Journal of Computational Physics 342 (2017) 139–160

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

On the Bayesian calibration of computer model mixtures through experimental data, and the design of predictive models

Georgios Karagiannis [a], Guang Lin [b,*]

[a] Department of Mathematical Sciences, Durham University, Stockton Road, Durham, DH1 3LE, UK
[b] Department of Mathematics, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 11 May 2016
Received in revised form 30 January 2017
Accepted 1 April 2017
Available online 25 April 2017

Keywords:
Uncertainty quantification
Computer experiments
Gaussian process
Polynomial bases
Multinomial logistic model
MCMC

ABSTRACT

For many real systems, several computer models may exist with different physics and predictive abilities. To achieve more accurate simulations/predictions, it is desirable for these models to be properly combined and calibrated. We propose the Bayesian calibration of computer model mixture method which relies on the idea of representing the real system output as a mixture of the available computer model outputs with unknown input dependent weight functions. The method builds a fully Bayesian predictive model as an emulator for the real system output by combining, weighting, and calibrating the available models in the Bayesian framework. Moreover, it fits a mixture of calibrated computer models that can be used by the domain scientist as a mean to combine the available computer models, in a flexible and principled manner, and perform reliable simulations. It can address realistic cases where one model may be more accurate than the others at different input values because the mixture weights, indicating the contribution of each model, are functions of the input. Inference on the calibration parameters can consider multiple computer models associated with different physics. The method does not require knowledge of the fidelity order of the models. We provide a technique able to mitigate the computational overhead due to the consideration of multiple computer models that is suitable to the mixture model framework. We implement the proposed method in a real-world application involving the Weather Research and Forecasting large-scale climate model.

1. Introduction

Computer experiments often use computer models (simulators) to simulate the behavior of a complex real system under consideration. These models are usually designed according to theories believed to govern the real system. They usually include calibration parameters, that is unknown parameters that regulate the behavior of the computer model; hence we wish to tune (calibrate) them in order for the computer model to represent the real system accurately. Often, calibration of a computer model is performed in the presence of experimental data in order to find optimal values for the unknown

* Corresponding author.
E-mail addresses: georgios.karagiannis@durham.ac.uk, georgios.stats@gmail.com (G. Karagiannis), guanglin@purdue.edu (G. Lin).

http://dx.doi.org/10.1016/j.jcp.2017.04.003
0021-9991/

calibration parameters. In cases that the computer models are expensive to run, there is interest in building inexpensive predictive statistical models.

Kennedy and O'Hagan [1] proposed an effective Bayesian computer model calibration to address such cases. Briefly, the experimental observations are represented as a sum of three functional terms: the computer model output, a systematic discrepancy, and an observational error. These functional terms are modeled as Gaussian processes [1–4], because computer models are often computationally expensive, and available training data are limited. Literature includes several variations of computer model calibration which can handle different issues; e.g. discontinuity/non-stationarity in the outputs [5], discrete inputs [6], calibration in the frequentest context [7], high-dimensional outputs [8], dynamic discrepancy [9], large number of inputs and outputs [10], etc. However, these works are restricted in cases where a single computer model is available. Nowadays, there is a plethora of computer models that aim at simulating the same real system. These models may differ either in precision (multi-fidelity case) of the solvers involved, or in the theories based on which they are designed (multi-physics case). Recently, Goh et al. [11] proposed a procedure to perform Bayesian calibration of computer models available at different levels of fidelity. It combines the models in a nested structure according to a given fidelity order. However, this approach is restricted to address only multi-fidelity cases where the fidelity order of the computer models is known.

Often, there are available several computer models, based on different theories, that represent the same real system. Each single computer model may have its own unique properties and predictive capabilities in representing the real system. Therefore, there is not a commonly acceptable way to order such models. Possible reasons for example can be: (i) incomplete knowledge of the complex real system, (ii) different computational capabilities of research groups, (iii) different scientific theories or perspectives describing the same real system, etc. In such cases, using only a single computer model may lead to misleading inferences and predictions and ignore the physics considered by other computer models only. Furthermore, traditional multi-fidelity calibration methods, such as [11], are not suitable to address such cases because the fidelity order of the models is not available a priori, or because nesting one model to another could possibly impose unrealistic relations among the models. Moreover, in the presence of moderately large number of models, the direct implementation of standard multi-fidelity calibration method becomes very expensive. Here, the question of interest is how to properly combine and calibrate such computer models in order to represent the real system output accurately.

In this study, the motivation for addressing the aforesaid problem raises from the Weather Research and Forecasting (WRF) regional climate model [12]. WRF allows for different configurations (sub-models), e.g. different parametrization suits, physics schemes, or resolutions, which in principle can constitute different models. Briefly, here the available computer models consist of different combinations of radiation schemes (the Rapid Radiative Transfer Model for General Circulation Models [13], and the Community Atmosphere Model 3.0 [14]) that describe different physics, and different resolutions (25 km and 50 km grid spacing) that describe different fidelity levels. It is uncertain which radiation scheme leads to better simulations. Moreover, higher grid spacing does not necessarily lead to more accurate simulations because WRF is sensitive to other physical parametrizations which is uncertain how they are affected by the grid spacing. Combination of physics variability is expected to result better predictions in climate models [15]; hence interest lies in combining suitably these computer models in order to integrate the associated physics and fidelity variations. WRF is employed with the Kain Fritsch (KF) convective parametrization scheme (CPS) [16]. For climate models, it is important to better understand and constrain the convective parametrization, and hence interest lies in quantifying and reducing the uncertainties regarding of those parameters. The computational cost of running WRF is prohibitively high, and an exhausted direct simulation study is not possible in practice; hence there is interest in a predictive model.

In this article, we propose the Bayesian calibration of computer model mixture method, as an extension to the traditional Bayesian (single) model calibration [1,2]. Central to the proposed methodology is the idea of (i) representing the output function of the complex real system as a mixture of output functions of the available computer models with unknown input dependent weight functions, and (ii) specifying a fully Bayesian model to quantify the associated uncertainties. The proposed method allows one to build a predictive model (emulator) for the output of a real system by properly calibrating, weighting, and combining the available computer models in the Bayesian framework. Additionally, it allows the design of a calibrated mixture of computer models (simulators) by evaluating the associated weight functions and the calibration parameters. The resulting computer model mixture, as well as the predictive model, aims at representing the real system output more accurately than the single ones by aggregating the unique features of different models. We introduce the concept of shared calibration parameters that allows inference on calibration parameters to be based on multiple computer models (and hence different physics), however, the method allows different models to have different calibration parameters. The Bayesian computations are performed via Markov chain Monte Carlo methods. A computational highlight of the procedure is that it builds the unknown mixture weight functions via a stochastic bases selection from a pool of basis functions in a data-driven manner.

The method is suitable to address realistic problems that one model may be more accurate than the other at different (unspecified) input sub-regions. In particular, through the weight functions, it allows the determination of the input sub-region at which a individual computer model is more preferable to be used than the rest individual ones. The method is particularly suitable to address applications where the outputs of the available computer models tend to differ from the output of the real system at different directions. This is because the weight functions can adjust the outputs of contributing models in the mixture, in a manner that the overall discrepancy of the mixture will be less that the individual ones. There-

fore, in such cases, the resulting calibrated computer model mixture is able to produce more accurate simulations than the single ones. This covers a large range of important real-world applications [15], such as the WRF one analyzed here.

The article is organized as follows. In Section 2, we present the proposed method. In Section 3.1, we validate the proposed method with that of Goh et al. [11] in a validation example. In Section 3.2, we assess the good performance of the method and compare it with that of Kennedy and O'Hagan [1] in a more challenging benchmark example. In Section 3.3, we implement the method on a real-world large-scale climate modeling application that involves the WRF with the KF CPS. In Section 4, we conclude and propose possible extensions. In Appendix B, we provide a technique that mitigates the computational overhead of the procedure which is caused by the consideration of multiple computer models.

2. The method

The proposed method extends the standard Bayesian calibration of a single computer model [1,2] to the multiple computer model framework.

2.1. Basic formulation

Set-up We assume there is available a set of K different computer models $\{\mathscr{S}^{(k)};\ k \in \mathcal{K}\}$ where $\mathcal{K} = \{1, ..., K\}$. Each computer model $\mathscr{S}^{(k)}$ aims at simulating the same real system $\mathscr{Z}$.

We consider training data which consist of a collection of experimental data $\{(y_i, x_i);\ i = 1, ..., n\}$ generated from the real system $\mathscr{Z}$ after n realizations, and K designs of simulated data $\{(\eta_i^{(k)}, x_i^{(k)}, t_i^{(k)});\ i = n + \sum_{j<k} m^{(j)} + 1, ..., n + \sum_{j\le k} m^{(j)}\}$ generated from the computer model $\mathscr{S}^{(k)}$ after $m^{(k)}$ runs for $k = 1, ..., K$. Let $z^{\otimes K} = (y^\intercal, \eta^{(1),\intercal}, ..., \eta^{(K),\intercal})$ denotes the complete training data outputs, and $n^{\otimes K} = n + \sum_{k=1}^{K} m^{(k)}$ denotes the size of $z^{\otimes K}$. Here, $y_i \in \mathbb{R}$, $x_i \in \mathcal{X}$, $x_i^{(k)} \in \mathcal{X}$, and $t_i^{(k)} \in \Theta^{(k)}$, for any i, k, where $\mathcal{X}$ is the input domain, and $\Theta^{(k)}$ is the calibration parameter domain of computer model $\mathscr{S}^{(k)}$.

Regarding the real system $\mathscr{Z}$, the experimental observations $y_i := y(x_i)$ are generated for a given x_i via

$$y_i = \zeta(x_i) + \epsilon_{y,i},$$

where the $\epsilon_{y,i}$ denotes the observation error, and $\zeta(x_i)$ denotes the expected output of the real system at input x_i for $i = 1, ..., n$.

Regarding the computer model $\mathscr{S}^{(k)}$, the simulated data $\eta_i^{(k)} := \eta^{(k)}(x_i^{(k)}, t_i^{(k)})$ are generated for a given $x_i^{(k)}$, and calibration parameters $t_i^{(k)}$ via

$$\eta_i^{(k)} = S^{(k)}(x_i^{(k)}, t_i^{(k)}) + \epsilon_{\eta,i}^{(k)},$$

where $\epsilon_{\eta,i}^{(k)}$ denotes the random error, and $S^{(k)}(x_i^{(k)}, t_i^{(k)})$ denotes the expected output of the computer model $\mathscr{S}^{(k)}$ at $(x_i^{(k)}, t_i^{(k)})$, for $i = n + \sum_{j<k} m^{(j)} + 1, ..., n + \sum_{j\le k} m^{(j)}$. The inclusion of term $\epsilon_\eta^{(k)}$ as random error is necessary when $\mathscr{S}^{(k)}$ is stochastic, as well as beneficial, in terms of the stability of the statistical model, when $\mathscr{S}^{(k)}$ is deterministic as discussed by Gramacy and Lee [17].

Computer model mixture Although each computer model aims at simulating the same real system, it may be designed based on different theoretical background and present different properties. In order to aggregate different properties associated with different computer models, we model the output function of the real system $\mathscr{Z}$ as a mixture of the output functions of the available computer models plus a discrepancy. We define the computer model mixture representation of the real system, as

$$S^{\otimes K}(\cdot, \theta^{\otimes K}) = \sum_{k=1}^{K} \varpi_k(\cdot) S^{(k)}(\cdot, \theta^{(k)}); \qquad \zeta(\cdot) = S^{\otimes K}(\cdot, \theta^{\otimes K}) + \delta(\cdot). \tag{2.1}$$

Note that in our framework, the components of mixture (2.1) are computer models (simulators), unlike other works [18–20] in the literature where the components are different statistical models referring to the same computer model.

The main role of the weight functions $\{\varpi_k(\cdot)\}$ is to adjust the contribution of the corresponding computer models $\{S^{(k)}\}$ in the mixture $\{S^{\otimes K}\}$ as functions of the input in a principled manner; for this reason we consider them as a probability vector that depends on the inputs. Precisely, $\varpi(\cdot) := (\varpi_k(\cdot); k \in 1, ..., K)$ is the unknown vector of weight functions such that $\{\varpi_k(\cdot) : \mathcal{X} \to (0, 1);\ k = 1, ..., K-1\}$, and $\varpi_K(\cdot) : \mathcal{X} \to (0, 1)$ with $\varpi_K(\cdot) = 1 - \sum_{k=1}^{K-1} \varpi_k(\cdot)$. This allows differently weighted combinations of the computer models to represent the real system at different inputs. Hence, (2.1) is suitable to model realistic problems where the unknown fidelity order of the computer models may change over the input space $\mathcal{X}$. Moreover, $\delta(\cdot)$ is a discrepancy function, and it refers to a potential systematic disagreement between the real system output $\zeta(\cdot)$ and the computer model mixture output $S^{\otimes K}(\cdot, \theta^{\otimes K})$, e.g. due to 'missed' or 'missrepresented' physical properties. We highlight that different computer models $\{\mathscr{S}^{(k)}\}$ may have calibration parameters different in value, or dimensionality. As a result, there is need to evaluate the set of calibration parameters $\{\theta^{(k)} \in \Theta^{(k)}; k \in \mathcal{K}\}$, (or $\theta^{\otimes K} = (\theta^{(1)}, ..., \theta^{(K)})$).

The computer model mixture calibration problem can be summarized as

$$\mathscr{C}^{\otimes K}: \begin{cases} y(x_i) = \sum_{k=1}^{K} \varpi_k(x_i) S^{(k)}(x_i, \theta^{(k)}) + \delta(x_i) + \epsilon_{y,i}, & i = 1, ..., n \\ \eta_i^{(1)} = S^{(1)}(x_i^{(1)}, t_i^{(1)}) + \epsilon_{\eta,i}^{(1)}, & i = n+1, ..., n+m^{(1)} \\ \vdots & \vdots \\ \eta_i^{(K)} = S^{(K)}(x_i^{(K)}, t_i^{(K)}) + \epsilon_{\eta,i}^{(K)}, & i = n+\sum_{k<K} m^{(k)}+1, ..., n^{\otimes K} \end{cases}. \tag{2.2}$$

Weight functions parametrization The unknown weight functions $\{\varpi_k(\cdot)\}$ are modeled a polynomial expansion in the multivariate logistic space, as

$$\log(\frac{\varpi_k(\cdot)}{\varpi_K(\cdot)}) = g_k(\cdot); \tag{2.3}$$

$$g_k(\cdot) = \sum_{a \in \Lambda_{p_\varpi, d_x}} h_{\varpi,a}^{(k)}(\cdot)\omega_{k,a}, \tag{2.4}$$

for $k = 1, ..., K-1$, where $\varpi_K(\cdot) = 1 - \sum_{k=1}^{K-1} \varpi_k(\cdot)$, and $g_k(\cdot)$ is a polynomial expansion of degree p_ϖ. Specifically, $\{h_{\varpi,a}^{(k)}(\cdot)\}$ are multi-dimensional basis functions properly specified, for example from Askey family [21], $\{\omega_{k,a}\}$ are unknown coefficients, and Λ_{p_ϖ, d_x} is a set of multi-indices indicating the available bases up to a degree p_ϖ. The rational in (2.3) is that the additive logistic transformation is suitable to provide a monotonic mapping between $\mathbb{R}^{K-1}$ and the simplex of K-dimensional probability vector. Moreover, regarding (2.4), the polynomial expansions are able to accurately represent unknown functions under certain regularity conditions [22].

Often, only a subset $\mathcal{I}_k \in \Lambda_{p,d_x}$ of bases significantly contributes to the expansion (2.4), while the rest bases can be omitted without serious loss of accuracy [23]. Here, we consider that given a set of available bases Λ_{p,d_x}, there is an unknown subset of significant bases with indices $\mathcal{I}_k \subset \Lambda_{p,d_x}$, for $k = 1, ..., K$. Therefore, given (2.4) using only the subset of bases with indices $\mathcal{I}_k \subset \Lambda_{p,d_x}$, the unknown weight functions result from the inversion of (2.3) as

$$\varpi_k(\cdot) = \frac{\exp(h_{\varpi,\mathcal{I}_k}^{(k)}(\cdot)^\intercal \omega_{k,\mathcal{I}_k})}{1 + \sum_{j=1}^{K-1} \exp(h_{\varpi,\mathcal{I}_k}^{(k)}(\cdot)^\intercal \omega_{j,\mathcal{I}_k})}, \tag{2.5}$$

where $h_{\varpi,\mathcal{I}_k}^{(k)}(\cdot) := (h_{\varpi,a}^{(k)}(\cdot); a \in \mathcal{I}_k)$ and $\omega_{k,\mathcal{I}_k} := (\omega_{k,a}; a \in \mathcal{I}_k)$ for $k = 1, ..., K-1$, and $\varpi_K(\cdot) = 1 - \sum_{k=1}^{K-1} \varpi_k(\cdot)$. The consideration of smaller sets of bases $\{h_{\varpi,a}^{(k)}(\cdot); a \in \mathcal{I}_k\}$ may have computational benefits as it reduces the number of the unknown coefficients $\{\omega_{k,a}; a \in \mathcal{I}_k\}$ to be estimated [23–26]. Parametrization (2.5) leads to convenient computations, as well as reliable inferences and predictions. In the current framework, the use of Gaussian process priors [20] or an allocation model [27] for the representation of the weight functions would lead to expensive computations due to the introduction of many extra latent/nuisance variables. The special case where the weights are assumed to be constant values $\{h_{\varpi,\mathcal{I}_k}^{(k)}(\cdot) = 1\}$ implies that the fidelity of the computer models is constant across the input space, and hence it can be too restrictive in real-world problems.

Surrogate modeling We consider the realistic scenario where the available computer models $\{\mathscr{S}^{(k)}\}$ are computationally expensive, and hence they cannot be used directly in Bayesian computations that require a vast number of direct computer runs. The 'uncertain' functions $\{S^{(k)}(\cdot,\cdot)\}$ and $\delta(\cdot)$ are modeled as Gaussian processes (GP) [1,4], that allow the design of an emulator for the output function in a mathematically convenient manner. For $k \in \mathcal{K}$, we assign independent Gaussian processes priors on the functions $S^{(k)}(\cdot,\cdot)$ and $\delta(\cdot)$ as $S^{(k)}(\cdot,\cdot) \sim \mathrm{GP}(\mu_S^{(k)}(\cdot,\cdot), c_S^{(k)}((\cdot,\cdot),(\cdot,\cdot)))$, and $\delta(\cdot) \sim \mathrm{GP}(\mu_\delta(\cdot), c_\delta(\cdot,\cdot))$, where $\mu_S^{(k)}: \mathcal{X} \times \Theta^{(k)} \to \mathbb{R}$, $\mu_\delta: \mathcal{X} \to \mathbb{R}$ are the mean functions of the GPs, and $c_S^{(k)}: \mathcal{X} \times \Theta^{(k)} \times \mathcal{X} \times \Theta^{(k)} \to \mathbb{R}^+$, $c_\delta: \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ are the covariance functions of the GPs. For presentation purpose, we consider a traditional parametrization for $\mu_S^{(k)}(\cdot,\cdot)$, $\mu_\delta(\cdot)$, $c_S^{(k)}((\cdot,\cdot),(\cdot,\cdot))$, and $c_\delta(\cdot,\cdot)$, however more intricate ones can be used in our framework. The mean functions are specified as linear expansions $\mu_\delta(\cdot) = h_\delta(\cdot)^\intercal \beta_\delta$ and $\mu_S^{(k)}(\cdot,\cdot) = h_S^{(k)}(\cdot,\cdot)\beta_S^{(k)}$ where $h_\delta: \mathcal{X} \to \mathbb{R}^{d_{\beta,\delta}}$ and $h_S^{(k)}: \mathcal{X} \times \Theta^{(k)} \to \mathbb{R}^{d_{\beta,S}^{(k)}}$ are vectors of basis functions, such as polynomial bases [21], or wavelets [28], and β_δ, $\beta_S^{(k)}$ are vectors of unknown coefficients with $\beta_S^{(k)} \in \mathbb{R}^{d_{\beta,S}^{(k)}}$, $\beta_\delta \in \mathbb{R}^{d_{\beta,\delta}}$. The covariance functions can be specified according to the separable covariance function family [29,30] as

$$c_S^{(k)}((x,t),(x',t')) = \tau_S^{(k)} \prod_{l=1}^{q} (\phi_{S,x,l}^{(k)})^{(2|x_l - x_l'|)^2} \prod_{l=1}^{p^{(k)}} (\phi_{S,t,l}^{(k)})^{(2|t_l - t_l'|)^2}); \tag{2.6}$$

$$c_\delta(x,x') = \tau_\delta \prod_{l=1}^{q} (\phi_{\delta,x,l})^{(2|x_l - x_l'|)^2}, \tag{2.7}$$

where $\tau_S^{(k)} > 0$, $\tau_\delta > 0$ control the marginal variances; $\{\phi_{S,x,l}^{(k)} \in (0,1)\}$, $\{\phi_{S,t,l}^{(k)} \in (0,1);\}$, $\{\phi_{\delta,x,l} \in (0,1)\}$ control the dependence strength in each of the component directions of x and t. More intricate covariance functions, such as the stationary ones from the Matérn family [31,4], the non-stationary ones of Paciorek and Schervish [32], or the compact support (combined via tapering) ones [33, Chapter 9] can also be used in this set-up. Here, ϵ_y and $\epsilon_\eta^{(k)}$ are modeled as random noises with unknown variances $\sigma_y^2 > 0$ and $\sigma_\eta^{(k),2} > 0$ respectively. We caution that some applications may require $\epsilon_y(\cdot)$ and $\{\epsilon_\eta^{(k)}(\cdot,\cdot)\}$ to be treated as functions; such a case is out of the scope of this article.

2.2. The Bayesian model

To facilitate the presentation we make the notation more compact, and define unknown random parameters: $\theta^{\otimes K} := (\theta^{(1)}, ..., \theta^{(K)})$ on space $\Theta^{\otimes K}$, $\beta^{\otimes K} := (\beta_S^{(1)}, ...\beta_S^{(K)}, \beta_\delta^{(k)})$ on space $\mathcal{B}^{\otimes K} := \mathbb{R}^{\sum_{k=1}^K d_{\beta,S}^{(k)} + d_{\beta,\delta}}$, $\varphi^{\otimes K} := ((\phi_{S,x}^{(k)}, \phi_{S,t}^{(k)}, \tau_S^{(k)}, \sigma_\eta^{(k),2}; k = 1, ..., K), \phi_{\delta,x}, \tau_\delta, \sigma_y^2)$ on space $\Phi^{\otimes K}$, and $\mathcal{I} = (\mathcal{I}_1, ..., \mathcal{I}_K)$.

Statistical model The (marginal) likelihood function of $z^{\otimes K}$, marginalized with respect to the GP priors of $\{S^{(k)}(\cdot)\}$ and $\delta(\cdot)$, is a multivariate Normal distribution with $n^{\otimes K}$-dimensional mean vector $\mu_z^{\otimes K} := \mu_z^{\otimes K}(\mathcal{I}, \varpi, \beta^{\otimes K}, \theta^{\otimes K})$ and covariance matrix $\Sigma_z^{\otimes K} := \Sigma_z^{\otimes K}(\mathcal{I}, \varpi, \varphi^{\otimes K}, \theta^{\otimes K})$ of size $n^{\otimes K} \times n^{\otimes K}$ such that

$$\mu_z^{\otimes K} = \overbrace{\begin{bmatrix} H_S^{(1)} & H_S^{(2)} & \cdots & H_S^{(K)} & H_\delta \\ \dot{H}_S^{(1)} & 0 & \cdots & 0 & 0 \\ 0 & \dot{H}_S^{(2)} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & \dot{H}_S^{(K)} & 0 \end{bmatrix}}^{H_z^{\otimes K} =} \overbrace{\begin{bmatrix} \beta_S^{(1)} \\ \beta_S^{(1)} \\ \vdots \\ \beta_S^{(K)} \\ \beta_\delta \end{bmatrix}}^{\beta^{\otimes K} =}; \qquad \Sigma_z^{\otimes K} = \begin{bmatrix} \Sigma_z & \Sigma_z^{(1),\intercal} & \Sigma_z^{(2),\intercal} & \cdots & \Sigma_z^{(K),\intercal} \\ \Sigma_z^{(1)} & \Sigma_z^{(1,1)} & 0 & \cdots & 0 \\ \Sigma_z^{(2)} & 0 & \Sigma_z^{(2,2)} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Sigma_z^{(K)} & 0 & \cdots & 0 & \Sigma_z^{(K,K)} \end{bmatrix},$$

respectively. Here, $\{[H_S^{(k)}]_{i,:} = \varpi_k(x_i) h_S^{(k),\intercal}(x_i, \theta^{(k)}); i = 1, ..., n, k = 1, ..., K\}$, $\{[H_\delta]_{i,:} = h_\delta^\intercal(x_i); i = 1, ..., n\}$, $\{[\dot{H}_S^{(k)}]_{i,:} = h_S^{(k),\intercal}(x_i, t_i^{(k)}); i = n + \sum_{j'<k} m^{(j')} + 1, ..., n + \sum_{j'\le k} m^{(j')}, k = 1, ..., K\}$, and

$$\begin{aligned}
[\Sigma_z]_{i,j} &= \sum_{k=1}^K \varpi_k(x_i)\varpi_k(x_j) c_S^{(k)}((x_i, \theta^{(k)}), (x_j, \theta^{(k)})) + c_\delta(x_i, x_j) + \sigma_y^2 \mathbb{1}_0(i-j), \\
&\quad i = 1, ..., n;\ j = 1, ..., n; \\
[\Sigma_z^{(k)}]_{i,j} &= \varpi_k(x_j) c_S^{(k)}((x_i, t_i^{(k)}), (x_j, \theta^{(k)})), \\
&\quad i = n + \sum_{j'<k} m^{(j')} + 1, ..., n + \sum_{j'\le k} m^{(j')};\ j = 1, ..., n; \\
[\Sigma_z^{(k,k)}]_{i,j} &= c_S^{(k)}((x_i, t_i^{(k)}), (x_j, t_j^{(k)})) + \sigma_\eta^{(k),2} \mathbb{1}_0(i-j), \\
&\quad i = n + \sum_{j'<k} m^{(j')} + 1, ..., n + \sum_{j'\le k} m^{(j')};\ j = n + \sum_{j'<k} m^{(j')} + 1, ..., n + \sum_{j'\le k} m^{(j')},
\end{aligned}$$

according to (2.6) and (2.7).

The proposed framework allows the introduction of shared calibration parameters, namely calibration parameters which are common to different computer models, have the same interpretation, and have the same values across different models. Different computer models, e.g. $\mathscr{S}^{(k)}$ and $\mathscr{S}^{(k')}$, may share a set of common calibration parameters, e.g. $\theta_j^{(k)}$ and $\theta_{j'}^{(k')}$, that describe the same quantity. In many cases, it is desirable for some calibration parameters to be calibrated jointly across different models. Technically, this can be achieved by setting appropriate constrains on the space $\Theta^{\otimes K}$, e.g. $\theta_j^{(k)} = \theta_{j'}^{(k')}$. This allows inference on those parameters to be based on multiple computer models and hence possibly different physics. In the context of computer model mixture (2.2), the weight functions control the 'contribution' of each computer model in the calibration procedure through (2.1). Therefore, the training of shared calibration parameters is primary influenced by computer models with larger weights and hence those which represent the system output more accurately.

Prior model We specify a prior model for the unknown parameters $\pi(\mathcal{I}, \omega, \beta^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K})$.

Regarding the weight functions, we assign priors on $\mathcal{I}_k \sim \Pr(\mathcal{I}_k)$ in order to account uncertainty about the unknown set of the significant bases functions, and Normal priors on $\omega_k \sim \mathrm{N}(b_\omega, \xi_\omega^{-1})$ in order to account uncertainty about the unknown coefficients, for $k = 1, ..., K-1$. A priori information, for example related to the fidelity of the computer models,

can be included in the prior model by adjusting the prior hyper-parameters. Otherwise, weakly informative priors of the weight functions parameters can be used; e.g. $b_\omega = 0$, ξ_ω small, and $\Pr(\mathcal{I}_k) \propto 1$ for $k = 1, ..., K-1$.

A priori independent priors can be assigned on $\{\beta^{\otimes K}, \varphi^{\otimes K}\}$ such that

$$\begin{array}{lll} \phi_{S,x,l}^{(k)} & \sim \text{Be}(a_{\phi,S,x}, b_{\phi,S,x}),\ l=1,...,q; & \beta_S^{(k)} \quad \sim \text{N}(b_S^{(k)}, \xi_\beta^{-1}\Sigma_{\beta,S}^{(k)}); \\ \phi_{S,t,l}^{(k)} & \sim \text{Be}(a_{\phi,S,x}, b_{\phi,S,x}),\ l=1,...,p^{(k)}; & \beta_\delta \quad \sim \text{N}(0, \xi_\beta^{-1}\Sigma_{\beta,\delta}); \\ \phi_{\delta,x,l} & \sim \text{Be}(a_{\phi,\delta,x}, b_{\phi,\delta,x}),\ l=1,...,q; & \\ \tau_S^{(k)} & \sim \text{IG}(a_{\tau,S}, b_{\tau,S}); & \sigma_y^2 \quad \sim \text{IG}(a_{\sigma,y}, b_{\sigma,y}); \\ \tau_\delta & \sim \text{IG}(a_{\tau,\delta}, b_{\tau,\delta}); & \sigma_\eta^{(k),2} \sim \text{IG}(a_{\sigma,\eta}, b_{\sigma,\eta}), \end{array} \tag{2.8}$$

for $k = 1, ..., K$, where Be and IG denote the Beta and inverse Gamma distributions. The fixed hyper-parameters in (2.8) are defined by the researcher. If no a priori information for $\{\beta_S^{(k)}\}$ and β_δ is available, one can let $\xi_\beta \to 0$, so that ultimately $\{\beta_S^{(k)}\}$ and β_δ are a priori completely unknown [3]. This limiting prior 'distribution' is improper, non-informative, and independent of the values of $\{b_S^{(k)}\}$, $\{\Sigma_{\beta,S}^{(k)}\}$, b_δ, and $\Sigma_{\beta,\delta}$. The priors assigned on $\phi_{S,x,l}^{(k),-1}$, $\phi_{\delta,x,l}^{(k),-1}$, and $\phi_{S,t,l}^{(k),-1}$ are standard choices and suggested in [30]. The proposed methodology can be used even if different priors for the parameters of the covariance are specified.

Prior distribution on the calibration parameters $\pi(\theta^{\otimes K})$ is specified according to the available a priori information. Available prior information about the dependency of calibration parameters, e.g. $\theta^{(k)}$ and $\theta^{(k')}$, between different computer models, e.g. $\mathscr{S}^{(k)}$ and $\mathscr{S}^{(k')}$, can be included in the priors. Usually, the researcher is confident that the ideal values of the calibration parameters lie in a specific range, and hence the associated priors have positive mass over a bounded region of $\Theta^{\otimes K}$. In such cases, if a priori information is available about a calibration parameter, one can assign truncated multivariate Normal prior distributions, otherwise one can assign uniform prior distributions.

Posterior model The joint posterior distribution $\pi(\mathcal{I}, \omega, \beta^{\otimes K}\varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K})$ according to the Bayes theorem admits density

$$\pi(\mathcal{I}, \omega, \theta^{\otimes K}, \beta^{\otimes K}, \varphi^{\otimes K}|z^{\otimes K}) \propto f(z^{\otimes K}|\omega, \beta^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K}) \Pr(\mathcal{I})\pi(\omega|\mathcal{I})\pi(\beta^{\otimes K})\pi(\varphi^{\otimes K})\pi(\theta^{\otimes K}). \tag{2.9}$$

It can be factorized as

$$\pi(\mathcal{I}, \omega, \beta^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K}) = \pi(\beta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})\pi(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K}) \tag{2.10}$$

where, on the right hand side of (2.10), the first distribution is a multivariate normal distribution $\beta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K} \sim \text{N}(\hat{\beta}^{\otimes K}, \hat{W}^{\otimes K})$ with mean and covariance matrix

$$\hat{\beta}^{\otimes K} = \hat{W}^{\otimes K}(H_z^{\otimes K,\intercal}\Sigma_z^{\otimes K,-1}z^{\otimes K} + \xi_\beta\Sigma_\beta^{-1}b_\beta); \tag{2.11}$$

$$\hat{W}^{\otimes K} = (H_z^{\otimes K,\intercal}\Sigma_z^{\otimes K,-1}H_z^{\otimes K} + \xi_\beta\Sigma_\beta^{-1})^{-1}, \tag{2.12}$$

respectively and $\Sigma_\beta^{\otimes K} = \text{diag}(\text{diag}(\Sigma_{\beta,S}^{(k)}; k = 1, ..., K), \Sigma_{\beta,\delta})$, while the second one admits density

$$\pi(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K}) \propto f(z^{\otimes K}|\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) \Pr(\mathcal{I})\pi(\omega|\mathcal{I})\pi(\varphi^{\otimes K})\pi(\theta^{\otimes K}), \tag{2.13}$$

$$f(z^{\otimes K}|\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) \propto |\det(\hat{W}^{\otimes K})|^{\frac{1}{2}}|\det(\Sigma_z^{\otimes K})|^{-\frac{1}{2}} \times \exp(-\frac{1}{2}z^{\otimes K,\intercal}\Sigma_z^{\otimes K,-1}z^{\otimes K} + \frac{1}{2}\hat{\beta}^{\otimes K,\intercal}\hat{W}^{\otimes K,-1}\hat{\beta}^{\otimes K}). \tag{2.14}$$

The joint posterior density (2.9) is intractable and known up to a normalizing constant in realistic scenarios; hence one can resort to Markov chain Monte Carlo (MCMC) in order to perform the computations.

2.3. Computations

We consider MCMC methods in order to facilitate the Bayesian computations. This requires to generate sample from joint posterior (2.10), which can be performed in two steps: (i.) simulate $\pi(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K})$; and (ii.) sample from $\pi(\beta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})$ given the values drawn at step (i.).

The conditional distribution $\pi(\beta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})$ is a multivariate normal $\text{N}(\hat{\beta}^{\otimes K}, \hat{W}^{\otimes K})$ and can be sampled directly. To simulate from distribution $\pi(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K})$, we design a MCMC sampler with four blocks updating $\pi(\mathcal{I}, \varpi|z^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K})$, $\pi(\varpi|z^{\otimes K}, \mathcal{I}, \varphi^{\otimes K}, \theta^{\otimes K})$, $\pi(\theta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \varpi, \varphi^{\otimes K})$, and $\pi(\varphi^{\otimes K}|z^{\otimes K}, \mathcal{I}, \varpi, \theta^{\otimes K})$. The MCMC sweep is presented in Algorithm 1 as a pseudo-code, and the associated blocks are discussed briefly in what follows.

Block BL-1 performs structural changes in the parametrization of the weight functions by changing the bases composition of the expansion in (2.5). Because it proposes changes in the dimensionality of the sampling space, it can be performed through the reversible jump (RJ) algorithm [34]. Here, we design local birth & death RJ moves. Briefly, we randomly select to

Algorithm 1 MCMC sweep.

[BL-1] Update $(\mathcal{I}, \omega)$**:** Simulate from $\pi(\mathcal{I}_k, \omega_k | z^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K})$ via RJ algorithm, for $k = 1, ..., K-1$.
[BL-2] Update ω**:** Simulate from $\pi(\omega_k | z^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K})$ via HRMH algorithm, for $k = 1, ..., K-1$.
[BL-3] Update $\theta^{\otimes K}$**:** Simulate from $\pi(\theta^{\otimes K} | z^{\otimes K}, \varpi, \varphi^{\otimes K})$ via a mixture of HRMH kernels.
[BL-4] Update $\varphi^{\otimes K}$**:** Simulate from $\pi(\varphi^{\otimes K} | z^{\otimes K}, \varpi, \theta^{\otimes K})$ via a mixture of MH kernels.

Algorithm 2 RJ moves proposing changes to the parameters $(\mathcal{I}_k, \omega_k)$. Notation: c_k denotes the size of $\mathcal{I}_k$, c denotes the carnality of Λ_{p_ϖ, d_x}, $\oplus$ denotes adding an element to a vector, $\ominus$ denotes removing an element from a vector, and $\mathrm{N}(\cdot|\cdot, \cdot)$ denotes the normal density.

Randomly choose to perform either a birth or a death move with probabilities P_{birth}, P_{death}, respectfully.

Birth move: $(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) \to (\mathcal{I}^+, \omega^+, \varphi^{\otimes K}, \theta^{\otimes K})$

1. randomly select a currently non-significant base with index $j_0 \in \Lambda_{p,d_x} - \mathcal{I}_k$ to include in the expansion.
2. compute the candidate $(\mathcal{I}^+, \omega^+)$ by appending as $\mathcal{I}_k^+ \leftarrow \mathcal{I}_k \oplus j_0$ and $\omega_k^+ \leftarrow \omega_k \oplus w_{j_0}$, where w_{j_0} is generated from distribution $Q(\mathrm{d}\cdot)$
3. accept the move, and the proposed values $(\mathcal{I}^+, \omega^+, \varphi^{\otimes K}, \theta^{\otimes K})$ with probability

$$\min(1, \frac{f(z^{\otimes K}|\mathcal{I}^+, \omega^+, \varphi^{\otimes K}, \theta^{\otimes K})\ \Pr(\mathcal{I}_k^+)\ \mathrm{N}(w_{j_0}|b_\omega, \xi_\omega^{-1})\ P_{\text{death}}\ 1/(c_k+1)}{f(z^{\otimes K}|\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})\ \Pr(\mathcal{I}_k)\ P_{\text{birth}}\ 1/(c-c_k)\ Q(w_{j_0})}).$$

Death move: $(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) \to (\mathcal{I}^-, \omega^-, \varphi^{\otimes K}, \theta^{\otimes K})$

1. randomly select a currently significant base with index $j_0 \in \mathcal{I}_k$ to remove from the expansion
2. compute the candidate $(\mathcal{I}^-, \omega^-)$ by removing $\mathcal{I}_k^- \leftarrow \mathcal{I}_k \ominus j_0$ and $\omega_k^- \leftarrow \omega_k \ominus w_{j_0}$
3. accept the move, and the proposed value $(\mathcal{I}^-, \omega^-, \varphi^{\otimes K}, \theta^{\otimes K})$ with probability

$$\min(1, \frac{f(z^{\otimes K}|\mathcal{I}^-, \omega^-, \varphi^{\otimes K}, \theta^{\otimes K})\ \Pr(\mathcal{I}_k^-)\ P_{\text{birth}}\ 1/(c-c_k+1)\ Q(w_{j_0})}{f(z^{\otimes K}|\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})\ \Pr(\mathcal{I}_k)\ \mathrm{N}(w_{j_0}|b_\omega, \xi_\omega^{-1})\ P_{\text{death}}\ 1/c_k}).$$

perform a Birth move with probability P_{birth}, or a Death move with probability P_{death}. According to the Birth move, a currently non-significant base is randomly proposed to be included in the weight function $\varpi_k(\cdot)$. According to the Death move: a significant base is randomly proposed to be removed from the weight function $\varpi_k(\cdot)$. The RJ transitions are presented as a pseudo-code in Algorithm 2. The specification of probabilities P_{birth}, P_{death} is problem dependent. A random choice between the two moves usually leads to acceptable mixing. Here, we use: $(P_{\text{birth}} = 1, P_{\text{death}} = 0)$ if only one basis is currently used for the weight function; $(P_{\text{birth}} = 0, P_{\text{death}} = 1)$ if all the available bases are currently used for the weight function; and $(P_{\text{birth}} = 0.5, P_{\text{death}} = 0.5)$ otherwise. A particular choice of the proposal distribution $Q(\mathrm{d}\cdot)$ that leads to simpler acceptance probability ratio is the prior, i.e. the Normal distribution with mean b_ω and variance ξ_ω, however other distributions can be used.

In block BL-2, parameters $\{\omega_k\}$ can be updated via Metropolis–Hastings (MH) algorithm [35] targeting $\pi(\omega_k|z^{\otimes K}, \varphi^{\otimes K}, \theta^{\otimes K})$. In block BL-3, the calibration parameters $\theta^{\otimes K}$ can be updated via a hit-and-run MH (HRMH) algorithms [36,37] targeting the conditional distributions of $\pi(\theta^{\otimes K}|z^{\otimes K}, \varpi, \varphi^{\otimes K})$. HRMH can be useful to facilitate the MCMC updates in this block, because $\theta^{\otimes K}$ may have dimensions with different ranges, or sharply constraint parameter space. In Block BL-4, the parameters $\varphi^{\otimes K}$ are updated via random walk Metropolis (RWM) algorithm targeting the full conditional distributions of $\{\{\phi_{S,x}^{(k)}, \phi_{S,t}^{(k)}, \tau_S^{(k)}, \sigma_\eta^{(k),2}; k = 1, ..., K\}, \phi_{\delta,x}, \tau_\delta, \sigma_y^2\}$. The conditional posterior distributions required in Algorithm 1 can be easily derived from (2.13).

The proposed MCMC sampler is valid, i.e. irreducible, aperiodic, and reversible. The Metropolis–Hastings updates in blocks BL-2, 3, and 4 can be tuned via an adaptive scheme [38]; these updates are presented briefly in Appendix A. In the presence of moderately large number of computer models, the computational overhead can be mitigated by using a convenient technique we provide in Appendix B.

At each iteration, the MCMC sampler requires the evaluation of the likelihood (2.14) involving the inversion of $\Sigma_z^{\otimes K}$. Because of the consideration of multiple computer models the size of $\Sigma_z^{\otimes K}$ may become large, and hence the direct inversion of $\Sigma_z^{\otimes K}$ via Cholesky decomposition (which scales $O(\cdot^3)$ with the matrix size) is computationally prohibitive. In Appendix B, we suggest a tailored technique to invert $\Sigma_z^{\otimes K}$ via Cholesky which can mitigate the computational overhead caused by the consideration of multiple models. It takes advantage of the block-sparse structure of $\Sigma_z^{\otimes K}$.

2.4. Inference, calibration, and prediction

The specification of the Bayesian model and design of the MCMC sampler allows one to perform inference, calibration, and prediction based on the proposed computer model calibration framework. Let $\mathcal{S}_N = \{(\mathcal{I}_t, \omega_t, \beta_t^{\otimes K} \varphi_t^{\otimes K}, \theta_t^{\otimes K});\ t = 1, ..., N\}$ be a MCMC sample generated according to Algorithm 1.

The posterior distributions of the statistical parameters $(\mathcal{I}, \omega, \beta^{\otimes K}, \varphi^{\otimes K})$, calibration parameters $\theta^{\otimes K}$, and their functions can be recovered from $\mathcal{S}_N$ via standard MCMC methods [39]. Inference on the weight functions provides a mean to

'rank' the available computer models at different input values in cases that the fidelity order is a priori unknown. This is because they indicate the contribution of each individual model in the mixture for the representation of the real system output. Posterior estimates for the weight functions $\{\varpi_k(\cdot)\}$ can be computed as

$$\hat{\varpi}_k(\cdot) \approx \frac{1}{N}\sum_{t=1}^{N} \frac{\exp(h^{(k)}_{\varpi,\mathcal{I}_t}(\cdot)^\intercal \omega_{k,\mathcal{I}_t})}{1+\sum_{j=1}^{K-1}\exp(h^{(k)}_{\varpi,\mathcal{I}_t}(\cdot)^\intercal \omega_{j,\mathcal{I}_t})}, \tag{2.15}$$

along with the associated standard errors according to the Markov chain CLT [40]. Moreover, the weight functions allow the determination of a reasonable input space partition $\{\mathcal{X}_k\}_{k=1}^K$ where each sub-region $\mathcal{X}_k = \{x \in \mathcal{X} | \varpi_k(x) = \max(\varpi_1(x), ..., \varpi_K(x))\}$ includes the input values that model $\mathscr{S}^{(k)}$ is more preferable to be used than the rest. Let as define the integrated posterior weight over input sub-region $\mathcal{A} \subseteq \mathcal{X}$ as $\{\varpi_k(\mathcal{A}) = \int_{\mathcal{A}} \varpi_k(x)\mathrm{d}x\}$. Then $\{\varpi_k(\mathcal{A})\}$ can be used as an indicator of the total contribution of computer model $\{\mathscr{S}^{(k)}\}$ to the representation of $\mathscr{Z}$ throughout an input sub-region $\mathcal{A}$. The estimation of $\{\varpi_k(\mathcal{A})\}$ can be performed numerically by using (2.15). Bayesian point estimates of the calibration parameter $\theta^{\otimes K}$, can be computed, for example, such as the maximum a posteriori (MAP) estimate or the posterior mean.

For $\mathscr{C}^{\otimes K}$, the full conditional predictive distribution of $\zeta(x)|z^{\otimes K}, \mathcal{I}, \varpi, \beta^{\otimes K}\varphi^{\otimes K}, \theta^{\otimes K}$ integrated out with respect to $\pi(\beta^{\otimes K}|z^{\otimes K}, \mathcal{I}, \varpi, \varphi^{\otimes K}, \theta^{\otimes K})$ is denoted as $f(\zeta(\cdot)|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})$. It is a Gaussian process, with mean and covariance functions

$$\mu_\zeta^{\otimes K}(x|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) = h^{\otimes K}(x,\theta^{\otimes K})\hat{\beta}^{\otimes K} + v^{\otimes K}(x,\theta^{\otimes K})^\intercal \Sigma_z^{\otimes K,-1}(z^{\otimes K} - H^{\otimes K}\hat{\beta}^{\otimes K}); \tag{2.16}$$

$$\begin{aligned} c_\zeta^{\otimes K}(x,x'|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}) = &\sum_{k=1}^{K} \varpi_k(x)\varpi_k(x')c_S^{(k)}((x,\theta^{(k)}),(x',\theta^{(k)})) \\ &+ c_\delta(x,x') - v^{\otimes K}(x,\theta^{\otimes K})^\intercal \Sigma_z^{\otimes K,-1} v^{\otimes K}(x',\theta^{\otimes K}) \\ &+ [h^{\otimes K}(x,\theta^{\otimes K}) - H_z^{\otimes K,\intercal}\Sigma_z^{\otimes K,-1}v^{\otimes K}(x,\theta^{\otimes K})]^\intercal \hat{W}^{\otimes K} \\ &\times [h^{\otimes K}(x',\theta^{\otimes K}) - H_z^{\otimes K,\intercal}\Sigma_z^{\otimes K,-1}v^{\otimes K}(x',\theta^{\otimes K})], \end{aligned} \tag{2.17}$$

correspondingly, where

$$h^{\otimes K}(x,\theta^{\otimes K}) = [\varpi_1(x)h_S^{(1)}(x,\theta^{(1)}), ..., \varpi_K(x)h_S^{(K)}(x,\theta^{(K)}), h_\delta(x)]^\intercal; \text{ and}$$

$$v^{\otimes K}(x,\theta^{\otimes K}) = \begin{bmatrix} (\sum_{k=1}^{K}\varpi_k(x)\varpi_k(x_i)c_S^{(k)}((x,\theta^{(k)}),(x_i,\theta^{(k)})) + c_\delta(x,x_i);\ i=1:n)^\intercal \\ (\varpi_1(x)c_S^{(1)}((x,\theta^{(1)}),(x_i,t_i^{(1)}));\ i=n+1:n+m^{(1)})^\intercal \\ \vdots \\ (\varpi_K(x)c_S^{(K)}((x,\theta^{(K)}),(x_i,t_i^{(K)}));\ i=n+\sum_{k<K}m^{(k)}+1:n^{\otimes K})^\intercal \end{bmatrix}.$$

The marginal predictive distribution density, needed to perform predictions,

$$f(\zeta(x)|z^{\otimes K}) = \sum_{\mathcal{I}}\int f(\zeta(x)|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})\pi(\mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K}|z^{\otimes K})\mathrm{d}(\omega, \varphi^{\otimes K}, \theta^{\otimes K}) \tag{2.18}$$

is not available in closed form, however it can be approximated via MCMC integration as

$$\hat{f}(\zeta(x)|z^{\otimes K}) = \frac{1}{N}\sum_{t=1}^{N} f(\zeta(x)|z^{\otimes K}, \mathcal{I}_t, \omega_t, \varphi_t^{\otimes K}, \theta_t^{\otimes K}). \tag{2.19}$$

A common choice that leads to reliable, as well as mathematically convenient, surrogate models for $\zeta(x)$ is based on the expectation $\mu_\zeta^{\otimes K}(x|z^{\otimes K}, \mathcal{I}, \omega, \varphi^{\otimes K}, \theta^{\otimes K})$ with respect to the joint posterior, which can be approximated in a $\sqrt{N}$-CLT fashion as

$$\hat{\mu}_\zeta^{\otimes K}(x|z^{\otimes K}) = \frac{1}{N}\sum_{t=1}^{N}\mu_\zeta^{\otimes K}(x|z^{\otimes K}, \mathcal{I}_t, \omega_t, \varphi_t^{\otimes K}, \theta_t^{\otimes K}), \tag{2.20}$$

for a given $x \in \mathcal{X}$. Note that for the computation of (2.19) and (2.20) we do not need to generate values $\{\beta_t^{\otimes K}\}$ and hence the associated sampling step can be omitted.

Suppose we wish to predict the real system output in the context that one or more of the inputs is subject to parametric variability. Here, uncertainty analysis can be performed along the same lines of [1,41,42] by using the surrogate model estimate (2.20) and marginal predictive density estimate (2.19).

Remark 1. The procedure builds the unknown weight functions by selecting significant bases and evaluating the corresponding coefficients in a stochastic data-driven manner. This bases selection mechanism can provide parsimonious bases representations for the weight functions.

3. Numerical examples

We provide a validation study of the proposed method with that of Goh et al. [11] in a simple benchmark example (Sec. 3.1). We demonstrate the performance of the method and compare it with that of Kennedy and O'Hagan [1] in a more challenging example with PDEs where the fidelity order of the models is unknown and changes over the spatial space (Sec. 3.2). We use the proposed method to address a challenging real-world large-scale climate application involving multiple computer models with different physics (Sec. 3.3).

3.1. Validation example: a simple multi-fidelity case

We consider there are available two computer models $\mathscr{S}^{(1)}$, $\mathscr{S}^{(2)}$ that aim at simulating the real system $\mathscr{Z}$ with different levels of fidelity, and there is interest in designing a predictive model for $\mathscr{Z}$. To validate the performance of our method, we pretend that we do not know which model is more accurate; although $\mathscr{S}^{(2)}$ has higher fidelity than $\mathscr{S}^{(1)}$ by construction. Moreover, we validate our method with respect to the multi-fidelity method of Goh et al. [11] which is exclusively designed to address only cases with known fidelity order; hence for method of Goh et al. [11] we use the extra information that $\mathscr{S}^{(2)}$ is of higher fidelity than $\mathscr{S}^{(1)}$.

Let us consider 2D elliptic PDEs

$$\left.\begin{aligned} -\nabla\cdot(c(x,(\vartheta_1,\vartheta_2))\nabla u^{(1)}(x,(\vartheta_1,\vartheta_2)) &= f(x), \quad x\in\mathcal{X}-\partial\mathcal{X} \\ u^{(1)}(x,(\vartheta_1,\vartheta_2)) &= 0, \quad x\in\partial\mathcal{X} \end{aligned}\right\}; \tag{3.1}$$

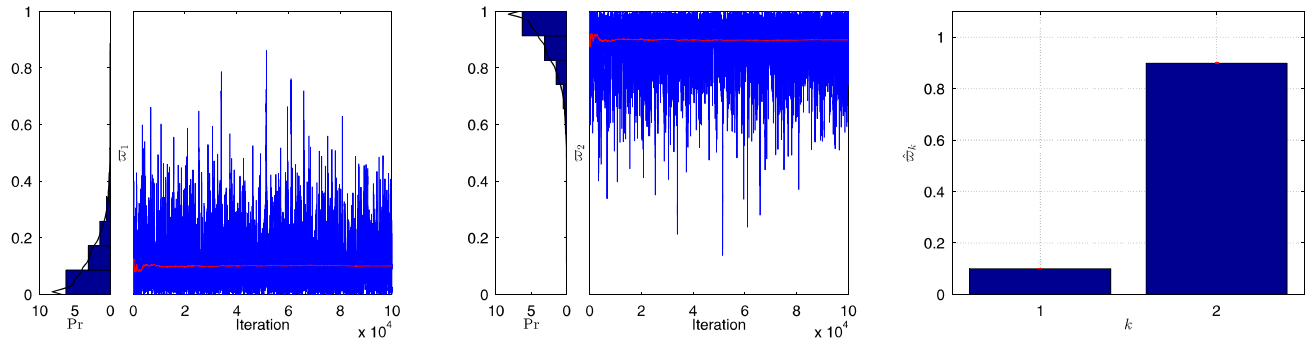
$$\left.\begin{aligned} -\nabla\cdot(c(x,(\vartheta_1,\vartheta_2))\nabla u^{(2)}(x,(\vartheta_1,\vartheta_2,\vartheta_3)) & \\ + a(x,\vartheta_3)u^{(2)}(x,(\vartheta_1,\vartheta_2,\vartheta_3)) &= f(x), \quad x\in\mathcal{X}-\partial\mathcal{X} \\ u^{(2)}(x,(\vartheta_1,\vartheta_2,\vartheta_3)) &= 0, \quad x\in\partial\mathcal{X} \end{aligned}\right\}, \tag{3.2}$$

where $x=(x_1,x_2)$, $\mathcal{X}=[0,1]^2$, $\vartheta_1\in(0,1)$, $\vartheta_2\in(0,1)$, and $\vartheta_3\in(0,1)$. Let $f(x)=-100\cos(\frac{\pi}{2}(1-x_1+x_2))$, $c(x,(\vartheta_1,\vartheta_2))=\exp(\sum_{j=1}^{2}(\frac{1}{j})^2\sin(2j\pi x_1)\cos(2(3-j)\pi x_2)\vartheta_j)$, and $a(x,\vartheta_3)=5\exp(\vartheta_3 x_1+(1-\vartheta_3)x_2)$.

We assume that the real system $\mathscr{Z}$ under study has output function $\zeta(x)=u^{(2)}(x,(\theta_1,\theta_2,\theta_3))+\delta(x)$, where $\delta(x)=2(x_1-0.5)^2(x_2-0.5)^2$, and noise scale $\sigma_y=0.01$. The computer model $\mathscr{S}^{(1)}$ has output function $S^{(1)}(x,t^{(1)})=u^{(1)}(x,(t_1^{(1)},t_2^{(1)}))$, where $t^{(1)}=(t_1^{(1)},t_2^{(1)})\in[0,1]^2$, and uses a FEM solver with the domain $\mathcal{X}$ discretized in 177 nodes and 317 triangles by the Delaunay triangulation algorithm. Computer model $\mathscr{S}^{(2)}$ has output function $S^{(2)}(x,t^{(2)})=u^{(2)}(x,(t_1^{(2)},\theta_2,t_2^{(2)}))$, where $t^{(2)}=(t_1^{(2)},t_2^{(2)})\in[0,1]^2$, and uses a FEM solver with the domain $\mathcal{X}$ discretized in 665 nodes and 1248 triangles by the Delaunay triangulation algorithm. Due to the more accurate PDE solver involved, it is clear to see that computer model $\mathscr{S}^{(2)}$ has higher fidelity level than $\mathscr{S}^{(2)}$ with respect to $\mathscr{Z}$, by contraction. For the calibration parameters of $\mathscr{S}^{(1)}$ and $\mathscr{S}^{(2)}$ we consider ideal values $\theta^{(1)}=(\theta_1,\theta_2)^\intercal$ and $\theta^{(2)}=(\theta_1,\theta_3)^\intercal$ respectively, with $\theta_1=0.3$, $\theta_2=0.6$, and $\theta_3=0.5$. Here, the calibration parameters $\theta_1^{(1)}$ and $\theta_1^{(2)}$ are assumed to have the same physical meaning, and hence are treated as shared calibration parameters; therefore $\theta_1^{(1)}=\theta_1^{(2)}$. Calibration parameters $\theta_2^{(1)}$ and $\theta_2^{(2)}$ belong to models $\mathscr{S}^{(1)}$ and $\mathscr{S}^{(2)}$ correspondingly, and affect different parts of the corresponding PDEs; hence they are treated as separate parameters. We use the Bayesian calibration mixture model set-up. The means of the Gaussian process priors were modeled as constants. On the free calibration parameters, i.e. θ_1, θ_2, and θ_3, we assigned independent uniform priors. To make the challenge bigger, we pretend that we do not know a priori the fidelity order of the models, and we assign weakly-informative priors on the weights; i.e. $b_\omega=0$, $\xi_\omega^{-1}=10^2$. On the rest statistical parameters, we assign weakly-informative priors; specifically $a_{\tau,S}=b_{\tau,S}=10^{-3}$, $a_{\tau,\delta}=b_{\tau,\delta}=10^{-3}$, $a_{\sigma,y}=b_{\sigma,y}=10^{-3}$, and $a_{\sigma,\eta}=b_{\sigma,\eta}=10^{-3}$.

We assume there are available 10 experimental data, which in reality are generated by drawing randomly the input values, and computing the corresponding output contaminated with noise. The involved PDE was solved by using an acceptably accurate FEM solver with the domain $\mathcal{X}$ discretized in 2577 nodes and 4992 triangles by the Delaunay triangulation algorithm. For the computer models $\mathscr{S}^{(1)}$ and $\mathscr{S}^{(2)}$, we use a 40-run, and 25-run LHS to generate the input values and compute the corresponding outputs. We generate a validation data set at 150 randomly selected input points. The join posterior distribution is simulated via MCMC sampler with 11000 iterations where the first 1000 where discarded as burn in.

Regarding the weights, in Figs. 3.1a and 3.1b, the trace plots of the generated weights suggest that the MCMC mixing was adequate, and that the ergodic average (and hence the MCMC estimate) of the weights converges. Precisely, the MCMC estimates (posterior expectations) of the weights ϖ_1 and ϖ_2 are 0.11 and 0.89 with standard errors $3\cdot10^{-4}$ and $3\cdot10^{-4}$ correspondingly, (Fig. 3.1c). The estimates of the associate marginal posterior densities, provided as histograms in Figs. 3.1a and 3.1b, have the main mass around the posterior expectation estimate which indicates a clear evidence that the weights associated with $\mathscr{S}^{(2)}$ are more likely to have higher values than those of $\mathscr{S}^{(1)}$. This result is consistent with the fact that $\mathscr{S}^{(2)}$ is more accurate than $\mathscr{S}^{(1)}$ with respect to $\mathscr{Z}$, and hence it suggests that the mixture weights can give an



(a) Weight ϖ_1 (b) Weight ϖ_2 (c) Posterior estimates

Fig. 3.1. (a–b): Histograms and trace-plots of the MCMC sample of weights $\{\varpi_k; k = 1, 2\}$. (c): The estimated posterior expectation is $\hat{\varpi} = (0.11, 0.89)$ (Section 3.1).

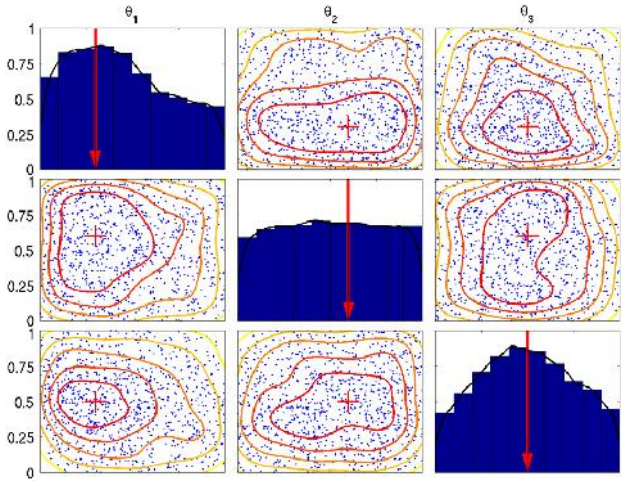


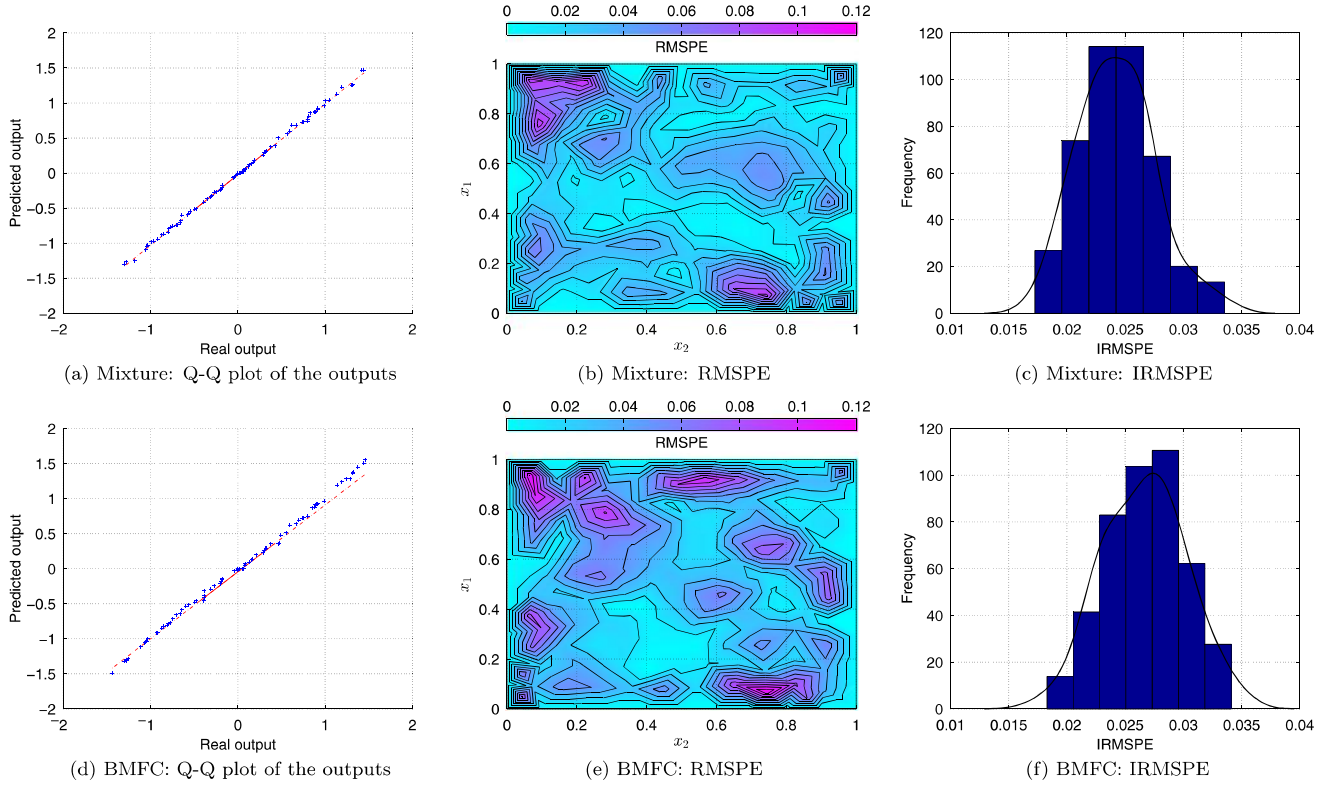
Fig. 3.2. Estimated marginal posterior distribution densities of the calibration parameters. The 'ideal' values of the parameters are pointed by red arrows and red crosses (Section 3.1). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

indication about the fidelity order of the computer models. Regarding inference on the calibration parameters, Fig. 3.2 presents the estimated posterior densities of the calibration parameters. The MAP estimates of the calibration parameters are $\hat{\theta}_1^{\text{MAP}} = 0.36$, $\hat{\theta}_2^{\text{MAP}} = 0.41$, and $\hat{\theta}_3^{\text{MAP}} = 0.43$. We observe that our method produced unimodal posterior densities for the 'ideal' calibration parameters θ_1 and θ_3, while the main mass is above the area around the corresponding ideal calibration values. Regarding θ_2, our method produces a rather uniform marginal posterior density which suggests that this parameter might not significantly affect the response of $\mathscr{J}^{(2)}$.

We compare our method with the Bayesian multi-fidelity calibration (BMFC) procedure of Goh et al. [11] in terms of predictive ability (Fig. 3.3). For BMFC, we use the default Gaussian processes and prior model specifications suggested in [11], which actually resemble to those specified for our method. Additionally, for BMFC, we consider the extra information that the fidelity order is a priori known (i.e., $\mathscr{J}^{(2)}$ more accurate than $\mathscr{J}^{(1)}$). As performance measures, we consider the root mean squared predictive error (RMSPE),[1] and the integrated RMSPE (IRMSPE).[2] Figs. 3.3a and 3.3d suggest that both procedures present adequate predictive ability. Figs. 3.3c, 3.3f, 3.3c, 3.3f were produced based on 32 realizations of the training data sets. We observe that our method has successfully managed to produce predictions as accurate as those produced by the problem specific BMFC procedure, throughout the input domain (Figs. 3.3c and 3.3f). In Figs. 3.3c and 3.3f, we observe that both methods produced comparable IRMSPE. It is quite encouraging to observe that our method, which has a more general scope, can present comparable predictive ability with the problem specific BMFC procedure. This is because our method can address problems that the fidelity order is unknown and hence has a more general scope. Hence, this validation study suggests that the proposed method can be a reliable counterpart to the BMFC.

[1] $\text{RMSPE}(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\zeta}_i(x) - y(x))^2}$ computed based on N generated training data-sets.

[2] $\text{IRMSPE} = \frac{1}{\text{Card}(\mathcal{X}_{\text{grid}})}\sum_{x\in\mathcal{X}_{\text{grid}}} \text{RMSPE}(x)$, where $\mathcal{X}_{\text{grid}}$ is a set of gridded points in the input domain $\mathcal{X}$.



(a) Mixture: Q-Q plot of the outputs (b) Mixture: RMSPE (c) Mixture: IRMSPE

(d) BMFC: Q-Q plot of the outputs (e) BMFC: RMSPE (f) BMFC: IRMSPE

Fig. 3.3. (a, d): The Q–Q plots present the predicted output of the surrogate against the real output of the real system. (b, e): The contour plots present the RMSPEs as functions of the input parameter $x \in \mathcal{X}$, and (c, f): the histograms represent the distribution of the IRMSPEs, generated based on 32 realizations of the training data and fitting the predictive model. Procedures under comparison: the proposed method (Mixture), and Bayesian multi-fidelity calibration method (BMFC) (Section 3.1).

3.2. *Numerical example: a case of computer models with unknown fidelity order*

A simulation study is conducted to assess the performance of the proposed method, and compare it with that of the standard Bayesian single model calibration (BSMC) method of Kennedy and O'Hagan [1]. We consider there are available computer models $\mathscr{S}^{(1)}$, $\mathscr{S}^{(2)}$ aiming at simulating the real system $\mathscr{Z}$, with unknown fidelity order that changes across the input. Here, $\mathscr{S}^{(1)}$, $\mathscr{S}^{(2)}$ have their own unique abilities to represent $\mathscr{Z}$ and hence combining them can lead to better predictions and simulations.

Let us consider two 2D elliptic PDEs, differing on the diffusion coefficients and source terms,

$$\left.\begin{aligned} -\nabla\cdot(c^{(k)}(x,\vartheta^{(k)})\nabla u^{(k)}(x,\vartheta^{(k)}) &= f^{(k)}(x); \quad x\in\mathcal{X}-\partial\mathcal{X},\\ u^{(k)}(x,\vartheta^{(k)}) &= 0; \quad x\in\partial\mathcal{X}, \end{aligned}\right\} \tag{3.3}$$

for $k=1,2$, where $f^{(1)}(x)=10^2$, $f^{(2)}(x)=-10^2$,

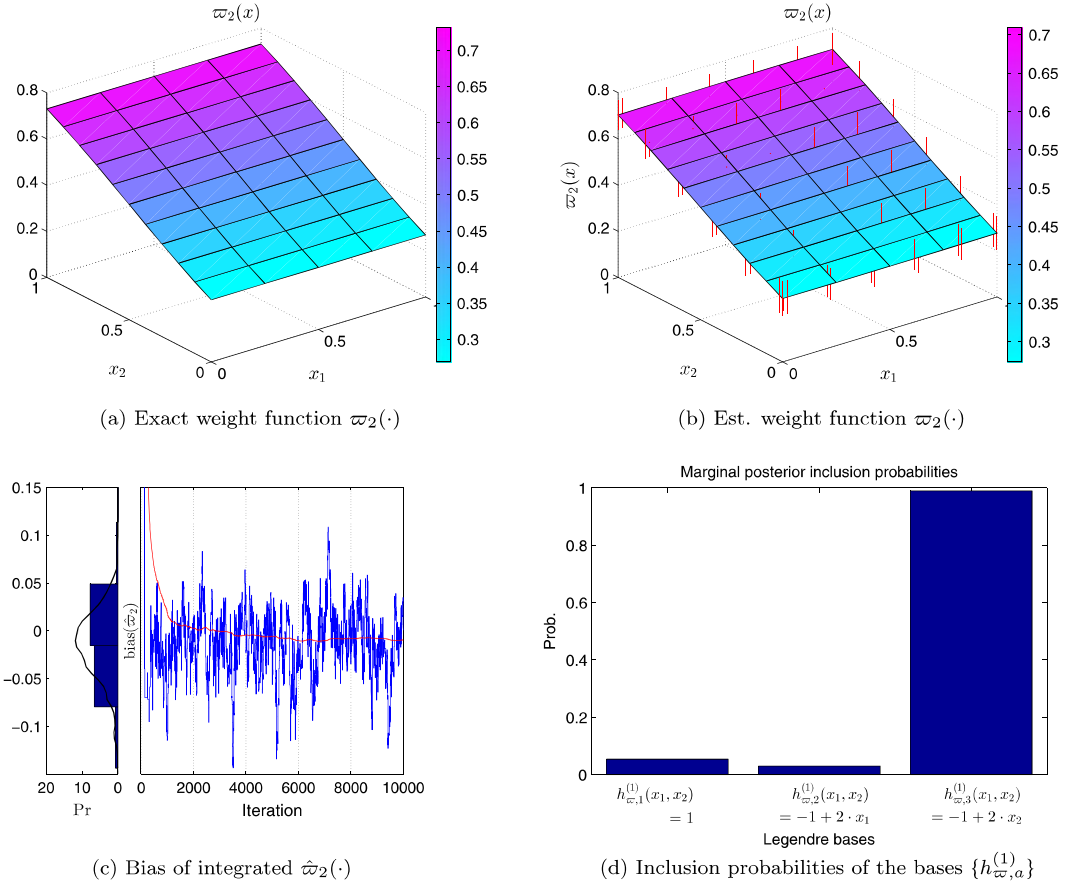
$$c^{(1)}(x,\vartheta^{(1)}) = 2\exp\Big(\sum_{i=1}^{2}\frac{1}{i}\sin(2\pi x_1 i)\cos(2\pi x_2(3-i))\vartheta_i^{(1)}\Big)(\mathbb{1}_{(-\infty,0.5)}(x_1)+\exp(4x_1)\mathbb{1}_{(0.5,+\infty)}(x_1));$$

$$c^{(2)}(x,\vartheta^{(2)}) = 2\exp\Big(\sum_{i=1}^{3}\frac{1}{i}\sin(2\pi(x_1-x_2)i)\cos(2\pi(x_1-x_2)(4-i))\vartheta_i^{(2)}\Big)(\mathbb{1}_{(-\infty,0.5)}(x_2)+\exp(4x_1)\mathbb{1}_{(0.5,+\infty)}(x_2))$$

with $x\in\mathcal{X}$, $\mathcal{X}=(0,1)^2$, $\vartheta^{(1)}\in(0,1)^2$, and $\vartheta^{(2)}\in(0,1)^3$.

We assume that the real system $\mathscr{Z}$ under study has output function $\zeta(x)=\sum_{k=1}^{2}\varpi_k(x)u^{(k)}(x,\theta^{(k)})+\delta(x)$, with $\varpi_1(x)=(1+\exp(-1+2x_2))^{-1}$, $\varpi_2(x)=1-\varpi_1(x)$, $\delta(x)=0.1(x_1-0.5)(x_2-0.5)$, and noise scale $\sigma_y=0.01$. The ideal values of the calibration parameters are: $\theta^{(1)}=(0.8,0.5)^\intercal$, and $\theta^{(2)}=(0.6,0.7,0.1)^\intercal$. The computer models $\{\mathscr{S}^{(k)};k=1,2\}$, have output functions $\{S_k(x,t^{(k)})=u^{(k)}(x,t^{(k)});k=1,2\}$, where $t^{(1)}\in[0,1]^2$, and $t^{(2)}\in[0,1]^3$, and use finite element method (FEM) solvers [43] with the domain $\mathcal{X}$ is discretized in 665 nodes and 1248 triangles according to the Delaunay triangulation algorithm. We observe that the real system $\mathscr{Z}$ can be represented by the computer models $\mathscr{S}^{(1)}$, $\mathscr{S}^{(2)}$ in a combination; i.e. $\zeta(x)=\sum_{k=1}^{2}\varpi_k(x)S_k(x,t^{(k)}=\theta^{(k)})+\delta(x)$.

The training data-set comprises a set of experimental observations at 14 randomly selected points; and two simulated data-sets for $\mathscr{S}^{(1)}$ and $\mathscr{S}^{(2)}$ at 30 and 35 input points selected through Latin hypercube sampling (LHS) [44]. For the



(a) Exact weight function $\varpi_2(\cdot)$

(b) Est. weight function $\varpi_2(\cdot)$

(c) Bias of integrated $\hat{\varpi}_2(\cdot)$

(d) Inclusion probabilities of the bases $\{h^{(1)}_{\varpi,a}\}$

Fig. 3.4. (a): Exact weight function $\varpi_2(x) = (1+\exp(1-2x_2))^{-1}$ presented by colored surface. (b): Estimated weight function $\hat{\varpi}_2(\cdot) = 1-\hat{\varpi}_1(\cdot)$ presented by colored surface and 95% credible intervals presented by red bars. (c): Ergodic estimate of bias of integrated $\varpi_2(\cdot)$ bias$(\varpi_2(\mathcal{X})) = \hat{\varpi}_2(\mathcal{X}) - \varpi_2(\mathcal{X})$. (d): Frequency that each basis was included in the weight function $\varpi(\cdot)$ as significant (Section 3.2). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

generation of the training data, the PDEs in (3.3) were solved by using FEM solver where the domain $\mathcal{X}$ was discretized in 665 nodes and 1248 triangles according to the Delaunay triangulation algorithm. The validation data-set is generated at 150 randomly selected input points. We consider the Bayesian calibration of computer mixture set-up in Section 2. The mean of the Gaussian process priors of the output function of computer models and the discrepancy were modeled as Legendre polynomial expansion of 2nd degree and 0th degree correspondingly. For the representation of the weight functions, we considered a pool of 1st degree multivariate Legendre polynomial bases. We assign non- or weakly-informative priors on the statistical parameters; specifically $a_{\tau,S} = b_{\tau,S} = 10^{-3}$, $a_{\tau,\delta} = b_{\tau,\delta} = 10^{-3}$, $a_{\sigma,y} = b_{\sigma,y} = 10^{-3}$, and $a_{\sigma,\eta} = b_{\sigma,\eta} = 10^{-3}$. We assign a priori independent uniform priors on the calibration parameters. The join posterior distribution was sampled via the proposed MCMC sampler (Algorithm 1) with 11000 iterations where the first 1000 where discarded as burn in.

We examine inference on the weight functions in Fig. 3.4. We observe that the exact $\varpi_2(\cdot)$ in Fig. 3.4a is close to the estimated one in Fig. 3.4b and inside the 95% credible intervals produced by the proposed method. In Fig. 3.4c, the histogram of the bias of the estimated $\varpi_2(\mathcal{X})$, i.e. bias$(\varpi_2(\mathcal{X})) = \hat{\varpi}_2(\mathcal{X}) - \varpi_2(\mathcal{X})$, has the main mass over a narrow area around zero (± 0.05), the associated ergodic average converges to zero, and the trace plot indicates that the chain has a good mixing. In Fig. 3.4d, we observe that the procedure has successfully determined a sparse representation for the weight functions. Precisely, it has discovered that $\varpi_1(\cdot)$, (and hence $\varpi_2(\cdot)$), can be represented by only one Legendre basis function; i.e., $h^{(1)}_{\varpi,3}(x_2) = (-1+2x_2)$. This is because the frequency of the bases in the MCMC sample (posterior inclusion probability estimate) is 0.98 for $h^{(1)}_{\varpi,3}(x_2)$, and smaller than 0.07 for $h^{(1)}_{\varpi,1}(x_2)$ and $h^{(1)}_{\varpi,2}(x_2)$. Furthermore, we assess inference on the calibration parameters. In Fig. 3.5, we observe that in most of the cases, the marginal posterior distribution densities of the calibration parameters are unimodal and mainly concentrated above areas around the corresponding ideal values. In Fig. 3.6, we plot the output of the computer model mixture (weighted and calibrated according to the proposed method), the output of single computer models (calibrated by the BSMC method), and the output of the real system without noise. We used MAP estimates for the calibration parameters. We observe that the calibrated computer model mixture, fitted by the proposed method, successfully represents the real system, while the single models calibrated by the BSMC method fail. Therefore, the proposed method can successfully address problems where there are available multiple computer models with unknown fidelity order and there is need to accurately simulate the real system.

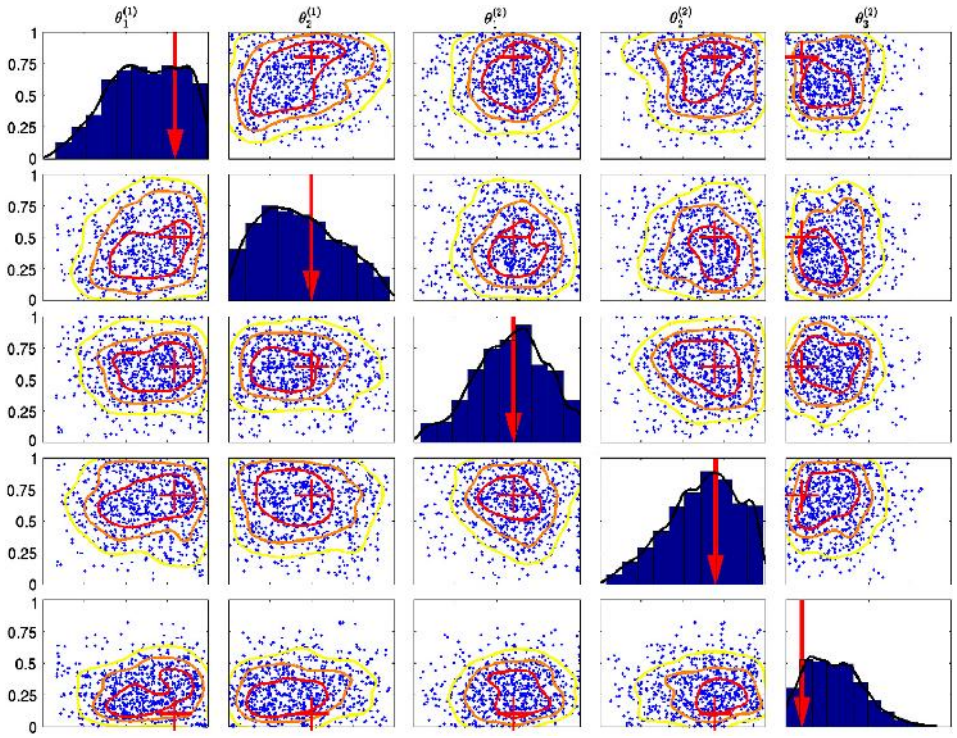
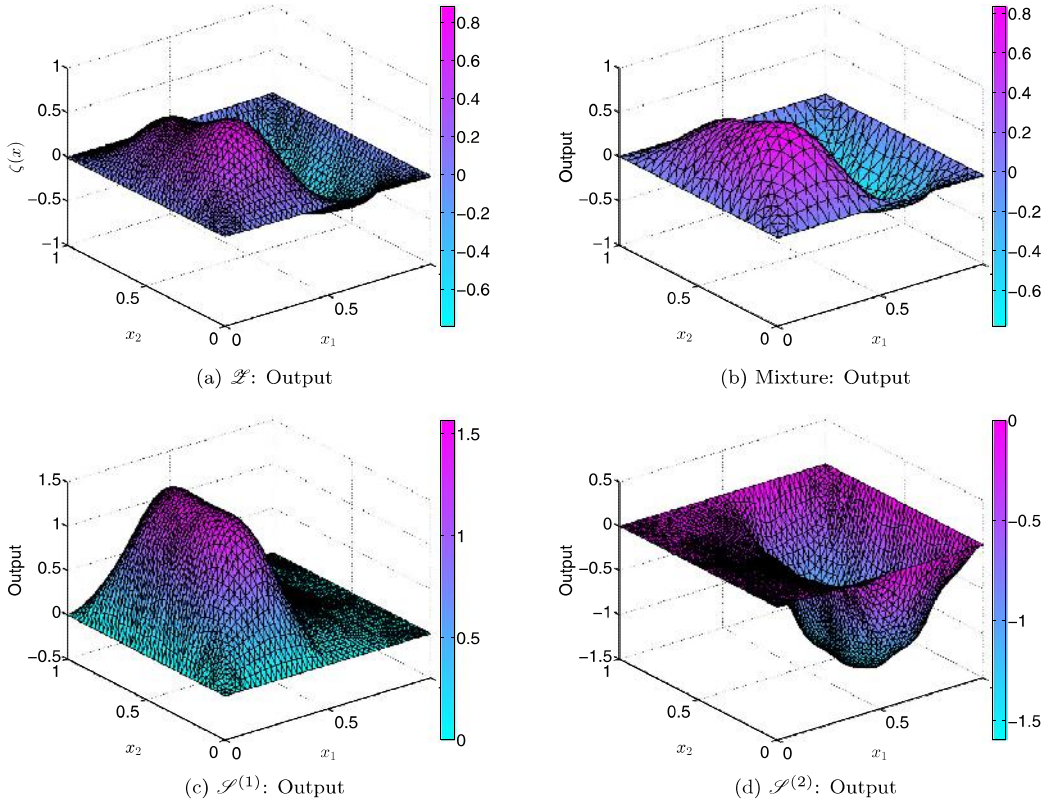


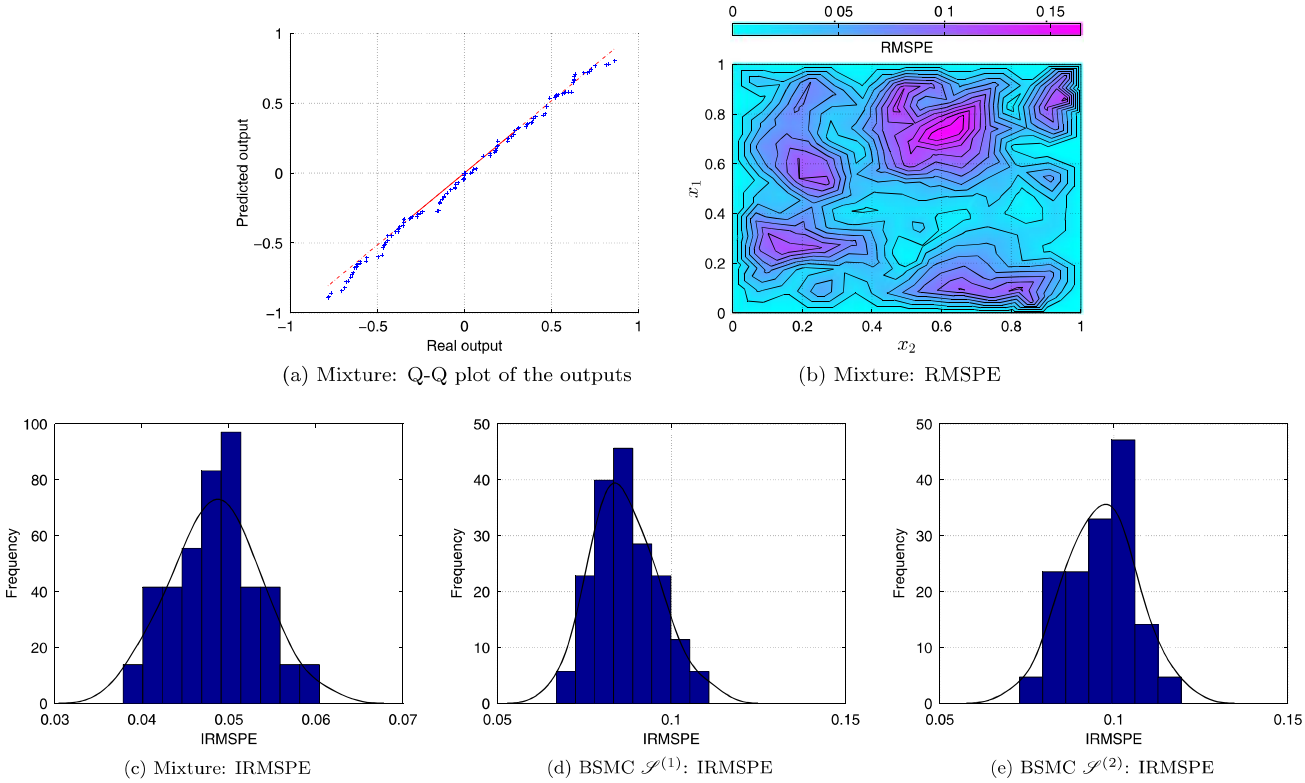
Fig. 3.5. Estimated marginal posterior distribution densities of the calibration parameters. The 'ideal' values of the parameters are pointed by red arrows and red crosses (Section 3.2). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



(a) $\mathscr{L}$: Output (b) Mixture: Output

(c) $\mathscr{S}^{(1)}$: Output (d) $\mathscr{S}^{(2)}$: Output

Fig. 3.6. Output functions of the: (a) real system, (b) the computer mixture model weighted and calibrated with the proposed method, (c) the $\mathscr{S}^{(1)}$ calibrated by BSMC, and (d) the $\mathscr{S}^{(2)}$ calibrated by BSMC (Section 3.2).



(a) Mixture: Q-Q plot of the outputs (b) Mixture: RMSPE

(c) Mixture: IRMSPE (d) BSMC $\mathscr{I}^{(1)}$: IRMSPE (e) BSMC $\mathscr{I}^{(2)}$: IRMSPE

Fig. 3.7. (a): The Q–Q plot presents the predicted output of the surrogate model against the real output of the real system. (b): The contour plots present the RMSPEs as functions of the input parameter $x \in \mathcal{X}$, and (c–e): the histograms represent the distribution of the IRMSPEs, generated based on 32 realizations of the training data and fitting the predictive model. Procedures under comparison: the proposed method (Mixture), and BSMC method (Section 3.2).

We examine the predictive ability of the proposed method. As performance measures, we consider the root mean squared predictive error (RMSPE),[3] and the integrated RMSPE (IRMSPE).[4] In Fig. 3.7a, we observe that the predictions produced by the proposed method are close to the output values generated by the real system at the same input points. Moreover, we observe that the produced RMSPE in Fig. 3.7b has small values throughout the input space. Hence, the proposed method can predict the output of the real system adequately. We compare the predictive ability of the proposed method with that of the standard Bayesian single model calibration (BSMC) procedure of Kennedy and O'Hagan [1] with respect to the IRMSPE. In Figs. 3.7c–3.7e, the histograms of the IRMSPE values were generated based on 32 realizations of the training data and fitting the predictive model. In Fig. 3.7c–3.7e, we observe that it is more likely for the proposed method to produce smaller IRMSPE than the BSMC method. This suggests that, the proposed method provides more accurate predictions than the BSMC, when multiple computer models with unknown fidelity order are available.

3.3. Application to large-scale climate modeling

Set-up of the application and computer models We consider the Advanced Research Weather Research and Forecasting Version 3.2.1 (WRF Version 3.2.1) climate model [12] constrained in the geographical domain 25°–44°N and 112°–90°W over the Southern Great Plains (SGP) region, and we concentrate on the average monthly precipitation response. WRF is employed with the Kain–Fritsch convective parametrization scheme (KF CPS) [16] as in [45]. The KF CPS is a simple 1D mass flux cloud model specifically designed for mesoscale models [16], including WRF, with a moderate grid spacing 10 km–100 km. The 5 most critical parameters [45] of the KF scheme are: the coefficient related to downdraft mass flux rate P_{d} that takes values in range $[-1, 1]$; the coefficient related to entrainment mass flux rate P_{e} that takes values in range $[-1, 1]$; the maximum turbulent kinetic energy in sub-cloud layer ($\mathrm{m}^2\,\mathrm{s}^{-2}$) P_{t} that takes values in range $[3, 12]$; the starting height of downdraft above updraft source layer (hPa) P_{h} that takes values in range $[50, 350]$; and the average consumption time of convective available potential energy P_{c} that takes values in range $[900, 7200]$. The ranges of the KF CPS parameters are quite wide and hence cause higher uncertainties in climate simulations due to the non-linear interactions and compensating errors of the parameters [46,47,45]. Other specifications used are the Morrison 2-moment cloud microphysics scheme [48], the Noah land surface model [49], and the Mellor–Yamada–Janjic [50] planetary boundary layer turbulence scheme. Here, we consider

[3] $\mathrm{RMSPE}(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\zeta}_i(x) - y(x))^2}$ computed based on N generated training data-sets.

[4] $\mathrm{IRMSPE} = \frac{1}{\mathrm{Card}(\mathcal{X}_{\mathrm{grid}})}\sum_{x\in\mathcal{X}_{\mathrm{grid}}} \mathrm{RMSPE}(x)$, where $\mathcal{X}_{\mathrm{grid}}$ is a set of gridded points in the input domain $\mathcal{X}$.

two different radiation schemes, the Rapid Radiative Transfer Model (RRTMG) for General Circulation Models [51], and the Community Atmosphere Model 3.0 (CAM) [14]. Moreover, we consider two grid spacing, 25 km and 50 km spacing, referring to the horizontal resolutions. Here, higher grid spacing does not necessarily lead to more accurate simulations with respect to the precipitation because WRF performance is sensitive to other physical parametrizations which is uncertain how they are affected to the grid spacing.

The available computer models are three different sub-models of the WRF with physics and fidelity variations. The first model involves the RRTMG radiation scheme with 25 km horizontal grid spacing and 36 sigma levels from the surface to 1000 hPa, and is labeled as RRTMG25; the second model involves the RRTMG radiation scheme with 50 km grid spacing, and is labeled as RRTMG50; and the third one involves the CAM 3.0 radiation scheme with 25 km grid spacing, and is labeled as CAM25. The output is the monthly average precipitation, the calibration parameters are the parameters of KF CPS, and the input are the coordinates in SGP.

Interest lies in combining properly the above computer models and hence their unique features; which allows to integrate both physics and fidelity variations. The reason is that aggregation of physics variability is expected to result in better prediction in climate models [15]. E.g., Yang et al. [45] observed that RRTMG radiation scheme tends to overestimate precipitation, while CAM tends to underestimate precipitation, given the default calibration values. An inexpensive but accurate surrogate model is of great interest because WRF requires several days to run. Also, it is of great interest to quantify the uncertainty ranges and identify the optimal values of the five key calibration parameters in the KF CPS used in WRF. Here, the calibration parameters have the same physical interpretation, however they may depend on the grid spacing. Therefore, it is of interest to conduct joint inference on these parameters across the models RRTMG25 and CAM25, and separately by the model RRTMG50.

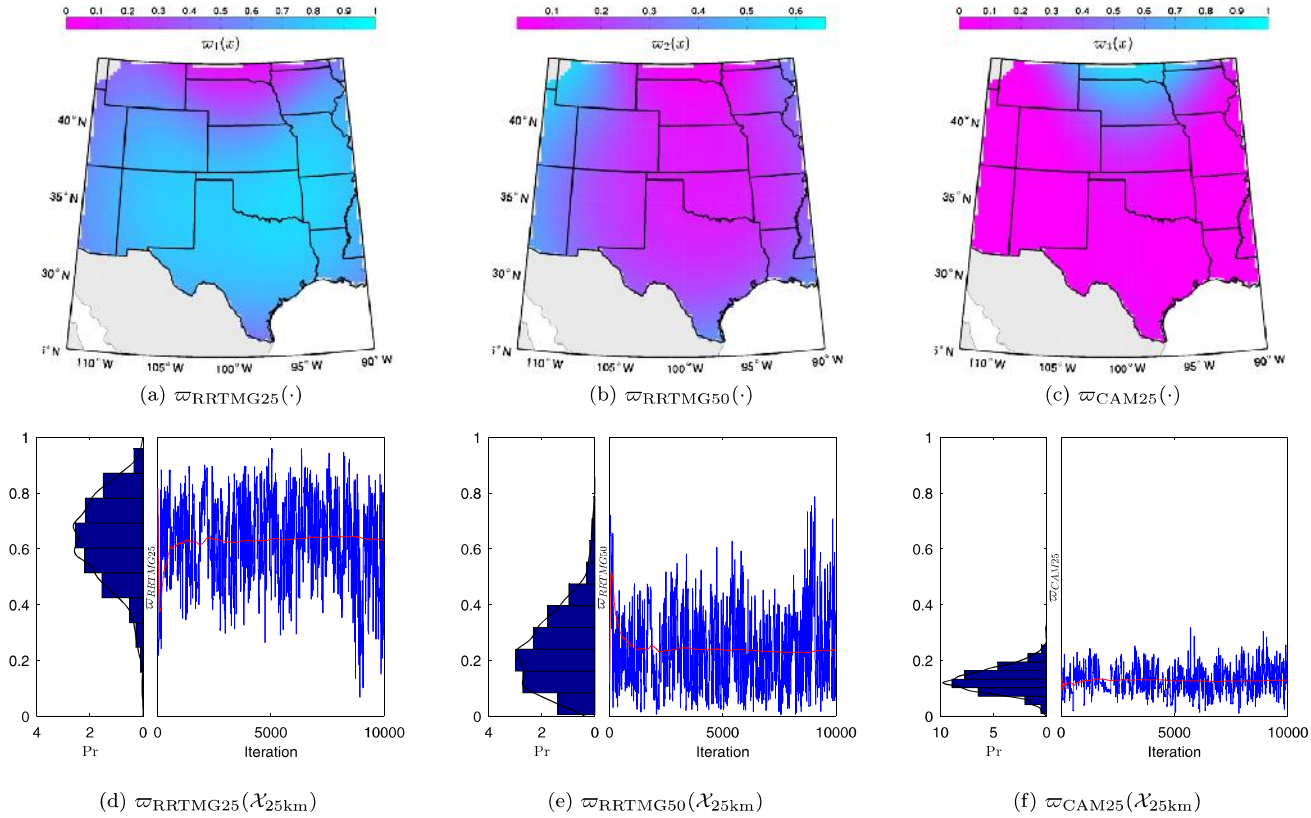
Training data Experimental data consist of 404 measurements from stations in the geographical domain 25°–44°N and 112°–90°W over the SGP region, and represent monthly average precipitation (in mm) in June 2007. The dataset is available from the U.S. Historical Climatological Network repository[5] [52].

Computer simulations were conducted over the same region by running the computer models: RRTMG25, CAM25, and RRTMG50 with specific configurations. The designs of RRTMG25, CAM25, and RRTMG50, consist of 50 simulations at different sets of calibration parameter values for each model, as well as at 4848, 4848, and 1211 coordinates on 25 km, 25 km, and 50 km grid spacing correspondingly. Briefly, WRF simulations for each computer model were driven by the 32 km North American Regional Reanalysis (NARR), and lateral boundary conditions were updated every 3 hours. The first simulation was initialized on May 1st, 2007 and run for 1 month with the standard KF scheme until June 1st. Afterwords, all generated ensembles ran for another month through June 2007, using identical initial land surface conditions from the first simulation on June 1st. Atmospheric conditions were reinitialized by using the NARR data every 2 days in all simulations in order to minimize the potential effects of error in the simulated large-scale circulation and isolate the impact of convective parametrization scheme on precipitation. Each simulation was run for 3 days, but the first day was discarded as model spin-up. Since we are interested in the averaged precipitation, all the ensembles were averaged out with respect to the time. Therefore our analysis represents an average of 15 two-day ensembles (totaling 1 month).

The validation data-set in order for us to assess the performance of the method consist of the post-processed University of Washington 1/8 gridded precipitation data [53] which are very accurate.

Uncertainty quantification analysis We consider the proposed Bayesian calibration computer model mixture set-up with non-informative priors. We transformed the precipitation values to the log scale to compensate for the positive values. The means of the Gaussian process priors assigned on the computer models output functions are modeled as 2nd degree multivariate Legendre polynomial bases expansions, while that of the discrepancy function is modeled as a constant. Because the application involves a large data-set, we tapered the covariance functions by using the Wendland-1 tapering function [54,33, Chapter 9]. Through try-and-error runs, we found that an acceptable value for the tapering parameter γ_W is 0.1 of the range, that do not cause a significant loss in the explanation of the variability. The reason is because small scale variabilities can be modeled by the compactly supported covariance function, while the larger scale variabilities can be explained by the bases expansion in the linear term of the Gaussian process [55,56]. Regarding the weight functions, we considered a pool of 2nd degree multivariate Legendre polynomial bases. We consider shared calibration parameters for computer models RRTMG25 and CAM25 as $(P_{\mathrm{d}}^{(25\ \mathrm{km})}, P_{\mathrm{e}}^{(25\ \mathrm{km})}, P_{\mathrm{t}}^{(25\ \mathrm{km})}, P_{\mathrm{h}}^{(25\ \mathrm{km})}, P_{\mathrm{c}}^{(25\ \mathrm{km})})$, and separate ones for computer models RRTMG50 as $(P_{\mathrm{d}}^{(50\ \mathrm{km})}, P_{\mathrm{e}}^{(50\ \mathrm{km})}, P_{\mathrm{t}}^{(50\ \mathrm{km})}, P_{\mathrm{h}}^{(50\ \mathrm{km})}, P_{\mathrm{c}}^{(50\ \mathrm{km})})$. On the calibration parameters, we assign independent truncated normal prior distributions whose hyper-parameters are specified through moment matching; and precisely by setting the prior means equal to the empirical values of the KF CPS scheme [45], and variances equal to the squared ranges. Namely, $P_{\mathrm{d}}^{(\ell)} \sim \mathrm{trN}(5.5 \cdot 10^{-9}, 122.37^2)$, $P_{\mathrm{e}}^{(\ell)} \sim \mathrm{trN}(5.5 \cdot 10^{-9}, 122.37^2)$, $P_{\mathrm{t}}^{(\ell)} \sim \mathrm{trN}(6.25, 507.68^2)$, $P_{\mathrm{h}}^{(\ell)} \sim \mathrm{trN}(175, 1.43^2 \cdot 10^{12})$, and $P_{\mathrm{c}}^{(\ell)} \sim \mathrm{trN}(3.37e+3, 7.23^2 \cdot 10^{12})$, for $\ell = 25$ km, 50 km. For comparison reasons, we consider the traditional Bayesian single model calibration of Higdon et al. [2] using the same specifications. The MCMC samplers ran for 20000 iterations where the first 10000 where discarded as burn in.

[5] http://www.ncdc.noaa.gov/oa/climate/research/ushcn/.



(a) $\varpi_{\mathrm{RRTMG25}}(\cdot)$ (b) $\varpi_{\mathrm{RRTMG50}}(\cdot)$ (c) $\varpi_{\mathrm{CAM25}}(\cdot)$

(d) $\varpi_{\mathrm{RRTMG25}}(\mathcal{X}_{25\mathrm{km}})$ (e) $\varpi_{\mathrm{RRTMG50}}(\mathcal{X}_{25\mathrm{km}})$ (f) $\varpi_{\mathrm{CAM25}}(\mathcal{X}_{25\mathrm{km}})$

Fig. 3.8. (a–c): Estimate of the posterior weight function $\{\varpi_k(\cdot); k = \mathrm{RRTMG25}, \mathrm{RRTMG50}, \mathrm{CAM25}\}$. (d–f): Histograms and trace plots of the integrated posterior weights $\{\varpi_k(\mathcal{X}_{25\ \mathrm{km}})$ computer on a 25 km space grid $\mathcal{X}_{25\ \mathrm{km}}$. Samples are generated by Algorithm 1 (Section 3.3).

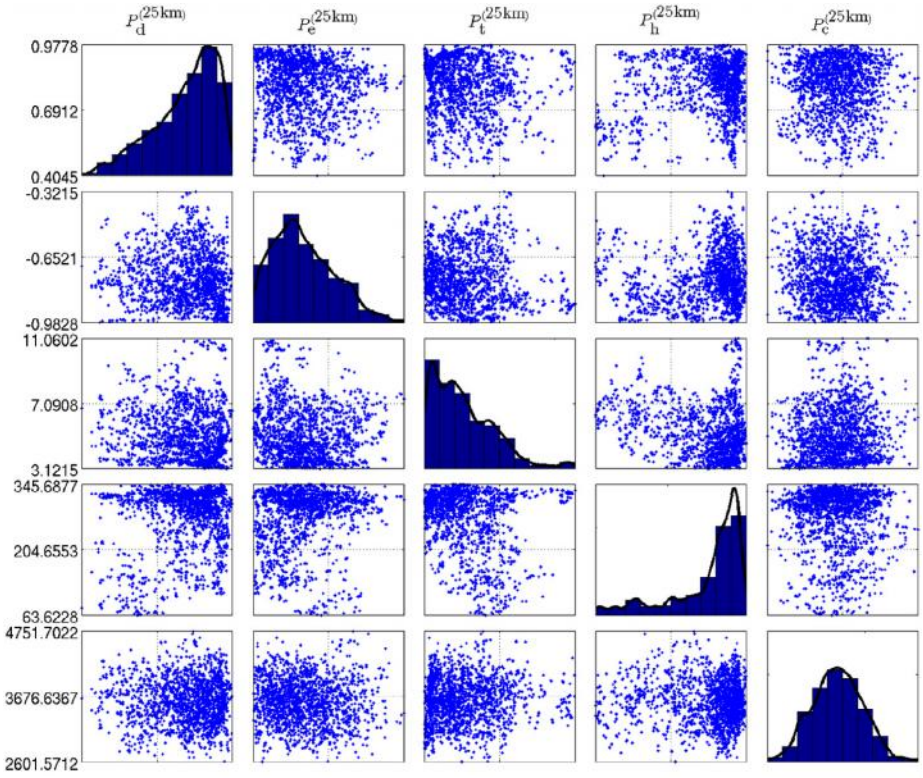
Table 1
Monte Carlo MAP and posterior mean estimates (with MC standard errors) of the 5 calibration parameters in the KF-CPS. The estimates are computed based on the MCMC sample generated (Section 3.3).

Calibration parameter	Posterior average est.		MAP est.	
	$\ell = 25$ km	$\ell = 50$ km	$\ell = 25$ km	$\ell = 50$ km
$P_{\mathrm{d}}^{(\ell)}$	0.8005 $(1.28 \cdot 10^{-5})$	0.7560 $(1.27 \cdot 10^{-5})$	0.8875	0.9430
$P_{\mathrm{e}}^{(\ell)}$	−0.7620 $(1.41 \cdot 10^{-5})$	−0.7746 $(1.28 \cdot 10^{-5})$	−0.8120	−0.9504
$P_{\mathrm{t}}^{(\ell)}$	5.1525 $(8.61 \cdot 10^{-5})$	6.5938 $(8.11 \cdot 10^{-5})$	4.1234	9.1657
$P_{\mathrm{h}}^{(\ell)}$	274.53 $(2.87 \cdot 10^{-3})$	297.18 $(2.81 \cdot 10^{-3})$	337.97	300.76
$P_{\mathrm{c}}^{(\ell)}$	3605.00 $(3.60 \cdot 10^{-2})$	3848.15 $(4.58 \cdot 10^{-2})$	3433.75	3887.49

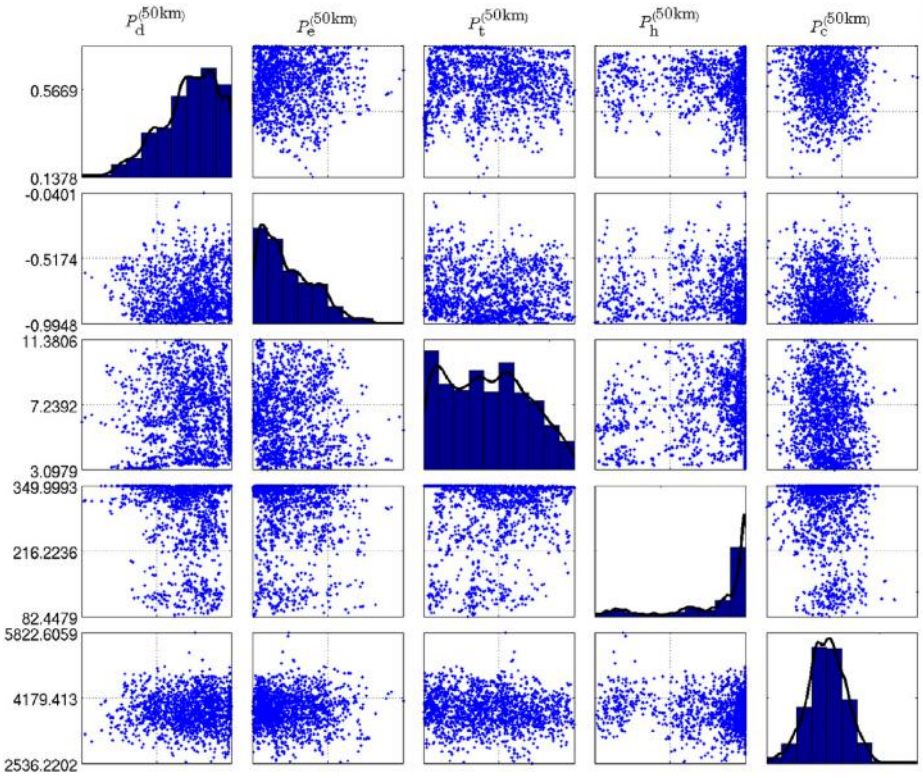
We perform Bayesian inference on the mixture weight function and present the results in Fig. 3.8. In Figs. 3.8a, 3.8b, 3.8c, we observe that the RRTMG25 model tends to outperform the other two models in most of the regions while CAM25 tends to outperform the other two models around the areas of South Dakota and Nebraska, in terms of the representation of the precipitation. The overall contribution of the computer models in the mixture is indicated by the a posteriori integrated weight functions (Figs. 3.8d, 3.8e, 3.8f). We observe that the posterior density of $\varpi_{\mathrm{RRTMG25}}(\mathcal{X})$ is over larger values than the others and without any significant overlapping, which indicates that, overally, RRTMG25 outperforms the other two. The associated trace plots suggest that the MCMC mixing was acceptable.

It is important to better understand the parameters of KF-CPS and constraint their ranges for future studies. Fig. 3.9 presents histogram estimates of marginal calibration parameter posterior densities, and scatter plots of the generated calibration parameter values. We observe that the calibration parameter posteriors, in the two grid spacing cases 25 km and 50 km, do not differ significantly, with the only exception of P_{t}. Moreover, we observe that the posterior densities of KF-CPS parameters are concentrated around narrower ranges than the default ones. In Table 1, we report the Monte Carlo average estimates, their standard errors and the MAP estimates of the calibration parameters as produced by the proposed method.

We examine the predictive ability of the proposed method and compare it with those of the standard Bayesian single model calibration (BSMC) method of Kennedy and O'Hagan [1] (Fig. 3.10). Fig. 3.10a presents the predicted precipitation computed according to the proposed method. Fig. 3.10b presents the relative absolute error computed as $\mathrm{RAE}(x) = |1 - \hat{\zeta}(x)/y_{\mathrm{valid.}}(x)|$, $x \in \mathcal{X}_{25\,\mathrm{km}}$, against the validation data $\{y_{\mathrm{valid.}}\}$. We observe that the proposed procedure can provide reliable

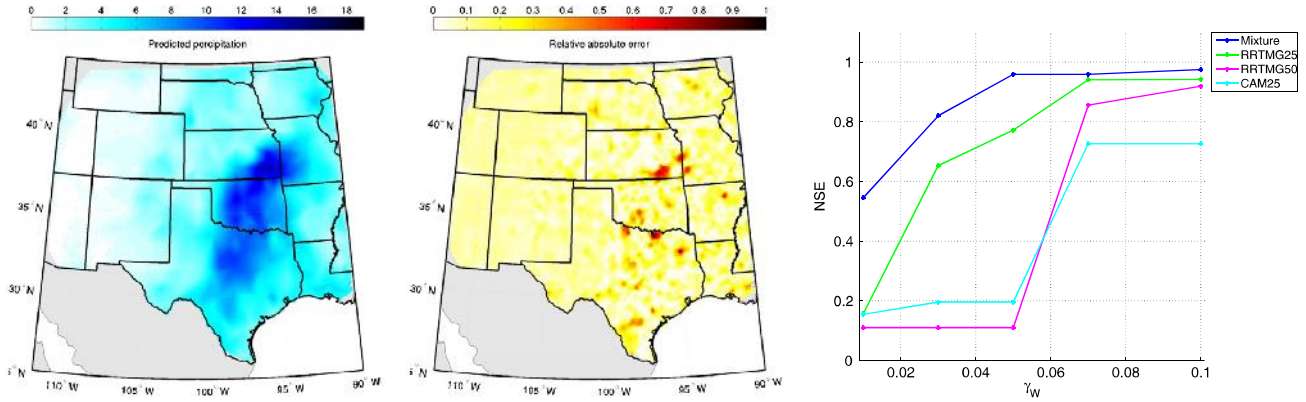


(a) 25Km case



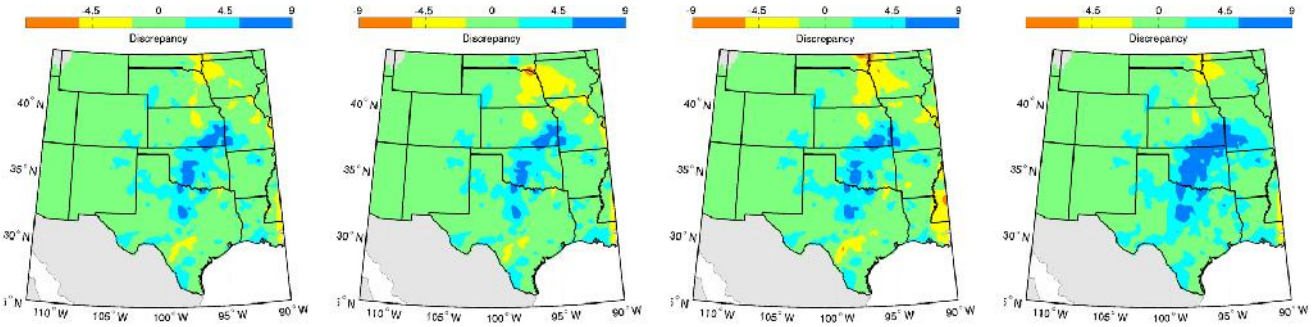
(b) 50Km case

Fig. 3.9. Calibration parameters of the KF-CPS. Estimated posterior densities, and generated MCMC samples of calibration parameters (Section 3.3).



(a) Predicted precipitation (b) Relative absolute error (c) NSE

Fig. 3.10. (a–b): Response surface of the predicted precipitation, and associate RAE, produced by the proposed method, against the validation data. (c): NSE of the proposed method (Mixture), and the BSMC on single models (RRTM25, RRTM50, and CAM25), with respect to the tapering parameter γ_W (Section 3.3).



(a) Mixture (b) BSMC, RRTM25 (c) BSMC, RRTM50 (d) BSMC, CAM25

Fig. 3.11. Discrepancy functions of the mixture model calibrated by the proposed method, and the single models RRTM25, RRTM50, and CAM25 calibrated by BSMC. The average absolute discrepancy are: 1.16 for Mixture, 1.18 for RRTM25, 1.27 for RRTM50, 1.51 for CAM25 (Section 3.3).

surrogate models for quick prediction of the precipitation since the RAE is acceptably low throughout the input domain. In Fig. 3.10c, we provide comparisons with respect to Nash–Sutcliffe model efficiency (NSE),[6] against the validation data, and for a set of different values of the tapering parameter γ_W. NSE is the average of the NSE produced from 4 independent realizations for each approach. We observe that the proposed method has better predictive ability compared to the BSMC method that uses only single models, for any value of γ_W considered, and that the associated NSE increases with γ_W. The observed difference in the performance appears to be more significant for more aggressive tapering (lower values of γ_W), and in favor of the proposed method.

Fig. 3.11 shows the discrepancy function of the calibrated computer model mixture produced by the proposed method, and those of the single models (RRTM25, RRTM50, and CAM25) produced by the BSMC method. The discrepancy functions were computed by approximating the computer model output functions through Kriging. We observe that the discrepancy function associated to the proposed method is smaller than that of RRTM25 and RRTM50 produced by BSMC, in regions northward 40°N. This is possibly because CAM25, which appears to be more accurate in sub-region northward 40°N, dominates RRTM25 and RRTM50 in the mixture in this sub-region. The mean absolute discrepancies, averaged out on the 25 km grid, are 1.16 for the computer model mixture (calibrated by our method), 1.18 for RRTM25 (calibrated by BSMC), 1.27 for RRTM50 (calibrated by BSMC), 1.51 for CAM25 (calibrated by BSMC). Therefore, the discrepancy of the mixture of computer models calibrated by our method is smaller than those of the single computer models calibrated by BSMC in most of the spatial space. This suggests that the computer model mixture can lead to more accurate simulations. In WRF application, an important reason that the computer model mixture outperforms the single models is because outputs from single models tend to differ from the real perspiration values at different directions. The weighting mechanism of the computer model mixture eliminates such discrepancies, and allows the mixture to have better predictive performance than the single ones.

[6] $\text{NSE} = 1 - \sum_{x\in\mathcal{X}_{\text{grid}}} (\hat{\zeta}_i(x) - y(x))^2 / \sum_{x\in\mathcal{X}_{\text{grid}}} (y(x) - \bar{y})^2$ where $\bar{y} = \sum_{x\in\mathcal{X}_{\text{grid}}} y(x)/\text{Card}(\mathcal{X}_{\text{grid}})$, and $\mathcal{X}_{\text{grid}}$ is a set of gridded points in the input domain $\mathcal{X}$. Larger values imply better prediction.

4. Conclusions and extensions

We proposed the Bayesian calibration of computer model mixture framework that extends the traditional Bayesian (single) model calibration. It builds a predictive model for the output of a real system by weighting, combining, and properly calibrating all the available computer models. The method allows to fit a calibrated mixture of computer models able to represent the real system more accurately since it aggregates unique features from different models. This allows the domain scientist to combine and weight the available computer models (simulators) to generate more accurate simulations. The method is suitable to address realistic problems that one model may be more accurate than the others at different input regions, due to the input dependent mixture weights. It is a suitable choice for a large number of real applications where the outputs of the available computer models fluctuate around the output of the real system. The procedure recovers the unknown weight functions by stochastically selecting significant bases from a pool of given bases functions in a data-driven manner. The estimated weight functions can provide a mean to rank the models at different inputs. Inference on the calibration parameters can be based on multiple computer models (and hence different physics) properly weighted. The proposed method does not require any knowledge of the fidelity order of the available models, however any available information can be taken into account through the prior model. It allows the use of a simple technique (presented in Appendix B) that mitigates the computational overhead to invert $\Sigma_z^{\otimes K}$ which is caused by the consideration of multiple computer models.

The proposed method was applied to a large-scale climate modeling application of the Weather Research and Forecasting with the Kain–Fritsch convective parametrization scheme that involved multiple computer models, based on different physical theories and levels of fidelity. Our UQ analysis produced a calibrated computer mixture model which was observed to lead to more reliable simulations than the single models calibrated via the traditional Bayesian model calibration of Kennedy and O'Hagan [1]. Yet, it produced an efficient surrogate model for the average monthly precipitation which outperforms those produced by the traditional single model calibration method. We observed that the WRF with the RRTMG radiation scheme and 25 km grid spacing outperforms the others in the representation of the real system output in the largest part of the spatial domain. Our analysis produced valuable information about KF-CPS for future studies, because the resulted posterior densities for the KF-CPS parameters concentrated on narrower ranges than the original. Our comparison study showed that the proposed method outperforms the standard Bayesian model calibration method of Kennedy and O'Hagan [1] if multiple models are available. Moreover, in the special case that the fidelity order is known, the proposed method can be a reliable counterpart of the multi-fidelity method of Goh et al. [11].

The method can be extended towards the sequential design of experiments with multiple models to allow the adaptive selection of designs by using the mixture weights as a guide. An interesting extension would be to consider multi-output computer models by coupling the method with that of Bilionis et al. [57]. Another important extension would be towards the Bayesian optimization by using ideas of Perdikaris and Karniadakis [58].

Acknowledgements

This work is supported NSF Grants DMS-1555072. The simulations were performed at National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory.

Appendix A. Adaptive Metropolis–Hastings transitions

The adaptive RWM and the HRMH algorithms that simulate from distribution $\pi(x)$, $x \in \mathrm{R}^d$, are given as a pseudo-codes in Algorithms 3 and 4. The Sampling step simulates a Metropolis transition probability targeting $\pi(x)$. The Adaptive step adjusts the unknown scale parameter of the proposal so that the expected acceptance probability to be equal to α_{opt}. RWM and HRMH perform well in terms of integrated autocorrelation time for $\alpha_{\text{opt}} \approx 0.234$ [59]. Here, we use $\gamma_t = (1/t)^{\varsigma}$, $\varsigma = 0.6$ which satisfies the required conditions for the generated Markov chain to be ergodic [38]. In Algorithm 1, we used log and logit transformations, to simulate from the full conditional distributions defined on the constrained spaces $x \in (0, \infty)$, and $x \in (L, U)$.

Algorithm 3 Random walk Metropolis transition, with an adaptive scheme.

Given that the current state of the Markov chain is at x_t, and the scale of the proposal has value σ_t:

Sampling step

1. Compute proposed value x' as $x' = x_t + \sigma_t z$, where $z \sim \mathrm{N}(0, \mathbb{I}_d)$.
2. Accept x' as the next state of the Markov chain with prob. $\alpha = \min(1, \frac{\pi(x')}{\pi(x)})$.

Adaptive step

Adjust the scale of the proposal such as $\log(\sigma_{t+1}) = \log(\sigma_t) + \gamma_t(\alpha_t - \alpha_{\text{opt}})$.

Algorithm 4 Hit & run Metropolis–Hastings transition, with an adaptive scheme.

Given that the current state of the Markov chain is at x_t, and the scale of the proposal has value σ_t:

Sampling step

1. Compute proposed value x'
   as $x' = x_t + \sigma_t z e$, where $z \sim \mathrm{N}(0,1)$, and e is draw from a unit d-dimensional space.
2. Accept x' as the next state of the Markov chain with prob. $\alpha = \min(1, \frac{\pi(x')}{\pi(x)})$.

Adaptive step

Adjust the scale of the proposal such as $\log(\sigma_{t+1}) = \log(\sigma_t) + \gamma_t(\alpha_t - \alpha_{\text{opt}})$.

Appendix B. A numerical technique to perform computations under the presence of large number of computer models

We present a convenient technique, suitable for the proposed mixture model framework, that mitigates the computational cost caused by the consideration of multiple computer models, when the Cholesky factorization of $\Sigma_z^{\otimes K}$ is required in order to evaluate the likelihood. Because of the consideration of multiple computer models the size of $\Sigma_z^{\otimes K}$ may increase so that matrix operations requiring Cholesky decomposition of $\Sigma_z^{\otimes K}$ become prohibitively expensive; this is because Cholesky decomposition scales as $O(\cdot^3)$ with the matrix size. The suggested technique takes advantage of the block-sparse structure of $\Sigma_z^{\otimes K}$, in order to perform these computations faster; hence it is tailored to the proposed method. It is particularly useful in the cases that the researcher has access only to standard linear algebra libraries.

Inverting a matrix, such as $\Sigma_z^{\otimes K}$, directly can be unstable or too expensive, and hence solvers of linear systems (e.g. $\Sigma_z^{\otimes K} x = z^{\otimes K}$, $\Sigma_z^{\otimes K} x = H_z^{\otimes K}$) may be used instead. A typical approach to solve $\Sigma_z^{\otimes K} x = b$ is to find an appropriate permutation matrix P and:

1. compute the lower matrix $\tilde{L}$ of the Cholesky decomposition of $P\Sigma_z^{\otimes K} P^\intercal$,
2. solve $\tilde{L} y = Pb$ for y,
3. solve $\tilde{L} z = y$ for z,
4. compute $x = P^\intercal z$ to obtain the solution.

Moreover, $\det(\Sigma_z^{\otimes K}) = \det(\tilde{L}_z^{\otimes K})^2$. Interest lies in finding P that leads to computational savings.

Let $P = \text{antidiag}(\mathbb{I}_n, \mathbb{I}_{m_1}, ..., \mathbb{I}_{m_K})$ be the permutation matrix,[7] $\tilde{\Sigma}_z^{\otimes K} = P\Sigma_z^{\otimes K} P^\intercal$ be the rotated covariance matrix, and $\tilde{L}_z^{\otimes K}$ be the lower matrix of the Cholesky decomposition of $\tilde{\Sigma}_z^{\otimes K}$. Then $\tilde{\Sigma}_z^{\otimes K} = P\Sigma_z^{\otimes K} P^\intercal$ is a symmetric and sparse arrowhead matrix such that

$$\tilde{\Sigma}_z^{\otimes K} = \begin{bmatrix} \Sigma_z^{(K,K)} & & & \Sigma_z^{(K),\intercal} \\ & \ddots & & \vdots \\ & & \Sigma_z^{(1,1)} & \Sigma_z^{(1),\intercal} \\ \Sigma_z^{(K)} & \cdots & \Sigma_z^{(1)} & \Sigma_z \end{bmatrix}.$$

According to the block Cholesky decomposition, the lower matrix $\tilde{L}_z^{\otimes K}$ is

$$\tilde{L}_z^{\otimes K} = \begin{bmatrix} L_z^{(K,K)} & & & \\ & \ddots & & \\ & & L_z^{(1,1)} & \\ L_z^{(K)} & \cdots & L_z^{(1)} & L_z \end{bmatrix},$$

where $\{L_z^{(k,k)}\}$ are the lower matrices of the Cholesky decomposition of $\{\Sigma_z^{(k,k)}\}$, L_z is the lower matrix of the Cholesky decomposition of $\Sigma_z - \sum_{k=1}^K L_z^{(k),\intercal} L_z^{(k)}$, and $\{L_z^{(k)} = \Sigma_z^{(k)} L_z^{(k,k),-1}\}$.

This technique allows the faster computation of Eqs. (2.11), (2.12), and (2.14), because computing $\tilde{L}_z^{\otimes K}$ as above can be faster than performing standard Cholesky decomposition directly on $\Sigma_z^{\otimes K}$, when K is large enough. This is because the complexity of the former procedure is $O(2 \cdot (K+1) \cdot n_{\max}^3)$, where $n_{\max} = \max(n, m^{(1)}, ..., m^{(K)})$ while that of the latter one is $O(n^{\otimes K,3})$. This can be shown by considering that the procedure requires $K+1$ Cholesky decompositions with complexity $O(\cdot^3)$, K forward substitutions with $O(\cdot^2)$, and $K+1$ multiplications with $O(\cdot^3)$. If faster decomposition or multiplication algorithms are applied, the complexity will be reduced accordingly. Further computational savings can be achieved if parallel computing environment is available because pairs of sub-matrices $\{(L_z^{(k)}, L_z^{(k,k)}); k = 1, ..., K\}$ can be computed in parallel for all k.

[7] As $\text{antidiag}(\mathbb{I}_n, \mathbb{I}_{m_1}, ..., \mathbb{I}_{m_K})$, we denote the matrix with blocks $\{\mathbb{I}_n, \mathbb{I}_{m_1}, ..., \mathbb{I}_{m_K}\}$ in the anti-diagonal (row-wise) and zeros elsewhere.

References

[1] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc., Ser. B, Stat. Methodol. 63 (2001) 425–464.
[2] D. Higdon, M. Kennedy, J.C. Cavendish, J.A. Cafeo, R.D. Ryne, Combining field data and computer simulations for calibration and prediction, SIAM J. Sci. Comput. 26 (2004) 448–466.
[3] A. O'Hagan, J. Kingman, Curve fitting and optimal design for prediction, J. R. Stat. Soc., Ser. B, Methodol. (1978) 1–42.
[4] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2005.
[5] B.A. Konomi, G. Karagiannis, K. Lai, G. Lin, Bayesian Treed Calibration: an application to carbon capture with AX sorbent, J. Am. Stat. Assoc. (2016), http://dx.doi.org/10.1080/01621459.2016.1190279.
[6] C.B. Storlie, W.A. Lane, E.M. Ryan, J.R. Gattiker, D.M. Higdon, Calibration of computational models with categorical parameters and correlated outputs via bayesian smoothing spline anova, J. Am. Stat. Assoc. (2014).
[7] R.K. Wong, C.B. Storlie, T. Lee, A frequentist approach to computer model calibration, arXiv preprint arXiv:1411.4723, 2014.
[8] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, J. Am. Stat. Assoc. 103 (2008).
[9] K.S. Bhat, D.S. Mebane, C.B. Storlie, P. Mahapatra, Upscaling uncertainty with dynamic discrepancy for a multi-scale carbon capture system, arXiv preprint arXiv:1411.2578, 2014.
[10] D. Higdon, J. Gattiker, E. Lawrence, C. Jackson, M. Tobis, M. Pratola, S. Habib, K. Heitmann, S. Price, Computer model calibration using the ensemble Kalman filter, Technometrics 55 (2013) 488–500.
[11] J. Goh, D. Bingham, J.P. Holloway, M.J. Grosskopf, C.C. Kuranz, E. Rutter, Prediction and computer model calibration using outputs from multifidelity simulators, Technometrics 55 (2013) 501–512.
[12] W.C. Skamarock, J.B. Klemp, J. Dudhia, D.O. Gill, M. Barker, K.G. Duda, X.Y. Huang, W. Wang, J.G. Powers, A Description of the Advanced Research WRF Version 3, Technical Report, National Center for Atmospheric Research, 2008.
[13] R. Pincus, H.W. Barker, J.-J. Morcrette, A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields, J. Geophys. Res., Atmospheres 1984–2012 108 (2003).
[14] W.D. Collins, P.J. Rasch, B.A. Boville, J.J. Hack, J.R. McCaa, D.L. Williamson, J.T. Kiehl, B. Briegleb, C. Bitz, S. Lin, et al., 2004, Description of the NCAR community atmosphere model (CAM 3.0).
[15] J. Hacker, S.-Y. Ha, C. Snyder, J. Berner, F. Eckel, E. Kuchera, M. Pocernich, S. Rugg, J. Schramm, X. Wang, The US Air Force Weather Agency's mesoscale ensemble: scientific description and performance results, Tellus A 63 (2011) 625–641.
[16] J.S. Kain, The Kain–Fritsch convective parameterization: an update, J. Appl. Meteorol. 43 (2004) 170–181.
[17] R.B. Gramacy, H.K. Lee, Cases for the nugget in modeling computer experiments, Stat. Comput. 22 (2012) 713–722.
[18] P. Chen, N. Zabaras, I. Bilionis, Uncertainty propagation using infinite mixture of gaussian processes and variational bayesian inference, J. Comput. Phys. 284 (2015) 291–333.
[19] W. Li, G. Lin, An adaptive importance sampling algorithm for bayesian inversion with multimodal distributions, J. Comput. Phys. 294 (2015) 173–190.
[20] C.K. Williams, D. Barber, Bayesian classification with gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1342–1351.
[21] X. Wan, G.E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM J. Sci. Comput. 28 (2006) 901–928.
[22] R. Courant, D. Hilbert, Methods of Mathematical Physics, John Wiley & Sons, 1953.
[23] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 230 (2011) 3015–3034.
[24] X. Yang, G.E. Karniadakis, Reweighted ℓ_1 minimization method for stochastic elliptic differential equations, J. Comput. Phys. 248 (2013) 87–108.
[25] G. Karagiannis, G. Lin, Selection of polynomial chaos bases via bayesian model uncertainty methods with applications to sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 259 (2014) 114–134.
[26] G. Karagiannis, B.A. Konomi, G. Lin, A bayesian mixed shrinkage prior procedure for spatial–stochastic basis selection and evaluation of GPC expansions: applications to elliptic SPDEs, J. Comput. Phys. 284 (2015) 528–546.
[27] J. Shi, B. Wang, Curve prediction and clustering with mixtures of gaussian process functional regression models, Stat. Comput. 18 (2008) 267–283.
[28] O. Le Maıtre, O. Knio, H. Najm, R. Ghanem, Uncertainty propagation using Wiener–Haar expansions, J. Comput. Phys. 197 (2004) 28–57.
[29] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. (1989) 409–423.
[30] C. Linkletter, D. Bingham, N. Hengartner, D. Higdon, Q.Y. Kenny, Variable selection for gaussian process models in computer experiments, Technometrics 48 (2006).
[31] N. Cressie, Statistics for Spatial Data, Wiley Series in Probability and Statistics, Wiley, New York, NY, USA, 1993.
[32] C. Paciorek, M. Schervish, Nonstationary covariance functions for gaussian process regression, Adv. Neural Inf. Process. Syst. 16 (2004) 273–280.
[33] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2004, Cambridge Books Online.
[34] P. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (1995) 711–732.
[35] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109.
[36] V.F. Turchin, On the computation of multidimensional integrals by the Monte-Carlo method, Theory Probab. Appl. 16 (1971) 720–724.
[37] R.L. Smith, Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions, Oper. Res. 32 (1984) 1296–1308.
[38] C. Andrieu, J. Thoms, A tutorial on adaptive MCMC, Stat. Comput. 18 (2008) 343–373.
[39] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, 2nd ed., Springer, 2004.
[40] G.O. Roberts, J.S. Rosenthal, et al., General state space Markov chains and MCMC algorithms, Probab. Surv. 1 (2004) 20–71.
[41] A. O'Hagan, J.M. Bernardo, J.O. Berger, A.P. Dawid, A.F.M. e. Smith, M.C. Kennedy, J.E. Oakley, Uncertainty Analysis and Other Inference Tools for Complex Computer Codes (with Discussion), Oxford University Press, Oxford, 1999.
[42] M.C. Kennedy, A. O'Hagan, Supplementary Details on Bayesian Calibration of Computer Models, Technical Report, Internal Report. URL http://www.shef.ac.uk/~st1ao/ps/calsup.ps, 2001.
[43] R. Cook, D. Malkus, M. Plesha, Concepts and Applications of Finite Element Analysis, Wiley, 1989, https://books.google.com/books?id=irZHPgAACAAJ.
[44] W.J.C.M.D. McKay, R.J. Beckman, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.
[45] B. Yang, Y. Qian, G. Lin, R. Leung, Y. Zhang, Some issues in uncertainty quantification and parameter tuning: a case study of convective parameterization scheme in the WRF regional climate model, Atmos. Chem. Phys. 12 (2012) 2409.
[46] M.S. Gilmore, J.M. Straka, E.N. Rasmussen, Precipitation uncertainty due to variations in precipitation particle parameters within a simple microphysics scheme, Mon. Weather Rev. 132 (2004) 2610–2627.
[47] J.M. Murphy, B.B. Booth, M. Collins, G.R. Harris, D.M. Sexton, M.J. Webb, A methodology for probabilistic predictions of regional climate change from perturbed physics ensembles, Philos. Trans. R. Soc. Lond. A, Math. Phys. Eng. Sci. 365 (2007) 1993–2028.
[48] H. Morrison, J. Curry, V. Khvorostyanov, A new double-moment microphysics parameterization for application in cloud and climate models. Part I: Description, J. Atmos. Sci. 62 (2005) 1665–1677.
[49] F. Chen, J. Dudhia, Coupling an advanced land surface-hydrology model with the Penn State-NCAR MM5 modeling system. Part I: Model implementation and sensitivity, Mon. Weather Rev. 129 (2001) 569–585.
[50] Z.I. Janjić, Nonsingular implementation of the Mellor–Yamada level 2.5 scheme in the NCEP Meso model, NCEP office note 437 (2002) 61 pp.

[51] E.J. Mlawer, S.J. Taubman, P.D. Brown, M.J. Iacono, S.A. Clough, Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave, J. Geophys. Res., Atmospheres (1984–2012) 102 (1997) 16663–16682.
[52] T. Karl, C. Williams, F. Quinlan, T. Boden, United States Historical Climatology Network (HCN) Serial Temperature and Precipitation Data, Environmental Science Division, Publication No. 3404, Technical Report, Carbon Dioxide Information and Analysis Center, Oak Ridge National Laboratory, Oak Ridge, TN, 1990, 389 pp.
[53] E. Maurer, A. Wood, J. Adam, D. Lettenmaier, B. Nijssen, A long-term hydrologically based dataset of land surface fluxes and states for the conterminous United States*, J. Climate 15 (2002) 3237–3251.
[54] R. Furrer, M.G. Genton, D. Nychka, Covariance tapering for interpolation of large spatial datasets, J. Comput. Graph. Stat. (2006).
[55] H. Sang, J.Z. Huang, A full scale approximation of covariance functions for large spatial data sets, J. R. Stat. Soc., Ser. B, Stat. Methodol. 74 (2012) 111–132.
[56] C.G. Kaufman, D. Bingham, S. Habib, K. Heitmann, J.A. Frieman, Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology, Ann. Appl. Stat. (2011) 2470–2492.
[57] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable gaussian process: towards an efficient, fully bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239.
[58] P. Perdikaris, G.E. Karniadakis, Model inversion via multi-fidelity bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond, J. R. Soc. Interface 13 (2016) 20151107.
[59] G.O. Roberts, A. Gelman, W.R. Gilks, Weak convergence and optimal scaling of random walk Metropolis algorithms, Ann. Appl. Probab. 7 (1997) 110–120.