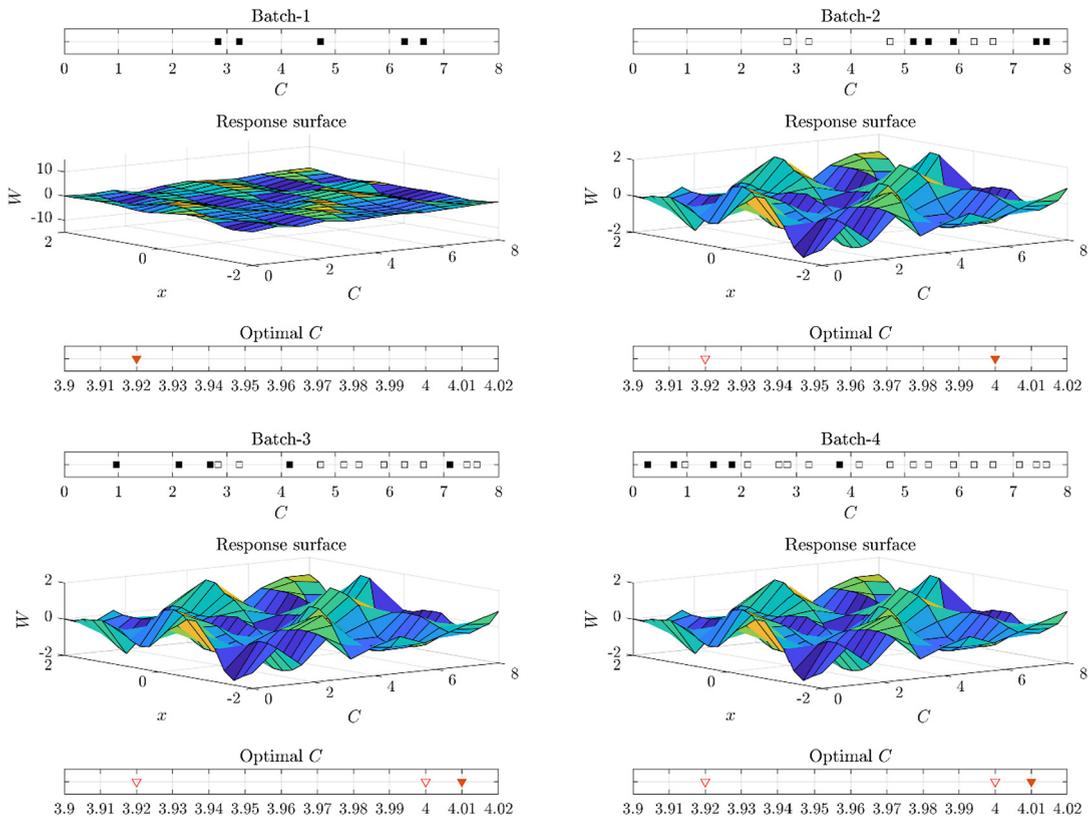


Fig. 7. The online calibration of the parameter in the toy case using FIPO-BC. Each pane shows the response surface of the inducing output (W) for the models as well as the calibrated parameter after a batch of model output using different C values, marked as solid points. Previously used (and discarded) C values and the calibrated constant are marked as empty points.

5. Calibrating C_s in a SAS-SST simulation

Turbulence modelling has been a continuous endeavour since the 1950s. The challenge is caused by the misalignment of the scale of a flow system and the turbulent eddies with the spatial scale of micrometre/millimetre and the time scale of microsecond/millisecond. Despite their small scale, the turbulent eddies dominate the behaviour of the turbulent flow. Major efforts have been spent in the development of turbulence models which describe the effect of turbulent eddies sufficiently well. In this section, the FIPO-BC algorithm is used to calibrate a damping factor (C_s) in a scale-resolving CFD turbulence model (SAS-SST). The SAS-SST model [39,40] aims to generate turbulent eddies using information obtained from the mean flow. Unrealistic turbulent eddies in the high-wave number region may be introduced into the simulated turbulent flow system. As a result, a specific term is designed to filter those unphysical turbulent eddies. The damping factor (C_s) controls the size of the filter and is the key parameter in this term. It changes the balance of the modelled and resolved turbulence effect, and hence affecting the model accuracy.

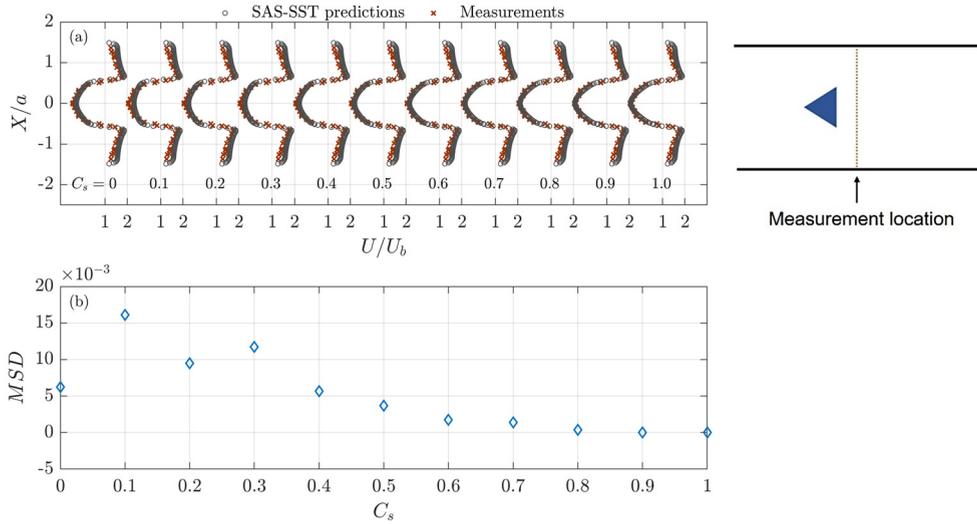


Fig. 8. (a) Profiles of U/U_b after the prism bluff-body predicted using SAS-SST model with different C_s , (b) mean square difference of U/U_b between different SAS-SST simulation and results obtained using $C_s = 0.9$.

Table 2

Numbers of the inducing points for calibrating C_s in SAS-SST using FIPO-BC and numbers of data point in the STD-BC.

Cases:	C_s	x-coordinate	Measurements	Inducing Points/Data points
FIPO-BC-1	6	36	11	227
FIPO-BC-2	9	36	16	340
FIPO-BC-3	11	36	21	417
FIPO-BC-4	11	41	26	477
FIPO-BC-5	11	46	26	532
STD-BC	11	121	54	1385

The FIPO-BC algorithm is applied to calibrate the C_s with regard to the streamwise velocity measurement of incompressible flow passing a prism bluff body [41,42]. The SAS-SST model is implemented within the commercial CFD solver, STAR-CCM+ 12.04, using user-defined functions. Eleven SAS-SST simulations are performed to provide the database for the surrogate model of the numerical simulation. Except for C_s , other settings, such as boundary conditions and numerical set-up, are the same in these simulations. Further details about how the simulations are set-up can be found in [4]. The chosen C_s are 0.0, 0.1, ..., 1.0. Each simulation takes more than 96,000 CPU hours to reach time-convergence. Experimental measurements and SAS-SST predictions of U/U_b just after the bluff-body are plotted in Fig. 8(a). After comparing the SAS-SST outputs to measurements graphically, it is clear that the SAS-SST with $C_s = 0.8, 0.9, \text{ and } 1.0$ better matches the measurements. Hereby, we plot the mean square difference (MSD) between SAS-SST prediction using $C_s = 0.9$ and other SAS-SST predictions in Fig. 8(b). The MSD of simulation using $C_s = 1.0$ is $6.6E-6$. It suggests the SAS-SST using C_s between 0.9 and 1.0 would produce almost the same U/U_b at the location.

The FIPO-BC with several different sets of inducing points are performed of which details are included in Table 2. In the first three cases, the number of the inducing points for C_s gradually increases from 6 to 11, while it is kept the same (36) on the spatial coordinate (X). Spatial inducing points increase to 41 and 46 in the last two cases, respectively. Again, the adaptive NM simplex algorithm is chosen to obtain the hyperparameters. But in this part of the study, the initial searching points are randomised. Dozens of adaptive NM searches have been made and the hyperparameters resulting in the most accurate surrogate model are chosen to show here. It should be noted that more attempt may require to obtain the proper hyperparameters for the surrogate models in other cases, since the initialization points are randomised.

Suitability checking of the surrogate models for CFD models and measurements are illustrated in Fig. 9. As in the toy case, the surrogate models in the FIPO-BCs are checked by using the inducing data points to predict the training dataset, whilst the leave-one-out cross-validation (LOO-CV) test is adopted to assess the validity of surrogate models in the STD-BC. As shown in Fig. 9(a), a visible improvement of surrogate model accuracy is achieved by refining the inducing points. Compared to the other FIPO-BC cases, data points produced in FIPO-BC-5 are closer to the equilibrium line in the quantity-to-quantity plot. With a small number of inducing points for spatial coordinate, the improvement on SVI prediction of CFD outputs due to refined C_s resolution is too small to be seen, referring to the MSE of FIPO-BC-1, 2, 3. As the number of inducing points for spatial coordinate increases, the MSE drops from $\sim 4.0E-4$ to $\sim 9.0E-5$ in FIPO-BC-4 and 5. Additionally, the surrogate model for CFD models in the STD-BC also performs well ($MSE \cong 1.0E-4$). As shown in Fig. 9(b), the SVI-GP model for measurement converges at FIPO-BC-2, containing 16 inducing points. The MSEs of SVI-GP predictions of measurements are around $3.0E-3$

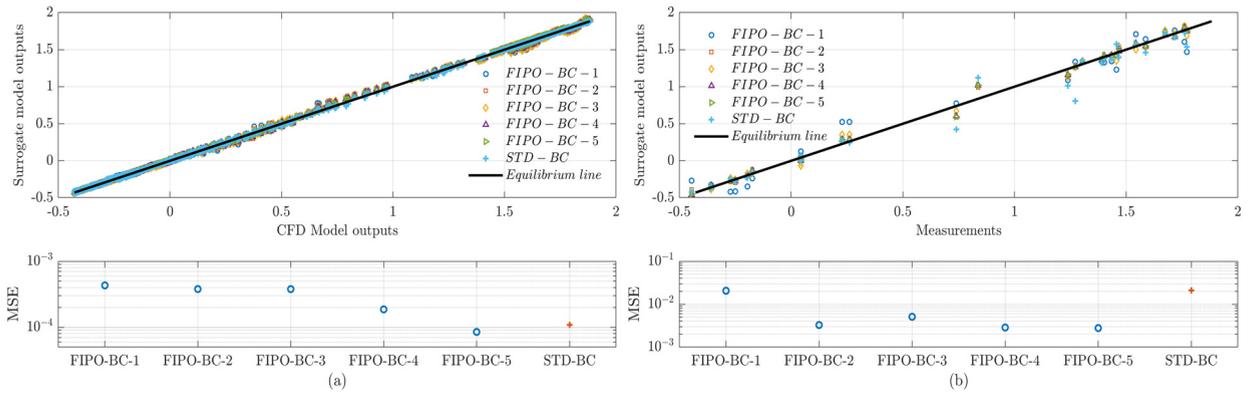


Fig. 9. Validation of surrogate models for (a) CFD outputs (b) measurements in FIPO-BC cases and STD-BC (Leave-one-out test is adopted to validate the surrogate models in STD-BC).

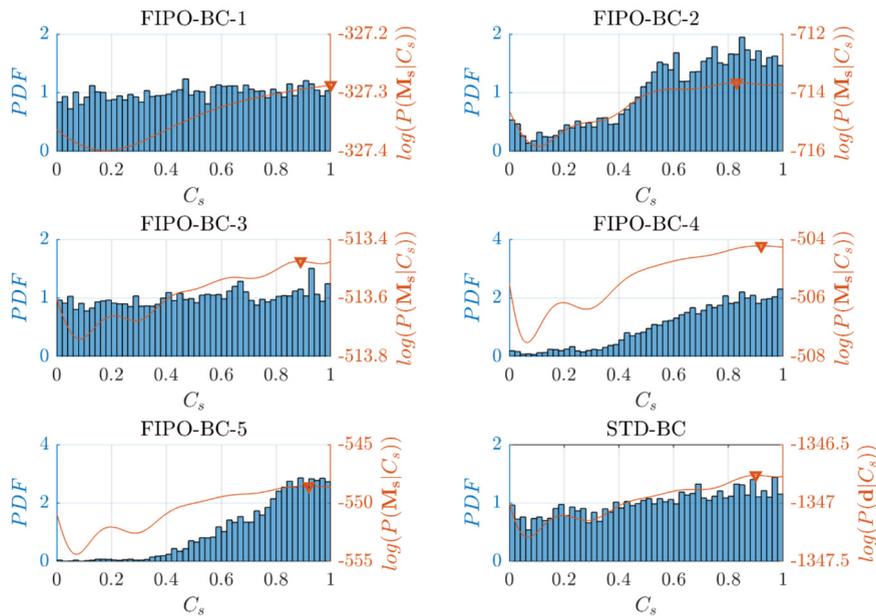


Fig. 10. The posterior PDF of C_s and the distributions log marginal likelihood for the FIPO-BC cases and the STD-BC case.

in FIPO-BC cases, except for FIPO-BC-1. Interestingly, the STD-GP model for the measurements in the STD-BC shows worse accuracy comparing to the SVI grid converged SVI-GP models, as its MSE is $\sim 2.0E-02$. This may be due to the sparsely distributed noisy data points and the LOO-CV test.

Fig. 10 illustrates the histogram of the posterior PDF of C_s (obtained using the MH algorithm), as well as profiles of log marginal likelihood and maximums in FIPO-BC cases and STD-BC are plotted. As the inducing points refined, the shape of $\log(p(\mathbf{M}_s | C_s))$ becomes more like that of $\log(p(\mathbf{d} | C_s))$. Since the effect of hyperparameters is not marginalised, the range of the marginal likelihood greatly varies and causes differences in the posterior PDF. For instance, the ranges of $\log(p(\mathbf{M}_s | C_s))$ in FIPO-BC-2 and FIPO-BC-4 are much larger than that in the other cases. Subsequently, the posterior distribution of C_s in these three cases is more concentrated than others. The maximum values in the log marginal likelihood of all cases are 1.0, 0.83, 0.89, 0.92, 0.92, and 0.9 for FIPO-BC-1, 2, 3, 4, 5 and STD-BC respectively. Since the SAS-SST prediction using $C_s \in [0.9, 1.0]$ can be deemed to be similar due to the very small MSD between them. It is reasonable to claim the inducing points convergence is achieved at FIPO-BC-4. Fig. 11 illustrates the computing efficiency (represented by CPU time(s) per 1000 MH sampling) of different cases. As expected, the time spent on MH sampling greatly reduces in all FIPO-BC cases. Even for FIPO-BC-5 (with finest SVI grid), CPU time(s) per 1000 MH sampling is less than a tenth of that of the STD-BC.

To demonstrate the online FIPO-BC, the original data is split into four different data batches and fed into the FIPO-BC solver sequentially, whilst the inducing is the same as that used in FIPO-BC-5. The results of this online calibration are shown in Fig. 12. Each pane of Fig. 12 includes the C_s used to generate the numerical outputs, response surface of absolute error ($|\text{numerical output} - \text{SVGP prediction}|$), as well as the optimal C_s . Clearly, with more data supplied to the FIPO-BC

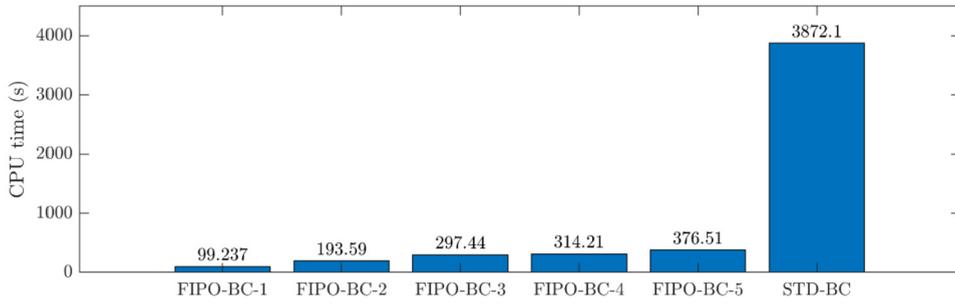


Fig. 11. CPU time(s) per 1000 MH sampling in each case for exploring posterior distribution of C_s .

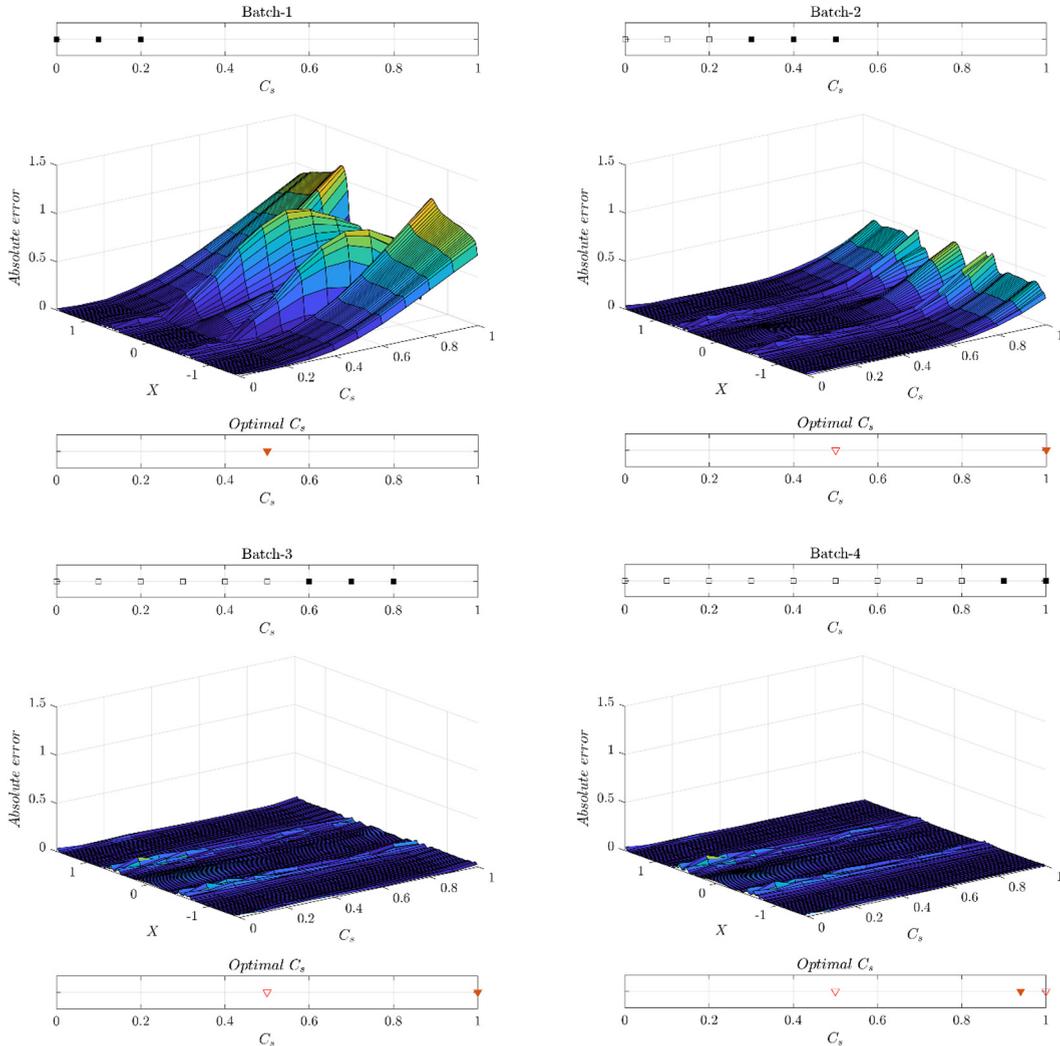


Fig. 12. The online calibration of C_s in SAS-SST model. Each pane shows the response surface of the inducing output for the models as well as the calibrated C_s after a batch of model output using different C_s , marked as solid points. The C_s used to generate model outputs considered in the previous step (but discarded the current step) and previously obtained calibrated value are marked as empty points.

solver, the absolute error is reduced across the parameter space. Meanwhile, the optimal C_s continuously improves. The optimal C_s obtained using 'Batch-1, 2, 3 & 4' are 0.5, 1.0, 1.0, and 0.94, respectively. It worth reiterating that the model outputs using any C_s between 0.9 and 1.0 are deemed to be the same because of ignorable differences. In particular, the absolute errors produced at the stage of 'Batch-3' and 'Batch-4' are very similar. It can be claimed with confidence that the calibration and posterior PDF approximation can be achieved only using Batch-1, 2 & 3, which means fewer CFD simulations are actually needed.

6. Summary and discussion

We have proposed a novel fixed inducing points online Bayesian calibration framework (FIPO-BC) algorithm for improving the accuracy of numerical simulations of science and engineering problems. The framework improves the computational efficiency and enables the online learning capability of the STD-BC algorithm by incorporating the SVI-GP. In term of the same training data, the FIPO-BC algorithm can use far fewer data points, resulting in a much smaller covariance matrix, and hence producing a much more computationally efficient algorithm. In our demonstration cases, the FIPO-BC is at least 10 times faster than the STD-BC. The new approach also allows online updates of the calibrated model parameter with data being supplied to the solver sequentially. The procedure of using the FIPO-BC algorithm is demonstrated by using it first to calibrate the parameter in a simple mathematical function, and then to calibrate an essential parameter of scale-resolving CFD modelling method, SAS-SST. The performance of the FIPO-BC algorithm is compared against the STD-BC algorithm. It has been demonstrated in both test cases that the FIPO-BC algorithm provides the same calibrated parameter as the STD-BC algorithm, once the FIPO-BC algorithm is converged on the number of inducing points. Due to the online learning capability of the FIPO-BC algorithm, the calibration can be achieved using less data.

Like the STD-BC algorithm, the posterior distribution of the parameter will be affected by the optimal hyperparameters. Both the STD-BC and FIPO-BC algorithms require marginalisation to eliminate this effect. In order to enable the algorithms to utilise marginalisation requires additional computational resources. However, the inclusion of marginalisation within the FIPO-BC algorithm would be more computationally efficient than the STD-BC algorithm with marginalisation. The FIPO-BC algorithm with marginalisation will be investigated in future work. By using the inducing points with the SVI-GP model, the FIPO-BC algorithm can handle much more training data compared to the STD-BC algorithm. It is possible to use the FIPO-BC algorithm in large data set applications if the dataset can be represented by a few thousand inducing data points. Extra research work is required to make the current algorithm more generic for large data sets. Moreover, the current framework requires prior knowledge of the possible range of the 'true' model parameter. However, the true model parameter may be located outside the prior defined range as more measurements revealed. In such a case, the FIPO-BC algorithm can only generate the best model parameter within the predefined range. One of the solutions is to allow the inducing points to be expanded with more measurements. Future research work has been planned to resolve this issue.

CRedit authorship contribution statement

Yu Duan: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Matthew D. Eaton:** Funding acquisition, Project administration, Writing – review & editing. **Michael J. Bluck:** Funding acquisition, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data statement

Some of the supporting data used to create figures in this paper can be accessed at the following URL: <https://doi.org/10.5281/zenodo.4562382>.

Acknowledgements

The authors would like to thank Rolls-Royce Civil Nuclear for their financial support via the funding of the Rolls-Royce Nuclear Modelling and Analytics Group which is hosted by the nuclear engineering group within the department of mechanical engineering at Imperial College London (project grant: P65165_MECM). In addition, the authors would like to thank the anonymous reviewers for their suggestions.

References

- [1] W.L. Oberkampf, M.F. Barone, Measures of agreement between computation and experiment: validation metrics, *J. Comput. Phys.* 217 (2006) 5–36, <https://doi.org/10.1016/j.jcp.2006.03.037>.
- [2] R. Rebba, S. Mahadevan, Validation of models with multivariate output, *Reliab. Eng. Syst. Saf.* 91 (2006) 861–871, <https://doi.org/10.1016/j.ress.2005.09.004>.
- [3] W.L. Oberkampf, T.G. Trucano, Verification and validation benchmarks, *Nucl. Eng. Des.* 238 (2008) 716–743, <https://doi.org/10.1016/j.nucengdes.2007.02.032>.
- [4] Y. Duan, C. Cooling, J. Soo, C. Jackson, A. Flint, M.D. Eaton, M.J. Bluck, Using a Gaussian process regression inspired method to measure agreement between the experiment and CFD simulations, *Int. J. Heat Fluid Flow* 80 (2019) 108497, <https://doi.org/10.1016/j.ijheatfluidflow.2019.108497>.
- [5] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, *J. R. Stat. Soc., Ser. B, Stat. Methodol.* 63 (2001) 425–464, <https://doi.org/10.1111/1467-9868.00294>.

- [6] M.J. Bayarri, J.O. Berger, R. Paulo, J. Sacks, J.A. Cafeo, C. Lin, J. Tu, M.J. Bayarri, J.O. Berger, R. Paulo, J. Sacks, J.A. Cafeo, A framework for validation of computer models, *Technometrics* 49 (2007) 138–153, <https://doi.org/10.1198/004017007000000092>.
- [7] M.J. Bayarri, J.O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R.J. Parthasarathy, R. Paulo, J. Sacks, D. Walsh, Computer model validation with functional output, *Ann. Stat.* 35 (2007) 1874–1906, <https://doi.org/10.1214/009053607000000163>.
- [8] M.J. Bayarri, J.O. Berger, M.C. Kennedy, A. Kottas, R. Paulo, J. Sacks, J.A. Cafeo, C.-H. Lin, J. Tu, Predicting vehicle crashworthiness: validation of computer models for functional and hierarchical data, *J. Am. Stat. Assoc.* 104 (2009) 929–943, <https://doi.org/10.1198/jasa.2009.ap06623>.
- [9] L.M.-E. Nina Glover, Serge Guillas, Statistical calibration of CFD modelling for street canyon flows, in: 12th Conf. Int. Build. Perform. Simul. Assoc., Sydney, 2011, pp. 14–16.
- [10] S. Guillas, N. Glover, L. Malki-Epshtein, Bayesian calibration of the constants of the $k-\epsilon$ turbulence model for a CFD model of street canyon flow, *Comput. Methods Appl. Mech. Eng.* 279 (2014) 536–553, <https://doi.org/10.1016/j.cma.2014.06.008>.
- [11] S.H. Cheung, T.A. Oliver, E.E. Prudencio, S. Prudhomme, R.D. Moser, Bayesian uncertainty analysis with applications to turbulence modeling, *Reliab. Eng. Syst. Saf.* 96 (2011) 1137–1149, <https://doi.org/10.1016/j.res.2010.09.013>.
- [12] T.A. Oliver, R.D. Moser, Bayesian uncertainty quantification applied to RANS turbulence models, *J. Phys. Conf. Ser.* 318 (2011) 1–10, <https://doi.org/10.1088/1742-6596/318/4/042032>.
- [13] W.N. Edeling, P. Cinnella, R.P. Dwight, Predictive RANS simulations via Bayesian model-scenario averaging, *J. Comput. Phys.* 275 (2014) 65–91, <https://doi.org/10.1016/j.jcp.2014.06.052>.
- [14] W.N. Edeling, M. Schmelzer, R.P. Dwight, P. Cinnella, Bayesian predictions of Reynolds-averaged Navier–Stokes uncertainties using maximum a posteriori estimates, *AIAA J.* 56 (2018) 2018–2029, <https://doi.org/10.2514/1.j056287>.
- [15] J.L. Wu, J.X. Wang, H. Xiao, A Bayesian calibration–prediction method for reducing model-form uncertainties with application in RANS simulations, *Flow Turbul. Combust.* 97 (2016) 761–786, <https://doi.org/10.1007/s10494-016-9725-6>.
- [16] J. Zhang, S. Fu, An efficient Bayesian uncertainty quantification approach with application to $k-\omega-\gamma$ transition modeling, *Comput. Fluids* 161 (2018) 211–224.
- [17] J. Zhang, S. Fu, An efficient approach for quantifying parameter uncertainty in the SST turbulence model, *Comput. Fluids* 181 (2019) 173–187, <https://doi.org/10.1016/j.compfluid.2019.01.017>.
- [18] A.F. Cortesi, P.G. Constantine, T.E. Magin, P.M. Congedo, Forward and backward uncertainty quantification with active subspaces: application to hypersonic flows around a cylinder, *J. Comput. Phys.* 407 (2020) 109079, <https://doi.org/10.1016/j.jcp.2019.109079>.
- [19] R.B. Gramacy, H.K.H. Lee, Experiments adaptive design and analysis of supercomputer experiments, *Technometrics* 51 (2009) 130–145, <https://doi.org/10.1198/TECH.2009.0015>.
- [20] X. Wu, T. Kozłowski, H. Meidani, K. Shirvan, Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, part 2: application to TRACE, *Nucl. Eng. Des.* 335 (2018) 417–431.
- [21] X. Wu, T. Kozłowski, H. Meidani, K. Shirvan, Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, part 1: theory, *Nucl. Eng. Des.* 335 (2018) 339–355, <https://doi.org/10.1016/j.nucengdes.2018.06.004>.
- [22] Y. Liu, X. Sun, N.T. Dinh, Validation and uncertainty quantification of multiphase-CFD solvers: a data-driven Bayesian framework supported by high-resolution experiments, *Nucl. Eng. Des.* 354 (2019) 110200, <https://doi.org/10.1016/j.nucengdes.2019.110200>.
- [23] X. Wu, K. Shirvan, T. Kozłowski, Demonstration of the relationship between sensitivity and identifiability for inverse uncertainty quantification, *J. Comput. Phys.* 396 (2019) 12–30, <https://doi.org/10.1016/j.jcp.2019.06.032>.
- [24] G. Karagiannis, G. Lin, On the Bayesian calibration of computer model mixtures through experimental data, and the design of predictive models, *J. Comput. Phys.* 342 (2017) 139–160, <https://doi.org/10.1016/j.jcp.2017.04.003>.
- [25] K.-L. Chang, S. Guillas, Computer model calibration with large non-stationary spatial outputs application to the calibration of a climate model, *Appl. Stat. Ser. C* 68 (2019) 51–78.
- [26] D. Higdon, C. Nakhleh, J. Gattiker, B. Williams, A Bayesian calibration approach to the thermal problem, *Comput. Methods Appl. Mech. Eng.* 197 (2008) 2431–2441, <https://doi.org/10.1016/j.cma.2007.05.031>.
- [27] D. Higdon, J. Gattiker, B. Williams, M. Rightley, D. Higdon, J. Gattiker, B. Williams, M. Rightley, D.H. Igdon, J.G. Attiker, B.W. Iliams, M.R. Ightley, Computer model calibration using high-dimensional output high-dimensional output, *J. Am. Stat. Assoc.* 103 (2008) 570–583, <https://doi.org/10.1198/016214507000000888>.
- [28] H. Liu, Y. Ong, X. Shen, J. Cai, S. Member, When Gaussian process meets big data: a review of scalable GPs, *IEEE Trans. Neural Netw. Learn. Syst.* (2020) 1–19, <https://doi.org/10.1109/TNNLS.2019.2957109>.
- [29] C.K.I. Williams, M. Seeger, The using Nyström method to speed up kernel machines, in: *Adv. Neural Inf. Process. Syst.* 13 (NIPS 2000), 2000, pp. 1–7.
- [30] A. Gittens, M.W. Mahoney, Revisiting the Nyström method for improved large-scale machine learning, *J. Mach. Learn. Res.* 17 (2016) 1–65.
- [31] M. Seeger, C. Williams, N. Lawrence, Fast forward selection to speed up sparse Gaussian process regression, in: *Artif. Intell. Stat.* 9, 2003, pp. 1–8.
- [32] A.G. Wilson, Hannes Nicki, Kernel interpolation for scalable structured Gaussian processes (KISS-GP), in: 32nd Int. Conf. Mach. Learn., Lille, France, 2015, pp. 1–10.
- [33] J. Hensman, N. Fusi, N.D. Lawrence, Gaussian processes for big data, in: *Conf. Uncertain. Artif. Intell.*, 2013, pp. 282–290.
- [34] C.E. Rasmussen, C.K.I. Williams, Gaussian processes for machine learning, <http://www.gaussianprocess.org/gpml/chapters/RW.pdf>, 2006.
- [35] S. Roberts, M. Osborne, M. Ebdon, S. Reece, N. Gibson, S. Aigrain, Gaussian processes for time-series modelling, *Philos. Trans. R. Soc. A* 371 (2013).
- [36] A. Damianou, Deep Gaussian processes and variational propagation of uncertainty, 2015.
- [37] M.K. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: 12th Int. Conf. Artif. Intell. Stat., Clearwater Beach, Florida, USA, 2009, pp. 567–574.
- [38] F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, *Comput. Optim. Appl.* 51 (2012) 259–277, <https://doi.org/10.1007/s10589-010-9329-3>.
- [39] F.R. Menter, Y. Egorov, The scale-adaptive simulation method for unsteady turbulent flow predictions. Part 1: theory and model description, *Flow Turbul. Combust.* 85 (2010) 113–138, <https://doi.org/10.1007/s10494-010-9264-5>.
- [40] Y. Egorov, F.R. Menter, R. Lechner, D. Cokljat, The scale-adaptive simulation method for unsteady turbulent flow predictions. Part 2: application to complex flows, *Flow Turbul. Combust.* 85 (2010) 139–165, <https://doi.org/10.1007/s10494-010-9265-4>.
- [41] A. Sjunnesson, C. Nelsson, E. Max, LDA measurements of velocities and turbulence in a bluff body stabilized flame, in: 4th Int. Conf. Laser Anemometry – Adv. Appl., Cleveland, OH, 1991.
- [42] A. Sjunnesson, P. Henrikson, C. Lofstrom, CARS measurements and visualization of reacting flows in a bluff body stabilized flame, in: 28th Jt. Propuls. Conf. Exhib., Nashville, Tennessee, The USA, 1992, pp. 1–10.