Bayesian Treed Calibration: an application to carbon capture with AX sorbent

Bledar A. Konomi[*]
Department of Mathematical Science, University of Cincinnati
and
Georgios Karagiannis
Department of Mathematical Sciences, Durham University
and
Kevin Lai
Pacific Northwest National Laboratory
and
Guang Lin
Department of Mathematics, School of Mechanical Engineering, Purdue University

January 15, 2017

Abstract

In cases where field (or experimental) measurements are not available, computer models can model real physical or engineering systems to reproduce their outcomes. They are usually calibrated in light of experimental data to better reproduce the real system. Statistical methods, based on Gaussian processes, for calibration and prediction have been especially important when the computer models are expensive and experimental data limited. In this paper, we develop the Bayesian treed calibration (BTC) as an extension of standard Gaussian process calibration methods to deal with non-stationarity computer models and/or their discrepancy from the field (or experimental) data. Our proposed method partitions both the calibration and observable input space, based on a binary tree partitioning, into subregions where existing model calibration methods can be applied to link a computer model with the real system. The estimation of the parameters in the proposed model is carried out using Markov chain Monte Carlo (MCMC) computational techniques. Different strategies have been applied to improve mixing. We illustrate our method in two artificial examples and a real application that concerns the capture of carbon dioxide with AX amine based sorbents. The source code and the examples analyzed in this paper are available as part of the supplementary materials.

Keywords: Calibration, Bayesian tree, Gaussian process, Markov chain Monte Carlo, Computer Experiments

[*] Corresponding authors gratefully acknowledge

1 Introduction

The direct observation of a complex physical or engineering system (e.g., climate change, nuclear reactor performance, carbon capture system) for the whole input space usually is impossible due to the high cost, limited resources, rare events, etc. The need to replace a physical or engineering system with an accurate computer model is crucial for further investigation and better understanding of the problem. Accurate computer models (simulators) have been made possible by advances in processing power and parallel computing. To evaluate and use computer model to predict a physical or engineering system, it may initially be necessary to calibrate it using some observed data. Improving the level of detail and model resolution may increase the fidelity of simulations to real systems, yet the rise in the associated computation cost may be significant. In spite of the availability of powerful computational resources, it often is computationally too expensive to run such complex models for all possible input conditions.

Model calibration is an important tool in problems where there are expensive computer models and experimental (or field) data available from the true system. Typically, experimental data are collected from some region of the input space called observable input space. The remaining input space consist of the calibration parameter input space, which is associated with parameters necessary to run the computer model. A statistically rigorous model based framework for computer model calibration has been suggested by Kennedy and O'Hagan (2001), Higdon et al. (2004), Williams et al. (2006) and Vernon et al. (2010). These approaches provide a posterior distribution for the computer model and the discrepancy term. Moreover, this model can be used to make predictions about outcomes in regions of the input space where data are not available, as well as to accurately estimate the parameters that are usually not observed but are necessary to run the computer model. Often, there are multiple outcomes (e.g., temperature and pressure) of interest, and the field data measure, at least, some of them. An extension to high dimensional output via functional decomposition has been considered in Higdon et al. (2008) and Bayarri et al. (2007). Calibration with categorical parameters are described in Storlie et al. (2013). In Higdon et al. (2008) the discrepancy term is also modeled with a non-stationary process-convolution approach Higdon (1998) where the number and the locations and central locations are determined in an *ad hoc* manner. Well known limitation of process-convolution approach such as dimensionality problem and smooth transition over space also should be taken into consideration. Although, this calibration model has been applied successfully to practical problems, an overall calibration approach available that can appropriately handle a non-stationary computer model and/or discrepancy term is still in the search. The mean is usually modeled with a smooth function, and the covariance function is considered stationary. In practice, the mean, variation, and/or variance of the computer model and/or discrepancy may changes for different parts of the input space. The computer model and/or the experimental nugget effect may also depend on the input space.

The motivation for addressing the preceding problem comes from a project, associated with the Carbon Capture Simulation Initiative (CCSI), that is related to the effects of man-made greenhouse gases on the global environment, notably an important issue at the moment. Briefly, the task we are mostly interested in concerns modeling the capture of carbon dioxide using an amine sorbent. The amine sorbent, comprised of small chemically reactive particles flowing through the adsorber reactor device, is capable of reacting with carbon dioxide and removing it from thermal power plant exhaust. The properties of the sorbent used, coded name AX, and its CO_2 adsorption characteristics can be found in Krutka and Sjostrom (2013). The experimental data are limited in size because they are collected via a time-demanding experiment that involves complex multiphase reactive flow phenomena, such as hydrodynamics, thermal transfer, and chemical reaction, simultaneously. For similar reasons, it is equally challenging to design effective modeling simulations capable of solving all physics aspects. The computational cost of such computer models (simulations) is expected to be extremely high. In these types of studies, sudden changes and discontinuities in the output for both the real system or computer model may occur (explained later in Section 7). Consequently, standard statistical calibration methods that do not account for non-stationarity can give inaccurate results.

Bayesian treed models have been successful in handling non-stationarity for the purpose of uncertainty quantification and prediction in computer models (experiments). They partition the input space into non-overlapping subregions by making binary splits recursively using a Markov chain Monte Carlo (MCMC) algorithm. Each of these non-overlapping subregions corresponds to a terminal node of a binary tree. The output within a terminal node is considered "homogeneous" and modeled independently from the output of the other nodes. Within a terminal node, Chipman et al. (1998, 2002) propose a regression model while Gramacy and Lee (2008) generalize it to the Gaussian process. An extension to the multivariate case is given by Konomi et al. (2014a). Bayesian treed techniques provide a straightforward mechanism for creating a non-stationary model by applying simple stationary models within each terminal node. Moreover, it reduces the computational cost by applying simpler models to less data in every MCMC iteration.

Lee et al. (2010) uses the treed Gaussian process (TGP) to facilitate a two step model calibration. In the first step TGP is used to help an adaptive sampling procedure to find the most informative input points for the observed data similar to Gramacy and Lee (2009). In the second step TGP is used to help an optimization algorithm to minimize an objective function for the distance between simulated and observed data. The observed and the simulated input space have to be the same. In the minimization process the TGP is used to provide a more robust solution since the function is usually multi-modal. Also their model do not count for discrepancy.

In this paper, we use Bayesian treed techniques to propose a general approach for calibration in situations where: (a) a physical or engineering system is observed in some region of the input space, (b) the computer model(s) are computationally expensive to run and possibly have non-stationary output behavior, and (c) the discrepancy between the computer model and the real system may be non-stationary. We develop a Bayesian treed calibration (BTC) model which partitions the input space (both observable and calibration input space) into disjoint subregions where we can apply independent standard calibration models, such as those in Kennedy and O'Hagan (2001) and Higdon et al. (2004). The proposed likelihood depends on the tree structure, calibration parameters, and the terminal node standard calibration models. Appropriate choice of the likelihood and prior specification lead to a properly defined posterior distribution, where inference is carried out through an MCMC sampler. Special care is taken in the formulation of the problem when we propose $\boldsymbol{\theta}$ in different subregions and in the Bayesian tree operations (*grow, prune, change, swap* and *rotate*). In addition, the MCMC sampler has been facilitated by integrating out the linear term of the Gaussian processes used to model the computer model and the discrepancy in each terminal nodes. To perform prediction, we use Bayesian model averaging (BMA) (Hoeting et al., 1999) and use covariance functions that can link different subregions at each MCMC iteration. We also offer an extension of the proposed model when dealing with multiple outcomes. Artificial cases have been constructed to demonstrate the usefulness of the proposed BTC with existing calibration models. Finally, BTC is applied to calibrate computer model and real data in the adsorber device with AX amine based sorbents from a carbon capture plant where non-stationary models are supported from data. The proposed BTC method can be used in cases where the non-stationarity is associated to the computer model output, the discrepancy function, or the nugget effects.

The rest of the paper is organized as follows: in Section 2, we review the Bayesian calibration approach. In Section 3, we describe the Bayesian treed calibration (BTC). In Section 4, we describe the Bayesian inference and prediction strategies. In Section 5, we conduct a simulation study for artificial examples. In Section 6, we demonstrate the benefits of our proposed method by analyzing AX data. Conclusions are presented in Section 7.

2 Review on Bayesian Calibration

In this section, we present a brief review of the calibration problem similar to Kennedy and O'Hagan (2001), and Higdon et al. (2004).

2.1 Model calibration

The calibration data are composed of field (experimental) measurements and computer model data. We denote the response of the n field measurement by $\boldsymbol{z} = (z_1, \ldots, z_n)$, where each component z_i is being subject to:

$$z_i = \zeta(\boldsymbol{x}_i) + e_i,$$

where $\zeta(\boldsymbol{x}_i)$ denotes the response of the actual physical system, $\boldsymbol{x}_i$ are the observable inputs, and e_i denotes the nugget error for the i^{th} observation.

The computer model aims to simulate the real system. The input of the computer model consists of $q-$dimensional observable input $\boldsymbol{x}$ and $p-$dimensional calibration input values $\boldsymbol{t}$. Without loss of generality, we also use the term calibration parameters for the tuning parameters, which are necessary to run the computer model, as in Higdon et al. (2004). The computer model output is an unknown function $\eta(\boldsymbol{x}, \boldsymbol{t})$ of the observable and calibration input, which simulates a physical system.

The observations z_i, the true process $\zeta(\,\cdot\,)$, and the computer model function $\eta(\,\cdot\,,\,\cdot\,)$ are linked through:

$$z_i = \zeta(\boldsymbol{x}_i) + e_i = \eta(\boldsymbol{x}_i, \boldsymbol{\theta}) + \delta(\boldsymbol{x}_i) + e_i, \tag{1}$$

where the term $\delta(\,\cdot\,)$ is a model disagreement between the real system from the computer model, and it is usually known as the discrepancy function. The discrepancy function is assumed to be independent of the computer model true value $\eta(\,\cdot\,, \boldsymbol{\theta})$, and $\boldsymbol{\theta}$ denotes the best fixed but unknown setting for the calibration input $\boldsymbol{t}$. Each of the n field measurements $\eta(\,\cdot\,,\,\cdot\,)$ consists of the known observable input value $\boldsymbol{x}_i$, and the unknown $p-$dimensional vector calibration parameter $\boldsymbol{\theta}$, which is considered fixed (for each of the n field measurements). Direct sampling from the posterior of the calibration parameters $\boldsymbol{\theta}$ is impossible in practice because the computation of $\eta(\boldsymbol{x}, \boldsymbol{t})$ is often expensive. To overcome these difficulties, $\eta(\boldsymbol{x}, \boldsymbol{t})$ is modeled with a Gaussian process-based emulator (Kennedy and O'Hagan, 2001).

Although some computer models are generated by deterministic solvers with no random error, to avoid an infinite differentiability covariance function it is better to add a nugget effect in the statistical model (Stein, 1999). Moreover, Gramacy and Lee (2012) argue that the use of a nugget helps protect against poor fit when assumptions are violated. Statistical emulators for computer models are not exact in practice. The output of the computer model is $y = \eta(\boldsymbol{x}, \boldsymbol{t}) + v$, where v represents the nugget error of the computer model. For m computer experiment runs at input points $((\boldsymbol{x}_1^*, \boldsymbol{t}_1), \ldots, (\boldsymbol{x}_m^*, \boldsymbol{t}_m))$ (both observable and calibration input) we denote the output as $\boldsymbol{y} = (y_1, \ldots, y_m)$, where $y_j = \eta(\boldsymbol{x}_j^*, \boldsymbol{t}_j) + v_j$.

2.2 Calibration with Gaussian process

Typically, the unknown functions $\eta(\,\cdot\,,\,\cdot\,)$ and $\delta(\,\cdot\,)$ are modeled as two independent Gaussian processes (GP) (Kennedy and O'Hagan, 2001; Higdon et al., 2004; Williams et al., 2006; Higdon et al., 2008), that is: $\eta(\,\cdot\,,\,\cdot\,) \sim N(\mu_\eta(\,\cdot\,,\,\cdot\,), c_\eta((\,\cdot\,,\,\cdot\,),(\,\cdot\,,\,\cdot\,)))$ and $\delta(\,\cdot\,) \sim N(\mu_\delta(\,\cdot\,), c_\delta(\,\cdot\,,\,\cdot\,))$. We will refer to this method as standard Bayesian Gaussian process calibration (SBGPC). For $\eta(\,\cdot\,,\,\cdot\,)$ and $\delta(\,\cdot\,)$ the mean is usually assumed to be a linear model as: $\mu_\eta(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{h}_\eta(\boldsymbol{x}, \boldsymbol{t})^T \boldsymbol{\beta}_\eta$ and $\mu_\delta(\boldsymbol{x}) = \boldsymbol{h}_\delta(\boldsymbol{x})^T \boldsymbol{\beta}_\delta$. The covariance function of $\eta(\,\cdot\,,\,\cdot\,)$ is modeled in a separable form, as in Kennedy and O'Hagan (2001):

$$c_\eta((\boldsymbol{x}, \boldsymbol{t}), (\boldsymbol{x}', \boldsymbol{t}')) = \sigma_\eta^2 \rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\phi}_{\eta,x}) \rho(\boldsymbol{t}, \boldsymbol{t}'; \boldsymbol{\phi}_{\eta,t}),$$

and the covariance function of $\delta(\,\cdot\,)$ is:

$$c_\delta(\boldsymbol{x}, \boldsymbol{x}') = \sigma_\delta^2 \rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\phi}_{\delta,x}),$$

where σ_η^2 and σ_δ^2 are the variance of $\eta(\,\cdot\,,\,\cdot\,)$ and $\delta(\,\cdot\,)$, correspondingly, and ρ denotes the correlation function.

The correlation function, ρ, is of particular importance as it defines the smoothness of the random field. Different choices, such as Matérn and power exponential covariance family, can be made. The separable power exponential covariance family is considered as a standard choice in the computer experiments (Santner et al., 2003) where the dimensionality of the input can be usually high. In specific, for the squared exponential family $\rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\phi}_{\eta,x}) = \exp\left(-\frac{1}{2}\sum_{l=1:q} \frac{||\boldsymbol{x}_l - \boldsymbol{x}_l'||^2}{\phi_{l,x}^2}\right)$, where $\phi_{l,x}$ is the correlation strength in the l direction. Different inputs usually have different meaning. Therefore, it is preferable to have different correlation parameters. The same formulations can be applied for $\rho(\boldsymbol{t}, \boldsymbol{t}'; \boldsymbol{\phi}_{\eta,t})$ and $\rho(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\phi}_{\delta,x})$. When the data are sampled in a grid, this covariance matrix can be expressed in a Kronecker product of one-dimensional matrices. Finally, this correlation function is invariant to the value of x_i when considered along x_j when $i \neq j$.

3 The Bayesian treed calibration

In many calibration problems, a stationary model for the computer model or the discrepancy function may not be appropriate since the mean, variance, and the spatial dependency may differ from one input subregion to another. The Bayesian tree (Chipman et al., 1998) provides a straightforward mechanism for modeling non-stationary data by partitioning the input space into subregions using binary splitting rules. Each partition then is modeled independently.

3.1 Modeling with a binary tree both the computer model output and the discrepancy

For simplicity in the formulation, we denote as $\mathcal{D}$ the $(q+p)$ dimensional input space, both observational and calibration. We also denote the observable input points of the n field measurements by $\boldsymbol{D}^1 = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, and the set of observable input points augmented with the calibration parameter by $\boldsymbol{D}^1(\boldsymbol{\theta}) = \{(\boldsymbol{x}_1, \boldsymbol{\theta}), \ldots, (\boldsymbol{x}_n, \boldsymbol{\theta})\}$. In addition, we denote the set of input points (both observable and calibration) of the computer model by $\boldsymbol{D}^2 = \{(\boldsymbol{x}_1^*, \boldsymbol{t}_1), \ldots, (\boldsymbol{x}_m^*, \boldsymbol{t}_m)\}$. We also represent all of the input dataset augmented by θ as $\boldsymbol{D} = (\boldsymbol{D}^1(\boldsymbol{\theta}), \boldsymbol{D}^2)$ and their output as $\boldsymbol{d} = (\boldsymbol{z}, \boldsymbol{y})$.

We apply a binary tree $\mathcal{T}$ to partition the input space in $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ disjoint subregions, such that $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ corresponds to a tree structure $\mathcal{T}$ with K external nodes. Each subregion consists of $\boldsymbol{D}_k = \{\boldsymbol{D}_k^1(\boldsymbol{\theta}), \boldsymbol{D}_k^2\}$ input points and the corresponding output $\boldsymbol{d}_k = (\boldsymbol{z}_k, \boldsymbol{y}_k)$. We model the output of the k^{th} subregions $\{\boldsymbol{d}_k\}$ with the SBGPC (explained in Section 2). Inside each treed partition, the unknown functions $\eta_k(\,\cdot\,,\,\cdot\,)$ and $\delta_k(\,\cdot\,)$ are modeled as two independent GPs. The calibration model formulation is independent for each partition, and the created mean and the covariance parameters have a step function form. Moreover, for a given $\boldsymbol{\theta}$, some of the input subregions may not contain $\boldsymbol{\theta}$. In these cases, the Bayesian formulation with the prior distributions helps to define the model. In the Bayesian framework, for a given subregion k, we update the prior of $\eta_k(\,\cdot\,,\,\cdot\,)$ and $\delta_k(\,\cdot\,)$ in light of the data. For subregions that do not have experimental observations, the unknown function $\delta_k(\,\cdot\,)$ is updated from the prior distribution.

We chose to work with a unique Bayesian tree for the output $\boldsymbol{d}$. Another alternative is to separately model the computer model output $\eta(\,\cdot\,,\,\cdot\,)$ and the discrepancy function $\delta(\,\cdot\,)$ with a Bayesian treed Gaussian process (BTGP). Because we are modeling both functions independently from each other, it may be more appropriate to deal separately with $\eta(\,\cdot\,,\,\cdot\,)$ and $\delta(\,\cdot\,)$. However, this translates into two independent Bayesian treed models: one for the computer model and the other for its discrepancy from the real system. In spite of the logical flow of this approach, it would make the structure of the model very complicated. For more details see Appendix A. Instead, if we work with only one Bayesian tree directly to the response $\boldsymbol{d}$, the inference is more direct.

Likelihood: For a given tree $\mathcal{T}$, calibration parameter $\boldsymbol{\theta}$, and GP parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_k\}_{k=1:K} = \{\boldsymbol{\beta}_k, \boldsymbol{\phi}_k, \boldsymbol{\sigma}^2_k, \boldsymbol{\tau}^2_k\}_{k=1:K} = (\boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2)$, the likelihood is a product of K independent different components,

$$f(\boldsymbol{d}|\mathcal{T}, \boldsymbol{\Theta}, \boldsymbol{\theta}) \propto \prod_{k=1:K} |\boldsymbol{V}_{\boldsymbol{d}_k}|^{-1/2} \exp[-\frac{1}{2}(\boldsymbol{d}_k - E(\boldsymbol{d}_k))^T \boldsymbol{V}_{\boldsymbol{d}_k}^{-1}(\boldsymbol{d}_k - E(\boldsymbol{d}_k))],$$

where $E(\boldsymbol{d}_k)$ and $\boldsymbol{V}_{\boldsymbol{d}_k}$ are the mean and variance of the output in the k^{th} partition (subregion) $\boldsymbol{d}_k$. When $\boldsymbol{D}^1_k(\boldsymbol{\theta}) = \emptyset$, the likelihood of the k^{th} partition does not depend on $\boldsymbol{\theta}$. As a result, some of the partitions will not depend on the calibration parameter $\boldsymbol{\theta}$ and the parameters associated with the discrepancy.

In order to represent more explicitly the mean and variance of each component in the preceding formulation, we introduce some new symbols. Let $\boldsymbol{H}_\eta(\boldsymbol{D}^2_k)$ denote the matrix with rows $\boldsymbol{h}_\eta(\boldsymbol{x}, \boldsymbol{t})$ for each $(\boldsymbol{x}, \boldsymbol{t}) \in \boldsymbol{D}^2_k$, and $\boldsymbol{H}_\delta(\boldsymbol{D}^1_k)$ denotes the matrix with rows $\boldsymbol{h}_\delta(\boldsymbol{x})$ for each $\boldsymbol{x} \in \boldsymbol{D}^1_k$. Let $\boldsymbol{V}_\eta(\boldsymbol{D}_k) = \boldsymbol{C}_\eta(\boldsymbol{D}_k, \boldsymbol{D}_k)$ be the covariance matrix with (i, i') elements, $c_\eta((\boldsymbol{x}_i, \boldsymbol{t}_i), (\boldsymbol{x}_{i'}, \boldsymbol{t}_{i'})$ for every pair $(\boldsymbol{x}_i, \boldsymbol{t}_i) \in \boldsymbol{D}_k$, and $(\boldsymbol{x}_{i'}, \boldsymbol{t}_{i'}) \in \boldsymbol{D}_k$. Similarly, we define $\boldsymbol{V}_\delta(\boldsymbol{D}^1_k) = \boldsymbol{C}_\delta(\boldsymbol{D}^1_k, \boldsymbol{D}^1_k)$. The mean of the output in the k^{th} partition is:

$$E(\boldsymbol{d}_k) = \boldsymbol{H}_k \boldsymbol{\beta}_k = \begin{bmatrix} \boldsymbol{H}_\eta(\boldsymbol{D}^1_k(\boldsymbol{\theta})) & \boldsymbol{H}_\delta(\boldsymbol{D}^1_k) \\ \boldsymbol{H}_\eta(\boldsymbol{D}^2_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_{\eta,k} \\ \boldsymbol{\beta}_{\delta,k} \end{bmatrix},$$

and its covariance matrix is:

$$\boldsymbol{V}_{\boldsymbol{d}_k} = \mathrm{cov}(\boldsymbol{d}_k, \boldsymbol{d}_k) = \boldsymbol{V}_{\eta_k}(\boldsymbol{D}_k) + \begin{pmatrix} \boldsymbol{V}_{\delta_k}(\boldsymbol{D}^1_k) + \tau^2_{e_k}\boldsymbol{I}_{n_k} & 0 \\ 0 & \tau^2_{v_k}\boldsymbol{I}_{m_k} \end{pmatrix},$$

where $\tau^2_{e_k}$ and $\tau^2_{v_k}$ are the variances of the nuggets, and $\boldsymbol{I}_{n_k}$ and $\boldsymbol{I}_{m_k}$ are identity matrices of dimension $n_k \times n_k$ and $m_k \times m_k$ respectively.

Prior: We assign a prior distribution on the parameter $(\mathcal{T}, \boldsymbol{\theta}, \boldsymbol{\Theta})$, such as:

$$\pi(\mathcal{T}, \boldsymbol{\theta}, \boldsymbol{\Theta}) = \pi(\mathcal{T})\pi(\boldsymbol{\theta})\pi(\boldsymbol{\Theta}|\mathcal{T}) = \pi(\mathcal{T})\pi(\boldsymbol{\theta}) \prod_{k=1:K} \pi(\boldsymbol{\beta}_{\eta_k}, \sigma^2_{\eta_k})\pi(\boldsymbol{\phi}_{\eta_k})\pi(\boldsymbol{\beta}_{\delta_k}, \sigma^2_{\delta_k})\pi(\boldsymbol{\phi}_{\delta_k})\pi(\tau^2_{e_k})\pi(\tau^2_{v_k}).$$

In the Bayesian framework, a binary tree $(\mathcal{T})$ is treated as random and assigned with a prior distribution through a tree-generating process (Chipman et al., 1998). Starting with a null tree (all data in a single region), a leaf node $\xi \in \mathcal{T}$, representing a subregion of the input space, splits with probability $P_{\text{split}}(\xi, \mathcal{T}) = a(1 + u_\xi)^{-b}$, where u_ξ is the depth of $\xi \in \mathcal{T}$, a controls the balance of the shape of the tree, and b controls the size of the of the tree.

The marginal treed prior distribution is:

$$P(\mathcal{T}) = P_{rule}(\rho|\xi, \mathcal{T}) \prod_{\xi_i \in \mathcal{I}} P_{split}(\xi_i, \mathcal{T}) \prod_{\xi_j \in \mathcal{E}} (1 - P_{split}(\xi_j, \mathcal{T})),$$

where $\mathcal{I}$ and $\mathcal{E}$ denote the internal and terminal nodes, respectively. $P_{rule}(\rho|\eta, \mathcal{T})$ involves the splitting process, which initially chooses the splitting variable ω from a discrete uniform distribution. Then the split location $\boldsymbol{s}$ is chosen uniformly from a continuous subset of the locations in the ω^{th} variable. We call the pair of the values $\{\omega, s\}$ the *splitting rule*. The parameters a and b are determined from a priori knowledge or from a pilot study.

The GP parameters $(\boldsymbol{\beta}_{\eta_k}, \boldsymbol{\phi}_{\eta_k}, \sigma^2_{\eta_k})$ and $(\boldsymbol{\beta}_{\eta_k}, \boldsymbol{\phi}_{\delta_k}, \sigma^2_{\delta_k})$ are a priori independent between different partitions and independent of each other within the partitions of the input domain. As in Gramacy and Lee (2008), hyper-parameters $\boldsymbol{\phi}_{\eta_k}$ and $\boldsymbol{\phi}_{\delta_k}$ are considered to be a mixture of *Gamma* distributions. We choose non-informative priors for $(\boldsymbol{\beta}_{\eta_k}, \sigma_{\eta_k})$ and $(\boldsymbol{\beta}_{\delta_k}, \sigma^2_{\delta_k})$. More precisely, the prior distributions of the GP hyper-parameters in the k^{th} partition are:

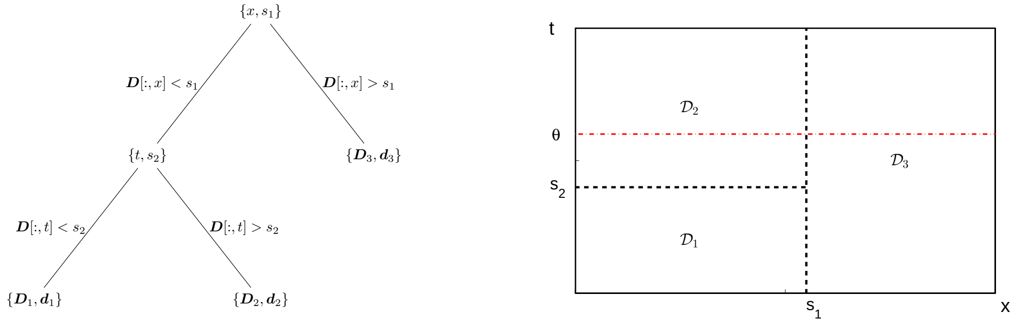
$$\pi(\boldsymbol{\beta}_{\eta_k}, \sigma^2_{\eta_k})\pi(\boldsymbol{\phi}_{\eta_k}) \propto \frac{1}{\sigma^2_{\eta_k}} \prod_{l=1:(q+p)} [G(\phi_{\eta_k,l}|\alpha_{G,1}, \beta_{G,1}) + G(\boldsymbol{\phi}_{\eta_k,l}|\alpha_{G,2}, \beta_{G,2})]/2$$

$$\pi(\boldsymbol{\beta}_{\delta_k}, \sigma^2_{\delta_k})\pi(\boldsymbol{\phi}_{\delta_k}) \propto \frac{1}{\sigma^2_{\delta_k}} \prod_{l=1:p} [G(\phi_{\delta_k,l}|\alpha_{G,1}, \beta_{G,1}) + G(\phi_{\delta_k,l}|\alpha_{G,2}, \beta_{G,2})]/2,$$

where $(q+p)$ is the dimension of the experimental input and calibration space, q is the dimension of the experimental input space, and $\alpha_{G,1}, \beta_{G,1}, \alpha_{G,2}, \beta_{G,2}$ express a prior knowledge. Moreover, we define the prior for $\boldsymbol{\theta}$ with a modified *Beta* distribution with parameters defined to represent previous studies or the domain scientist opinion. We also assign priors for the nugget hyper-parameter $\pi(\tau^2_{e_k})$ and $\pi(\tau^2_{v_k})$ as exponential distribution to ensure positive values.

Illustration: For illustration purposes, we assume a calibration problem with one experimental variable $(\boldsymbol{x})$ and one calibration variable $(\boldsymbol{t})$. Given a binary tree of depth two, three external nodes, and two splitting rules $\{\boldsymbol{x}, s_1\}$ and $\{\boldsymbol{t}, s_2\}$ we manage to partition the space into three different subregions as shown in Figure 1. For this particular tree and value of $\boldsymbol{\theta}$, subregions $\mathcal{D}_2$ and $\mathcal{D}_3$ contain experimental data, while subregion $\mathcal{D}_1$ does not contain experimental data. The data inside the subregions depend on the tree and the value of $\boldsymbol{\theta}$. Despite the fact that $\mathcal{D}_1$

does not have field (experimental) measurements, for the particular $\boldsymbol{\theta}$, the discrepancy function $\delta_2(\cdot)$ is updated from its prior.



$\{x, s_1\}$
$\boldsymbol{D}[:, x] < s_1$
$\boldsymbol{D}[:, x] > s_1$
$\{t, s_2\}$
$\{\boldsymbol{D}_3, \boldsymbol{d}_3\}$
$\boldsymbol{D}[:, t] < s_2$
$\boldsymbol{D}[:, t] > s_2$
$\{\boldsymbol{D}_1, \boldsymbol{d}_1\}$
$\{\boldsymbol{D}_2, \boldsymbol{d}_2\}$
t
θ
s_2
$\mathcal{D}_2$
$\mathcal{D}_3$
$\mathcal{D}_1$
s_1
x

(a) $\mathcal{T}$: Diagram

(b) $\mathcal{T}$: Graphically

Figure 1: Bayesian treed calibration in two forms: (a) $\mathcal{T}$: Diagram and (b) $\mathcal{T}$: Graphically

Posterior: The posterior distribution is known up to a normalizing constant as:

$$p(\boldsymbol{\Theta}, \boldsymbol{\theta}, \mathcal{T} | \boldsymbol{d}) \propto \pi(\mathcal{T})\pi(\boldsymbol{\theta}) \prod_{k=1:K} \pi(\boldsymbol{\phi}_{\eta_k}, \boldsymbol{\beta}_{\eta_k}, \sigma^2_{\eta_k}) |\boldsymbol{V}_{\boldsymbol{d}_k}|^{-1/2} \times \exp[-\frac{1}{2}(\boldsymbol{d}_k - E(\boldsymbol{d}_k))^T \boldsymbol{V}^{-1}_{\boldsymbol{d}_k}(\boldsymbol{d}_k - E(\boldsymbol{d}_k))]. \tag{2}$$

Posterior inference for the proposed model can be facilitated by MCMC methods.

4 Bayesian Inference and Computational Strategies

We follow a three-step scheme of the MCMC sampler as $(\boldsymbol{\theta} | \mathcal{T}, \boldsymbol{\Theta}, \boldsymbol{d})$, $(\boldsymbol{\Theta} | \boldsymbol{\theta}, \mathcal{T}, \boldsymbol{d})$, and $(\mathcal{T} | \boldsymbol{\Theta}, \boldsymbol{\theta}, \boldsymbol{d})$. Predictive distributions and uncertainty quantification are also derived through the MCMC algorithm. In this section, we present in details the steps of the MCMC algorithm and prediction analysis.

4.1 Posterior GP given tree $\mathcal{T}$ and calibration parameters $\boldsymbol{\theta}$

Conditional on the calibration parameters $\boldsymbol{\theta}$ and the tree $\mathcal{T}$, the full joint posterior distribution of the GP hyperparameters $(\boldsymbol{\Theta} | \boldsymbol{d}, \boldsymbol{\theta}, \mathcal{T})$ is analytically intractable. Exact posterior inference is performed by a customized MCMC algorithm. Analytically, for each external node $k = 1, \ldots, K$, we firstly sample from the closed posterior distribution of $\boldsymbol{\beta}_k | \boldsymbol{\theta}, \boldsymbol{\phi}_k, \boldsymbol{\tau}^2_k, \boldsymbol{d}_k$. Then, we sample from the posterior distribution of $\boldsymbol{\phi}_k, \boldsymbol{\sigma}^2_k, \boldsymbol{\tau}^2_k | \boldsymbol{d}_k, \boldsymbol{\theta}$, which we find by integrating out $\boldsymbol{\beta}_k$ with Metropolis-Hastings (M-H).

Given the prior specification for β_k and $\boldsymbol{\sigma}_k^2$ in Section 3, for each external node $k = 1, \ldots, K$, the posterior distribution of $\boldsymbol{\beta}_k$ given $\boldsymbol{\theta}, \boldsymbol{\phi}_k, \boldsymbol{\tau}_k, \boldsymbol{d}_k$ is a multivaraite Normal distribution with mean $\hat{\boldsymbol{\beta}}_k = \boldsymbol{W}_k \boldsymbol{H}_k^T \boldsymbol{V}_{\boldsymbol{d}_k}^{-1} \boldsymbol{d}_k$ and variance $\boldsymbol{W}_k = \boldsymbol{H}_k^T \boldsymbol{V}_{\boldsymbol{d}_k}^{-1} \boldsymbol{H}_k$ is:

$$\boldsymbol{\beta}_k | \boldsymbol{\theta}, \boldsymbol{\phi}_k, \boldsymbol{\tau}_k, \boldsymbol{d}_k \sim \mathcal{N}(\hat{\boldsymbol{\beta}}_k, \boldsymbol{W}_k). \tag{3}$$

Both $\hat{\boldsymbol{\beta}}_k$ and $\boldsymbol{W}_k$ depend on $(\boldsymbol{\phi}_k, \boldsymbol{\tau}_k^2)$ and $\boldsymbol{\theta}$. In Eq. 2 we integrate out $\boldsymbol{\beta}_k$ and compute the joint posterior distribution of $\boldsymbol{\phi}_k, \boldsymbol{\sigma}_k^2, \boldsymbol{\tau}_k^2 | \boldsymbol{d}, \boldsymbol{\theta}$ as:

$$p(\boldsymbol{\phi}_k, \boldsymbol{\sigma}_k^2, \boldsymbol{\tau}_k^2 | \boldsymbol{d}_k, \boldsymbol{\theta}) \propto \pi(\boldsymbol{\phi}_k)\pi(\boldsymbol{\sigma}_k^2)\pi(\boldsymbol{\tau}_k^2)|\boldsymbol{V}_{\boldsymbol{d}_k}|^{-1/2}|\boldsymbol{W}_k|^{1/2}\exp[-\frac{1}{2}(\boldsymbol{d}_k - \boldsymbol{H}_k\hat{\boldsymbol{\beta}}_k)^T \boldsymbol{V}_{\boldsymbol{d}_k}^{-1}(\boldsymbol{d}_k - \boldsymbol{H}_k\hat{\boldsymbol{\beta}}_k)]. \tag{4}$$

The conditional posteriors of $\boldsymbol{\phi}_k | \boldsymbol{\sigma}_k^2, \boldsymbol{\tau}_k^2$, $\boldsymbol{\sigma}_k^2 | \boldsymbol{\phi}_k, \boldsymbol{\tau}_k$ and $\boldsymbol{\tau}_k^2 | \boldsymbol{\sigma}_k^2, \boldsymbol{\phi}_k$ cannot be sampled directly. Therefore, we use Metropolis-Hastings updates within a Gibbs sampler, (Mueller, 1993; Gelfand and Smith, 1990; Hastings, 1970). For more details, refer to Appendix B.

Remark: Given the values of $\boldsymbol{\theta}$, some of the partitions may not have experimental observations. In this case, the likelihood of the output is a function of $\eta(\,\cdot\,,\,\cdot\,)$ and experimental nugget τ_v^2. However, the parameters of the discrepancy (except $\boldsymbol{\beta}_{\delta_k}$) and the observational error are updated and determined from their prior distribution.

4.2 Posterior inference calibration parameters given the tree $\mathcal{T}$ and GP hyperparameters

Conditional on the tree $\mathcal{T}$, and GPs parameters $\boldsymbol{\phi}, \boldsymbol{\sigma}^2$ and $\boldsymbol{\tau}$, the posterior distribution of the calibration parameter $\boldsymbol{\theta}$ is:

$$p(\boldsymbol{\theta} | \boldsymbol{d}, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2) \propto p(\boldsymbol{\theta}) \prod_{k=1}^{K} |\boldsymbol{V}_{\boldsymbol{d}_k}|^{-1/2}|\boldsymbol{W}_k|^{1/2}\exp[-\frac{1}{2}(\boldsymbol{d}_k - \boldsymbol{H}_k\hat{\boldsymbol{\beta}}_k)^T \boldsymbol{V}_{\boldsymbol{d}_k}^{-1}(\boldsymbol{d}_k - \boldsymbol{H}_k\hat{\boldsymbol{\beta}}_k)], \tag{5}$$

which is analytically intractable. At least for one of the external nodes k', $\boldsymbol{H}_{k'}, \hat{\boldsymbol{\beta}}_{k'}, \boldsymbol{V}_{\boldsymbol{d}_{k'}}$, and $\boldsymbol{W}_{k'}$ are functions of $\boldsymbol{\theta}$. Posterior inference is performed by a Metropolis-Hastings algorithm (we propose a values $\boldsymbol{\theta}$ and decide with an M-H step whether or not we accept or reject). By changing $\boldsymbol{\theta}$, we change part of the calibration input which is associated with $\boldsymbol{D}_k^1(\boldsymbol{\theta})$. This may change the number of observations inside each external node of the binary tree. When we propose $\boldsymbol{\theta}$ in a different terminal node from the current one we have to update also the GP hyperparameters of the discrepancy $\delta(\,\cdot\,)$. The parameters associated with the linear term, $\boldsymbol{\beta}$, are updated by sampling directly form posterior distribution, while the other parameters are updated from the prior distribution.

4.3 Bayesian tree given GP and the calibration parameter $\boldsymbol{\theta}$

The structure of the binary tree given the calibration parameter $\boldsymbol{\theta}$ and the GP parameters $\boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{\tau}$ is updated through a random scan MCMC sweep that includes as updates the *Grow*, *Prune*, *Change*, and *Swap* operations introduced by Chipman et al. (1998) and *Rotate* operator introduced by Gramacy and Lee (2008). The first three operations are Metropolis-Hastings updates operating on fixed parametric dimensional spaces while the last two are a reversible jump (RJ) pair of moves (Green, 1995) that perform changes to the dimension of the parameter space. The calibration parameters $\boldsymbol{\theta}$ do not change dimension in the RJ type of moves in the Bayesian tree. As such, we can update them given the Bayesian tree and ignore its update in the tree operations.

Because we can integrate out the linear model parameter $\boldsymbol{\beta}$, we do not need to generate proposed values for these parameters in the RJ pair of moves. In the grow and prune operation, we can propose in the kth partition $(\boldsymbol{\phi}_k, \boldsymbol{\sigma}_k^2) \equiv (\boldsymbol{\phi}_{\eta_k}, \sigma_{\eta_k}^2, \boldsymbol{\phi}_{\delta_k}, \sigma_{\delta_k}^2)$ from their prior specifications. However, to increase the acceptance ratio of the RJ type of moves, we build proposal distributions that can change in the MCMC moves. The proposal distribution used is of the form:

$$q(\boldsymbol{\phi}_k, \boldsymbol{\sigma}_k^2) = q(\boldsymbol{\phi}_{\eta_k}, \sigma_{\eta_k}^2, \boldsymbol{\phi}_{\delta_k}, \sigma_{\delta_k}^2) = \pi(\boldsymbol{\phi}_{\eta_k})\pi(\boldsymbol{\phi}_{\delta_k})p(\sigma_{\eta_k}^2|\boldsymbol{\phi}_{\eta_k}, \boldsymbol{y}_k)p(\sigma_{\delta_k}^2|\boldsymbol{\phi}_{\eta_k}, \boldsymbol{\phi}_{\delta_k}, \sigma_{\eta_k}^2, \boldsymbol{z}_k),$$

where $\pi(\boldsymbol{\phi}_{\eta_k})$ and $\pi(\boldsymbol{\phi}_{\delta_k})$ are the prior specifications for $\boldsymbol{\phi}_{\eta_k}$ and $\boldsymbol{\phi}_{\delta_k}$, respectively. The proposal distribution of $\sigma_{\eta_k}^2$ is equal to the conditional distribution $p(\sigma_{\eta_k}^2|\boldsymbol{\phi}_{\eta,k}, \boldsymbol{y}_k)$, ignoring the information gained from experimental data. This conditional distribution has an *inverse Gamma* distribution:

$$\sigma_{\eta_k}^2|\boldsymbol{y}_k, \boldsymbol{\phi}_{\eta_k} \sim IG((r+n_k), (n_k - m)\hat{\sigma}_{\eta_k} + \sigma_{\eta_k}^0),$$

where $\hat{\sigma}_{\eta_k}^2$ is the generalized least squares (GLS) estimator of $\sigma_{\eta_k}^2$. Similarly, we ignore the simulated data and compute the conditional posterior distribution of $\sigma_{\delta_k}^2|\boldsymbol{\phi}_{\eta_k}, \boldsymbol{\phi}_{\delta_k}, \sigma_{\eta_k}^2, \boldsymbol{z}_k$ to build the proposal distribution for σ_{δ_k}. To ensure reversibility of the Markov chain, we follow the same strategy for the grow and the prune operation.

Given the current state is at binary tree $\mathcal{T}$, the *Grow* operation involves several steps. We randomly select an external node ξ_{j0} that corresponds to a subregion $\mathcal{D}_{j0}$ with data $\{\boldsymbol{D}_{j0}, \boldsymbol{d}_{j0}\}$ and GP calibration model with parameters $(\boldsymbol{\phi}_{j0}, \boldsymbol{\sigma}_{j0}^2)$. We propose node ξ_{j0} to split into two new child nodes ξ_{j_1} and ξ_{j_2} according to the splitting rule P_{rule} used in the priors, and we denote the proposed tree as $\mathcal{T}'$. We consider that nodes ξ_{j_1} and ξ_{j_2} correspond to disjoint subregions $\mathcal{D}_{j_1}$ and $\mathcal{D}_{j_2}$, the union of which is $\mathcal{D}_{j_0}$, with data $\{\boldsymbol{D}_{j_1}, \boldsymbol{d}_{j_1}\}$ and $\{\boldsymbol{D}_{j_2}, \boldsymbol{d}_{j_2}\}$, respectively. Let $(\boldsymbol{\phi}_{j_1}, \boldsymbol{\sigma}^2_{j_1})$ and $(\boldsymbol{\phi}_{j_2}, \boldsymbol{\sigma}^2_{j_2})$ denote the vector parameters of the GP calibration associated with the new nodes ξ_{j_1} and ξ_{j_2}. A newly formed child, e.g., ξ_{j_1}, is randomly chosen to receive values for $(\boldsymbol{\phi}_{j_1}, \boldsymbol{\sigma}^2_{j_1})$ from the parent such that $(\boldsymbol{\phi}_{j_1}, \boldsymbol{\sigma}^2_{j_1}) = (\boldsymbol{\phi}_{j_0}, \boldsymbol{\sigma}^2_{j_0})$. Meanwhile, for the other, $(\boldsymbol{\phi}_{j_2}, \boldsymbol{\sigma}^2_{j_2})$, we generate values from a proposal $q(\boldsymbol{\phi}_{j_2}, \boldsymbol{\sigma}^2_{j_2})$ per our earlier explanation.

The *Grow* operation is accepted with probability $\min\{1, A\}$, where

$$A = \frac{1 - a(1 + u_{\xi_{j_0}})^{-b}}{a(1 + u_{\xi_{j_0}})^{-b}(1 - a(2 + u_{\xi_{j_0}})^{-b})^2} \frac{|G|}{|P'|} \frac{p(\boldsymbol{\phi}_{j_1}, \boldsymbol{\sigma}^2_{j_1} | \boldsymbol{d}_{j_1}, \boldsymbol{\theta}) p(\boldsymbol{\phi}_{j_2}, \boldsymbol{\sigma}^2_{j_2} | \boldsymbol{d}_{j_2}, \boldsymbol{\theta})}{p(\boldsymbol{\phi}_{j_0}, \boldsymbol{\sigma}^2_{j_0} | \boldsymbol{d}_{j_0}, \boldsymbol{\theta}) q(\boldsymbol{\phi}_{j_2}, \boldsymbol{\sigma}^2_{j_2})}, \tag{6}$$

where G the set of growable node in the current tree $\mathcal{T}$ and P' the set of prounable nodes in the proposed tree $\mathcal{T}'$. The *Prune* operation is the reverse analog *of Grow*, from tree $\mathcal{T}'$ to $\mathcal{T}$, and designed so the detailed balance condition is satisfied. The operation is accepted with probability $\min\{1, 1/A\}$.

In the change operation, the parameters of the discrepancy function $\delta(\,\cdot\,)$, which are associated with observations, may also change. The mean is directly updated from the closed form given in Section 4.1, while the correlation parameters are updated from the prior specification and the variance from the conditional representation.

4.4 Prediction and uncertainty quantification

The predictive distribution $\zeta(\boldsymbol{x})$ in partition k for a given tree $\mathcal{T}$, hyperparamters $\boldsymbol{\Theta}$, and the calibration parameter $\boldsymbol{\theta}$ is a Gaussian process with mean:

$$E(\zeta(\boldsymbol{x}) | \boldsymbol{\theta}, \mathcal{T}, \boldsymbol{\Theta}, \boldsymbol{d}) = \boldsymbol{h}_k(\boldsymbol{x}, \boldsymbol{\theta})^T \hat{\boldsymbol{\beta}}_k + \boldsymbol{t}(\boldsymbol{x}, \boldsymbol{\theta})^T \boldsymbol{V}^{-1}_{\boldsymbol{d}_k} \{\boldsymbol{d}_k - \boldsymbol{H}_k \hat{\boldsymbol{\beta}}_k\},$$

where $\boldsymbol{h}_k(\boldsymbol{x}, \boldsymbol{\theta})$=$(\boldsymbol{h}_{\eta_k}(\boldsymbol{x}, \boldsymbol{\theta}),\ \boldsymbol{h}_{\delta_k}(\boldsymbol{x}))^T$, $\boldsymbol{t}_k(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{C}_{\eta_k}((\boldsymbol{x}, \boldsymbol{\theta}), D_k) + (C_{\delta_k}(\boldsymbol{x}, D_{k,1})\ , \boldsymbol{0})^T$,

and covariance function:

$$\begin{aligned} \operatorname{cov}(\zeta(\boldsymbol{x}), \zeta(\boldsymbol{x}') | \boldsymbol{\theta}, \mathcal{T}, \boldsymbol{\Theta}, \boldsymbol{d}) = & c_{\eta_k}((\boldsymbol{x}, \boldsymbol{\theta}), (\boldsymbol{x}', \boldsymbol{\theta})) + c_{\delta_k}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}_k(\boldsymbol{x}, \boldsymbol{\theta})^T \boldsymbol{V}^{-1}_{\boldsymbol{d}_k} \boldsymbol{t}_k(\boldsymbol{x}', \boldsymbol{\theta}) \\ & + (\boldsymbol{h}_k(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{H}_k^T \boldsymbol{V}^{-1}_{\boldsymbol{d}_k} \boldsymbol{t}_k(\boldsymbol{x}, \boldsymbol{\theta}))^T \boldsymbol{W}_k (\boldsymbol{h}_k(\boldsymbol{x}', \boldsymbol{\theta}) - \boldsymbol{H}_k^T \boldsymbol{V}^{-1}_{\boldsymbol{d}_k} \boldsymbol{t}_k(\boldsymbol{x}', \boldsymbol{\theta})) \end{aligned} \tag{7}$$

for $\boldsymbol{x}' \in \mathcal{D}_k$. When $\boldsymbol{x} \notin \mathcal{D}_k$, the covariance is equal to zero due to the assumption of independent external nodes associated with the Bayesian tree.

The Bayesian predictive density function is calculated through Bayesian model averaging

(BMA) as:

$$p(\zeta(\boldsymbol{x})|\boldsymbol{d}) = \sum_{\mathcal{T}} \int_{\boldsymbol{\theta},\boldsymbol{\Theta}} p(\zeta(\boldsymbol{x})|\boldsymbol{\theta},\mathcal{T},\boldsymbol{\Theta},\boldsymbol{d})\pi(\boldsymbol{\theta},\mathcal{T},\boldsymbol{\Theta}|\boldsymbol{d})d\boldsymbol{\theta}d\boldsymbol{\Theta}. \tag{8}$$

We approximate the preceding equation with Monte Carlo samples as:

1. Generate MCMC samples $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\Theta}^{(1)}, \mathcal{T}^{(1)}), \ldots, (\boldsymbol{\theta}^{(M)}, \boldsymbol{\Theta}^{(M)}, \mathcal{T}^{(M)})$ from $p(\boldsymbol{\theta}, \boldsymbol{\Theta}, \mathcal{T}|\boldsymbol{d})$ as described in Section 4.

2. Approximate $p(\zeta(\boldsymbol{x})|\boldsymbol{d})$ by: $\hat{p}(\zeta(\boldsymbol{x})|\boldsymbol{d}) = \frac{1}{M}\sum_{k=1}^{M} p(\zeta(\boldsymbol{x})|\boldsymbol{\theta}^{(k)}, \boldsymbol{\Theta}^{(k)}, \mathcal{T}^{(k)}, \boldsymbol{d})$.

When we do not have discontinuity in the mean, this predictive process tends to smoothen the prediction surface around the tree limit subregion edges (refer to Gramacy and Lee (2008)).

Using covariance functions similar to those in Paciorek and Schervish (2006) and Konomi et al. (2014b), we can knit together multiple different subregions so the predictions depend from all the outputs in every MCMC iteration. Usually, the global covariance leads to better predictions as shown in Konomi et al. (2014b). Given the partition and the calibration parameters, we can construct global parametric non-stationary covariance functions for the calibration formulation.

Specifically, if an isotropic correlation function, $\rho_0(\,\cdot\,)$, is positive definite on $\mathbb{R}^{(q+p)}$, then a valid non-stationary covariance function on $\mathbb{R}^{(q+p)}$ is defined by:

$$c_{NS}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma(\boldsymbol{x}_i)\sigma(\boldsymbol{x}_j)|\boldsymbol{B}(\boldsymbol{x}_i)|^{\frac{1}{4}}|\boldsymbol{B}(\boldsymbol{x}_j)|^{\frac{1}{4}}\left|\frac{\boldsymbol{B}(\boldsymbol{x}_i)+\boldsymbol{B}(\boldsymbol{x}_j)}{2}\right|^{\frac{1}{2}}\rho_0(\sqrt{Q(\boldsymbol{x}_i, \boldsymbol{x}_j)}), \tag{9}$$

where $Q(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)'\big((\boldsymbol{B}(\boldsymbol{x}_i) + \boldsymbol{B}(\boldsymbol{x}_j))/2\big)^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)$, a weighted Mahalanobis distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $\sigma(\boldsymbol{x}_i)$ is the standard deviation at $\boldsymbol{x}_i$, and $\boldsymbol{B}(\boldsymbol{x}_i)$ is referred to as the $(q+p) \times (q+p)$ kernel covariance matrix at location $\boldsymbol{x}_i$. The separable square exponential correlation function, used in this paper, is a special case of the described covariance function where $\boldsymbol{B}_i$ is diagonal. The square roots of the eigenvalues of $\boldsymbol{B}_i$ control the range of the spatial dependence.

The proposed Bayesian tree models the kernel covariance matrices $\boldsymbol{B}(\boldsymbol{x}_i)$ and the standard deviations $\sigma(\boldsymbol{x}_i)$ as a step function. If $\xi(\boldsymbol{x}_i) \in \{1, \cdots, K\}$, denote the region that $\boldsymbol{x}_i$ belongs to. Each region has its corresponding kernel matrix and standard deviation, that is, $\boldsymbol{B}_i = \boldsymbol{B}_{\xi(\boldsymbol{x}_i)} \in \{\boldsymbol{B}_k, \xi = 1, \cdots, K\}$ and $\sigma_i = \sigma_{\xi(\boldsymbol{s}_i)} \in \{\sigma_\xi, \xi = 1, \cdots, K\}$. Both $\boldsymbol{V}_\eta(D)$ and $\boldsymbol{V}_\delta(D_1)$ can be modeled separately with the above covariance function, leading to a non-stationary covariance function $\boldsymbol{V}_{\boldsymbol{d}}$ for the calibration model.

When one or more variables $\boldsymbol{X}$ are subject to parametric variability, the task of the uncertainty quantification is to make inference about the distribution of $\zeta(\boldsymbol{X})$. We use the close form of the distribution $\zeta(\boldsymbol{x})$ for particular values of $\boldsymbol{x}$ to help us compute the distribution properties of $\zeta(\boldsymbol{X})$, e.g., the first moment of $\zeta()$ is $E_{\boldsymbol{X}}(\zeta(\boldsymbol{X})) = \int_{\mathcal{X}} \zeta(x) dG_{\boldsymbol{x}}$, where $G_{\boldsymbol{x}}$ is the distribution of $\boldsymbol{X}$.

5 Case Studies

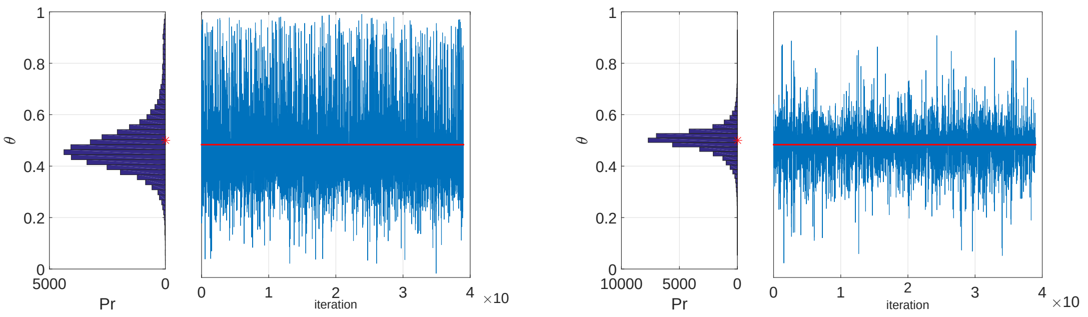
To better evaluate the proposed BTC method, we first consider a case scenario where $\eta(\,\cdot\,,\,\cdot\,)$ and $\delta(\,\cdot\,)$ are known functions. We assume to know exactly the calibration values $\boldsymbol{\theta}$. Despite explicitly knowing the functions of interest, we assume the simulator is computationally demanding and must be evaluated at some specific design points prior to the calibration procedure. We select the design points $(\boldsymbol{x}_1^*, \boldsymbol{t}_1), \ldots, (\boldsymbol{x}_m^*, t_m)$ via a space filling design, such as Latin hypercube sampling (LHS) (McKay et al., 1979). We follow the same sampling strategy for the experimental observable input points $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. The calibration parameters can usually be confounded with the discrepancy in cases where different combinations of $\boldsymbol{\theta}$ and δ give the same posterior distribution for the prediction. In our case studies we have observed that selecting the right prior distributions for the calibration parameters and using constant discrepancy functions can usually minimize the confounding problem. For the purpose of this paper, we will not explore this issue further.

First case study: In our first example, we assume a function of $\eta(\,\cdot\,)$ in a three-dimensional input space (one observable input x_1 and two calibration inputs t_1, t_2):

$$\eta(x_1, t_1, t_2) = \begin{cases} (1 - x_1)\cos(\pi t_1) + 0t_2, \;\; 0 \leq x_1 < 0.4 \\ (1 - x_1)\cos(\pi t_1) + 0t_2 + 0.5, \;\; 0.4 \leq x_1 < 1, \end{cases}$$

where the calibration variable t_2 does not affect the output of the $\eta(\,\cdot\,,\,\cdot\,,\,\cdot\,)$. From this formulation, it is clear that the computer model has a discontinuity in the observable input x_1. To make the case more realistic, we assume the computer model output has a normally distributed nugget effect with zero mean and variance $\tau_v^2 = 0.1$. We also assume the discrepancy function is $\delta(x_1) = 0.1$, and the real experimental output is $\zeta(x_1) = \eta(x_1, 0.5, 0.5) + \delta(x_1) + e$ for calibration parameters $\boldsymbol{\theta} = (0.5, 0.5)$. The nugget effect e is considered independent and normally distributed with mean zero and variance $\tau_e^2 = 0.1$. We evaluate $n = 13$ times $\zeta(x_1)$, using the LHS design, and assume these to be the “real” experimental measurements.

For both the calibration parameters θ_1 and θ_2, we assume a $Beta(1, 1)$ prior distribution. The parameters in the prior distribution of the Bayesian tree are set at $a = 0.6$ and $b = 2$. Since the variation of the functions in the two subregions is similar the prior distribution



(a) Calibration of θ_1 using GP calibration

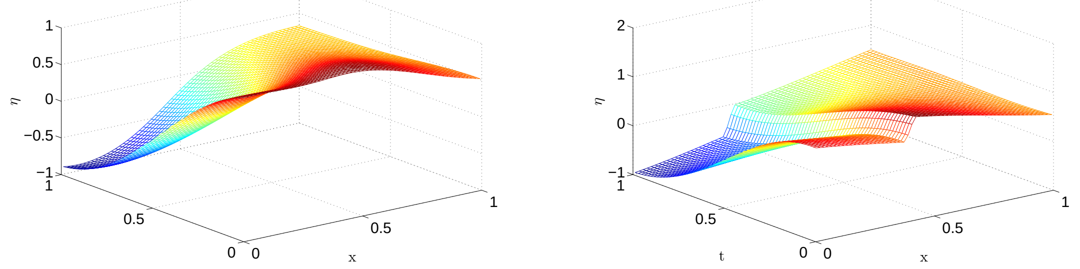
(b) Calibration of θ_1 using TGP calibration

Figure 2: MCMC for the calibration parameter θ_1 for two different calibration methods (a) GP calibration and (b) proposed TGP calibration.

for the correlation parameters can be expressed with only a simple Gamma distribution or a mixture of Gamma distributions which have similar parameter values. This is way, the priors for the correlation parameters are chosen as a mixture of two *Gamma* distributions, $\pi(\lambda) = 0.5G(\lambda|5,3) + 0.5G(\lambda|5,2)$. The prior for σ_δ^2 is exponential with parameter 0.005 and for τ_v^2, τ_e^2 is exponential with parameter 0.01.

For model calibration, we use two methods:(a) the SBGPC and (b) our proposed BTC. We run the two MCMC algorithms for $50,000$ iterations, the first $10,000$ of which are taken as burn-in. Convergence is diagnosed based on examination of trace plots in a pilot study. The posterior distribution of θ_1 using the SBGPC is shown in Figure 2(a), and the posterior distribution of θ_1 using the proposed BTC is shown in Figure 2(b). The results favor the BTC. The SBGPC gives an obvious bias and larger variance for the posterior distribution of θ_1. The mean squared error (MSE) for the proposed calibration method is 0.0031, while the MSE when using the SBGPC is 0.0264. Both methods give similar posterior distribution of θ_2 and are basically the same with the prior distribution.

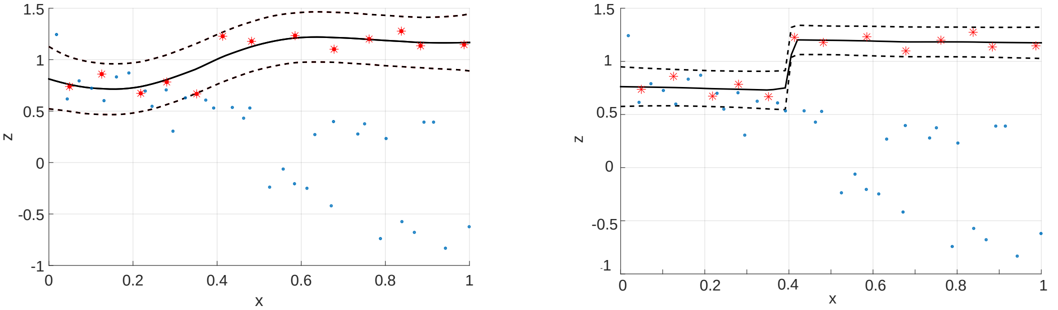
We compare the SBGPC and proposed BTC's prediction abilities. The prediction of the real system mean and 95% credible interval using SBGPC is shown with black lines in Figure 4(a), and using the proposed BTC is shown in Figure 4(b). The proposed BTC predictions are more accurate with less variance than the SBGPC predictions. Substantial disagreement between the real system and SBGPC prediction means are observed close to the discontinuity point. The mean squared prediction error (MSPE) for 200 equally spaced observable input values is $MSPE_{BTC} = 0.021$ when using BTC and $MSPE_{SBGPC} = 0.086$ when using SBGPC. The $MSPE_{BTC}$ is approximately four times smaller than the $MSPE_{SBGPC}$. In addition, prediction mean variance using SBGPC is larger for the observable input. Gaussian processes are unable to model the discontinuity in the response surface of the computer model η. Figure 4(a) shows



(a) Prediction of the η using GP calibration (b) Prediction of η using TGP calibration

Figure 3: MCMC for the calibration parameter θ_1 for two different calibration methods (a) GP calibration and (b) proposed TGP calibration.

the predicted mean for the response surface of the computer model $\eta(x, t_1)$ using SBGPC. Figure 4(b) shows the predicted mean for the response surface of the computer model $\eta(x, t_1)$ using the proposed BTC. The predicted mean response surface using the BTC model has a discontinuity at 0.4 similar to the computer model $\eta(x, t_1)$.



(a) Prediction of the real system using GP calibration (b) Prediction of the real system using TGP calibration

Figure 4: Prediction mean and 95% prediction intervals using two different calibration methods (a) GP calibration and (b) proposed TGP calibration.

Second case study: In our second case study, we consider a six-dimensional input (one observable input x_1 and five calibration inputs $(t_1, \ldots, t_5)$) example with a real computer model function:

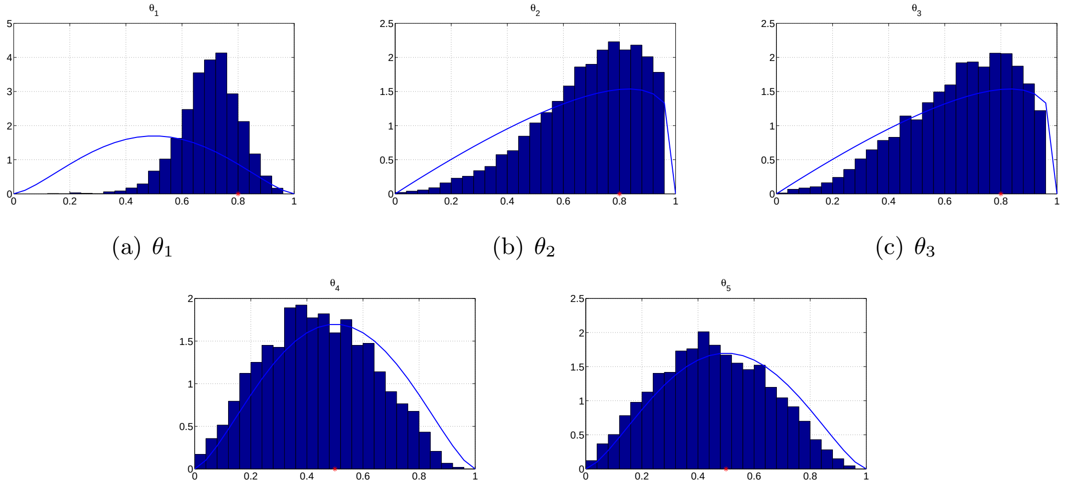
$$\eta(x_1, t_1, t_2, t_3, t_4, t_5) = (t_1 + 0.5) \exp\{\sin((0.9(x_1 + 0.48)^{10}))\} + t_2 t_3,$$

on the hypercube $[x_1, t_1, \ldots, t_5] = [0, 1]^6$. This function varies wildly as a function of x_1 and t_1, and it is quadratic with respect to t_2 and t_3, and constant with respect to t_4 and t_5. Localized features with respect to x_1 are observed on η. To generate a link between this computer model and "real" system, we assume the discrepancy function is constant $\delta(x_1) = 2$. We evaluate $\eta(x_1, t_1, \ldots, t_5)$ at $m = 120$ LHS design points and add a nugget effect with variance $\tau_v^2 = 0.1$ to represent the output of the computer model. Let $\boldsymbol{\theta} = (0.8, 0.8, 0.8, 0.5, 0.5)$ is the calibration parameter vector. We evaluate 40 times $\zeta(x_1) = \eta(x_1, 0.8, 0.8, 0.8, 0.5, 0.5) + \delta(x_1) + e$, which acts as the "real" experimental data. The nugget e is considered independent with variance, depending on the location x_1: $\tau_e^2 = 1$ for $x_1 < 0.4$ and $\tau_e^2 = 0.2$ for $x_1 > 0.4$.

For the Bayesian inference, the prior distribution for calibration parameters $(\theta_1, \theta_4, \theta_5)$ are assumed $Beta(2.5, 2.5)$. We also assume $Beta(2.5, 1.25)$ prior distributions for the other two parameters (θ_2, θ_3). The prior distributions of the Bayesian tree and covariance functions are chosen similar to the first case study. The parameters in the prior distribution of the Bayesian tree are set at $a = 0.6$ and $b = 2$. Since the variation of the functions in the two subregions are different, it is preferable the prior distribution for the correlation parameters to be mixture of Gamma distributions with two distinct modes. This is why the priors for the correlation parameters are chosen two be $\pi(\lambda) = 0.5G(\lambda|2,1) + 0.5G(\lambda|20,2)$. The prior for σ_δ^2 is exponential with parameter 0.4 and for τ_v^2, τ_e^2 is exponential with parameter 0.3.

We apply the BTC MCMC algorithm, described in Section 4, to this artificial calibration problem. We run $25,000$ MCMC iterations, the first $5,000$ of which are taken as burn-in. Figure 5 shows the marginal posterior distribution of the calibration parameters $(\theta_1, \ldots, \theta_5)$ with the prior densities (blue curves) and real values (red stars). Although it has a slight preference toward the prior distribution centered at 0.5, the posterior distribution of θ_1 is closer to the real value. From the Bayesian point of view, this is expected behavior. The posterior distributions of θ_2 deviate a bit from the prior distribution toward the real value. This possibly indicates that the first calibration parameter θ_1 is more important than θ_2 in this model calibration problem. The posterior distribution of θ_3 prefers posterior values close to the prior distribution, even though the real value of θ_3 is 0.8 and the prior distribution is left skewed. The same happens with SBGPC as well. We believe this has to do mostly with the restricted information (sample size) in this problem. As the sample size of the computer model increases, we have observed considerable improvement on the posterior distributions of the calibration parameters. Finally, the posterior distributions of θ_4 and θ_5 are close to their prior distributions, which is expected because the real model calibration does not depend on these parameters.

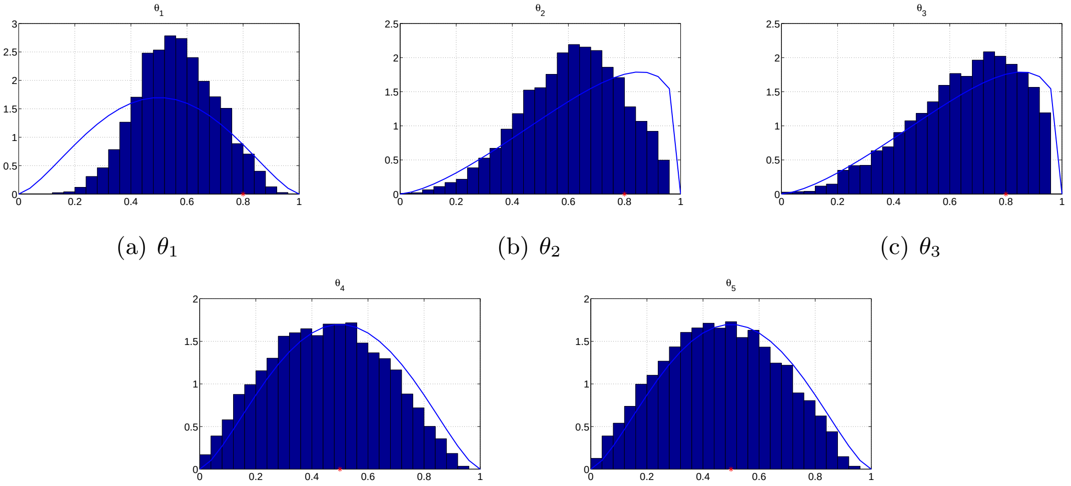
To demonstrate the proposed BTC method's performance in correctly estimating the calibration parameters, we compare its posterior calibration parameter distributions with those obtained using SBGPC (Figure 6). Compared to our method, we observe the bias and variance of the posterior distribution of θ_1 using SBGPC have increased. The same is observed for the posterior distribution of θ_2, where the bias is toward the central values. For θ_3, the posterior distribution using SBGPC method is similar to the one obtained using BTC. The posterior distributions of θ_4 and θ_5 are similar to the prior distribution and the posterior distributions using the proposed BTC method. Our experience with the problem is that significant differences between the two methods are observed in only a few parameters. The two methods usually agree in most posterior distributions of the calibration parameters.



(a) θ_1 (b) θ_2 (c) θ_3

(d) θ_4 (e) θ_5

Figure 5: Marginal posterior distributions for the model parameters from the MCMC sample (histograms after 10,000 iterations) along with the prior density (blue curves) using the proposed BTC.

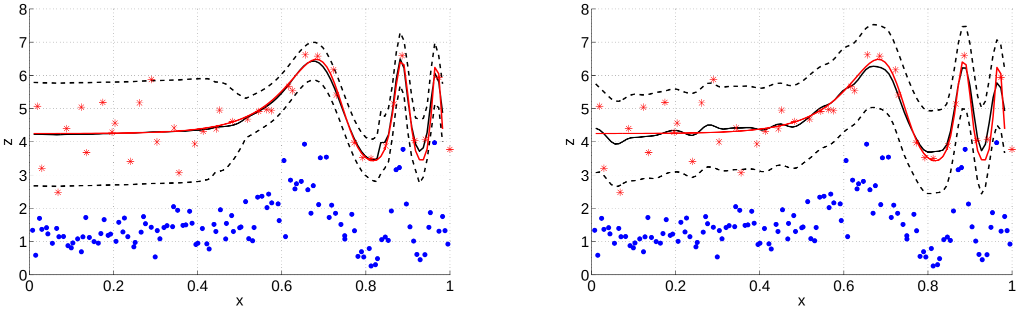


(a) θ_1 (b) θ_2 (c) θ_3

(d) θ_4 (e) θ_5

Figure 6: Marginal posterior distributions for the model parameters from the MCMC sample (histograms after 10,000 iterations) along with the prior density (blue curves) using SBGPC.

A better comparison between the two methods is the predictive distribution for the real



(a) Prediction of $\zeta(\cdot)$ using TGP calibration

(b) Prediction of the $\zeta(\cdot)$ using GP calibration

Figure 7: Prediction mean and 95% prediction intervals using: (a) the proposed BTC and (b) the SBGPC.

system as a function of the observable input. The MSPE for 200 equally spaced observable input values is $MSPE_{BTC} = 0.0139$ when using BTC and $MSPE_{SBGPC} = 0.0385$ using SBGPC. To better visualize the prediction performance, Figure 7(a) shows the prediction mean with a 95% prediction interval using BTC, while Figure 7(b) illustrates the prediction mean with a 95% prediction interval using SBGPC. In Figure 7, the red stars (*) represent real data, the blue dots (.) represent computer experiment output, the red line is the mean of the real output, the black solid line shows the predicted mean, and the dashed lines represent the 95% prediction interval. From these two graphs, it is obvious that the proposed TGP calibration gives better prediction for the mean. The black solid line in Figure 7(a) is closer to the red solid line than the black solid line in Figure 7(b). Moreover, the 95% prediction interval are larger in the subregion with large variance and small in the subregion with smaller variance. Conversely, SBGPC tends to underestimate the predicted variance in the first half and overestimate it in the second half of the input parameter. The prediction variance using BTC better represents the field output for all values of the observable input x. The advantage of our approach is that the mean, variation, and variance for both the computer model and the discrepancy term can change as a function of the observable and/or calibration inputs. The variance of the two nugget terms can also change. This flexibility results in better predictions for the real system and calibration posterior distributions closer to the real values.

6 Application: AX Cold Flow

The conceptual carbon capture system developed by Carbon capture sequestration initiative (CCSI) group is a post-combustion solid-sorbent system composed of two main components: the adsorber and the regenerator (Figure 8). These modeling tools involve multiphysics simulations, which include hydrodynamics, heat transfer, and chemical reactions in the system. To achieve the ultimate modeling and simulation goal of quantifying predictive confidence in large commercial devices, a hierarchical validation methodology has been developed and implemented, from basic unit problems and upscaling with filtering models, to Carbon Capture Unit (C2U) batch and eventually to large scale systems.

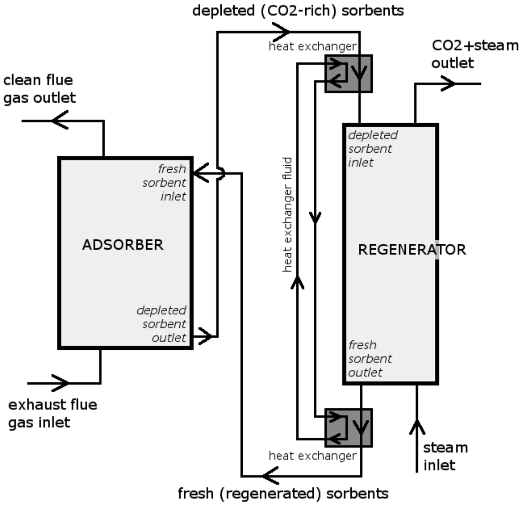


Figure 8: Schematic of a carbon capture unit with adsorber and regenerator.

In this paper we focus on the cold and non-reacting flow involving an amine based sorbent in the adsorber. The properties of this chosen sorbent with coded name AX including its CO_2 adsorption characteristics can be found in Krutka and Sjostrom (2013). The AX sorbent, comprised of small chemically reactive particles flowing through the device, is capable of reacting with the carbon dioxide and removing it from the thermal power plant exhaust.

6.1 Experiment setup and computer model

The C2U unit is a cylinder with 1.003 m in height and 0.0685 m in radius. A cooling coil stacks up in the lower 1/3 of the C2U unit and forms the inner coil and outer coil. Cooling or heating oil is circulating in the coil to achieve a desired bed temperature. A specific amount of sorbent, either 3.1 kg of AX particles or 1.62 kg of 32D powder is placed inside the C2U unit. The gas with specific flow rate, composition, and temperature is blown from the bottom of the unit by a flow transmitter controller (FTC). The gas exits at the top via a 1-inch diameter tube.

In the cold flow experiments with sorbent AX, the inlet gas consists of mainly of nitrogen N2 with limited H2O. Without CO2 in the gas inlet, there is no reaction. Thus, no heat is generating and the coil is not filled with heating or cooling oil. The main quantity of interest in this set of experiments has been the pressure drop, noted as PDT3820. Forty experiments with various flow rates have been performed, and PDT3820 has been recorded in a one-minute interval, as the main quantity of interest (QOI). For all cold flow experiments, gas flow rate through the plenum were controlled at different flow rates, ranging from 15 to 60 slpm.

The efficiency of $CO2$ adsorption by sorbent particles is partially determined by the hydrodynamics of the multi-phase flow. If the bed is not fully fluidized, the solid particles stay packed or semi-packed at the bottom, and as a result, there is not enough mixture between the solid particles and air. This leads to inadequacy in both spatial and temporal space for reaction, as well as a very low CO2 adsorption efficiency. If the fluidized bed height is too large, some solid particles will escape, and the inventory loss reduces the adsorption capacity. The bed height of a fluidized flow is characterized by the distribution of void fraction, and this hard-to-measure quantity is closely related to the easy-to-measure quantity pressure distribution. $PDT3820$ is for pressure drop at the lower bed.

An open source software, named Multiphase Flow with Interphase eXchanges (MFIX), has been used to conduct computational fluid dynamics (CFD) simulations. The effectiveness of CO_2 adsorption in a C2U unit depends on how well sorbent is mixed with gas in the fluidized bed. The characteristic multiphase fluidized flow pattern is determined by many physics variables, among them, the so-called minimum fluidized velocity Umf. The bed height of a fluidized flow is characterized by the distribution of void fraction, which is quantitatively related to the pressure distribution. Pressure drop is highly dependent on Umf. Among many earlier researchers in the field of fluidized beds, Wen and Yu (1966) suggested that for a laminar flow, Umf is proportional to the square of the solid particle size. The relationship is based on solid phase with one uniform size. In reality, the sorbent size is a random quantity that follows a distribution. One simplification in MFIX simulations is to use one single particle size for all of the solid phase. One can reasonably foresee that Umf and the flow pattern of this AX flow will depend on an average of particle size of some sort, such as Sauter mean diameter (SMD), which is around $115 - 118$ μm for AX particles. SMD is defined as the diameter of a sphere that has the same volume/surface area ratio as a particle of interest. Even with a uniform solid particle size, matching the CFD predictions of fluidized beds to lab-scale experimental measurements can be difficult, especially given the idealizations and simplifications often made in the CFD models. With a distributed solid particle size, it becomes even more challenging to use numerical simulations to predict an accurate pressure distribution in the multiphase fluidized bed.

6.2 Model calibration

Table 1 summarizes the experimental output, inputs, and simulation model parameters with their corresponding prior distribution. All six calibration parameters are assumed to be independent and were obtained through a review of CFD literature on fluidized beds (Li et al., 2011; Chao et al., 2011; Herzog et al., 2012; Asegehegn et al., 2011, 2012; Yusuf et al., 2012). The first five calibration parameters are also used by Storlie et al. (2013) with some small deviations in the boundaries of the calibration parameters.

Table 1: Summary of inputs, outputs, and the CFD model parameters with their corresponding prior using the AX sorbent

Outputs $\boldsymbol{d}$: Pressure drop $PDT3820$.

Experimental Inputs $\boldsymbol{x}$: Gas Velocity (GV), [15, 60] SLPM which stands for *standard liter per minute.*

CFD Model parameters $\boldsymbol{t}$:
θ_1 : Coefficient of restitution, particle-particle (*Res.PP*), $\theta_1 \sim \text{Beta}(2.5, 2.5, 0.8, 0.997)$
θ_2 : Coefficient of restitution, particle-wall (*Res.PW*), $\theta_2 \sim \text{Beta}(2.5, 2.5, 0.8, 0.997)$
θ_3 : Friction angle, particle-particle (*FA.PP*), $\theta_3 \sim \text{Beta}(1.2, 2.5, 25.0, 45.0)$
θ_4 : Friction angle, particle-wall (*FA.PW*), $\theta_4 \sim \text{Beta}(1.2, 2.5, 25.0, 45.0)$
θ_5 : Packed bed void fraction (*PBVF*), $\theta_5 \sim \text{Beta}(2.5, 2.5, 0.3, 0.4)$
θ_6 : Particle size (*PSize*), $\theta_6 \sim \text{Beta}(1.2, 2.5, 105.0, 135.0)$

An LHS is used to sample values of the observable input (x_1) in $n = 40$ points, where we obtain experimental measurements. An LHS is also used to determine where to sample $m = 120$ observable input and CFD model parameters $(x_1, t_1, \ldots, t_6)$ for the computer model simulations. We have a model calibration problem with $n = 40$ experimental measurements and $m = 120$ computer model outputs. Two calibration methods have been used to analyze this problem: the proposed BTC and the SBGPC. Table 1 illustrates the prior distribution of the six calibration parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)$ used for the two Bayesian methods. The parameters in the prior distribution of the Bayesian tree are set at $a = 0.6$ and $b = 2$.

Table 1 illustrates the prior distribution of the six calibration parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)$ used for the two Bayesian methods. The parameters in the prior distribution of the Bayesian tree are set at $a = 0.6$ and $bF = 2$ to ensure relatively small number of external nodes. A preliminary analysis was done to determine the other parameters in BTC.

We start by applying the proposed BTC algorithm to this model calibration problem. We run $25,000$ MCMC iterations, the first $5,000$ of which are taken as burn-in. Along with the prior input distributions (blue curves), Figure 10 shows the marginal posterior distributions with their MCMC trace plots of the calibration parameters $(\theta_1, \ldots, \theta_6)$. We observe the sampler mixes well,

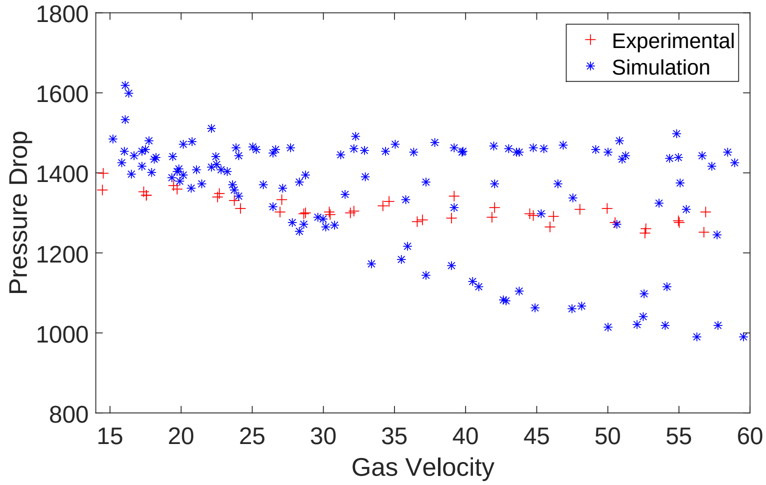
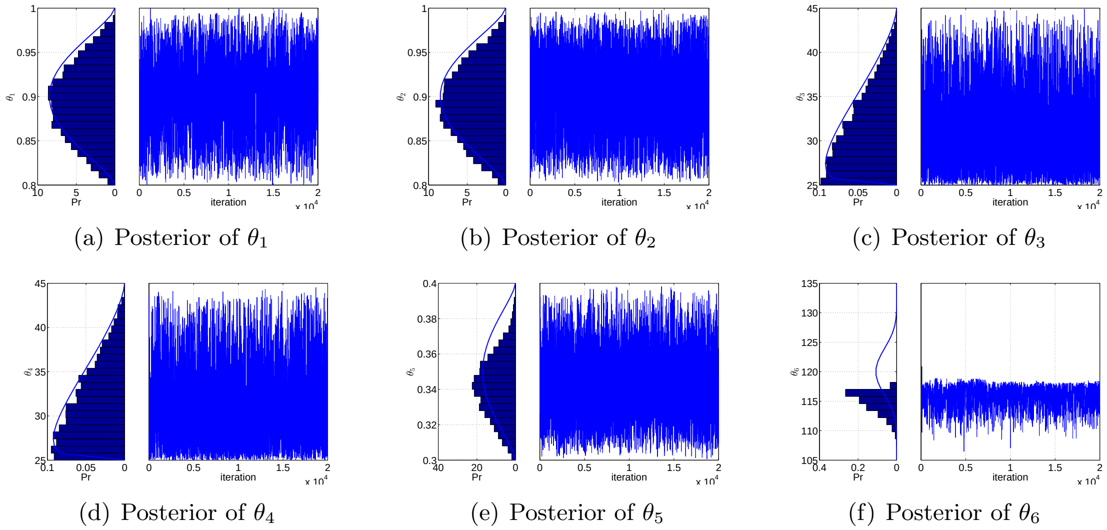


Figure 9: Experimental and computer model data: Experimental observations are denoted by red cross and the computer model observations are depicted by blue stars.



(a) Posterior of θ_1

(b) Posterior of θ_2

(c) Posterior of θ_3

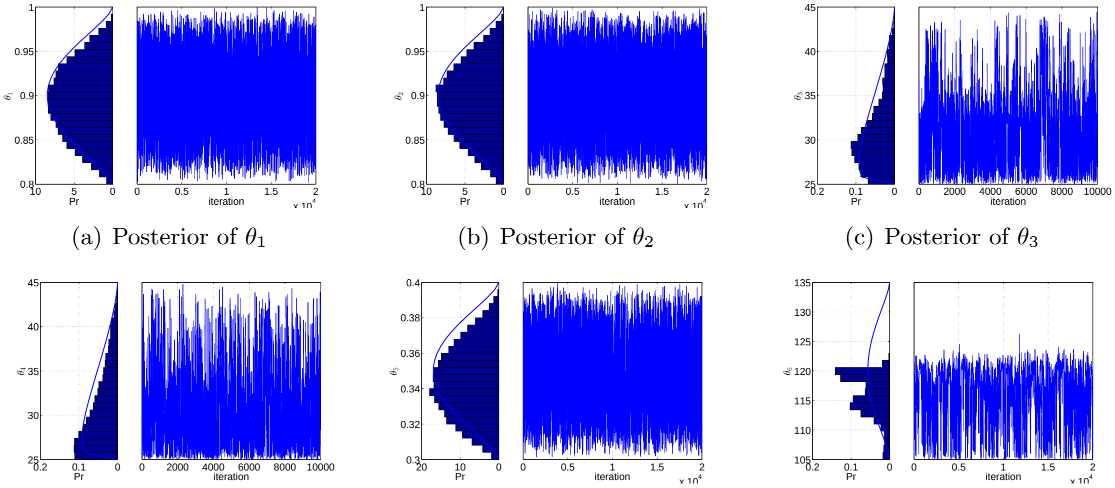
(d) Posterior of θ_4

(e) Posterior of θ_5

(f) Posterior of θ_6

Figure 10: MCMC for the calibration parameter $(\theta_1, \ldots, \theta_6)$ for the proposed TGP calibration.

and the associated ergodic averages, which correspond to the point estimates, converge quickly. The posterior distributions of Res.PP (θ_1), Res.PW (θ_2), θ_3 and θ_4 are similar to their prior distributions. This indicates that the model calibration is not sensitive to these calibration parameters. The posterior distribution of θ_5 has a slight deviation from the prior density indicating a possible sensitivity of the model calibration to packed bed void fraction $(PBVF)$. The most distinct differences between the posterior and prior distributions are observed for the particle size θ_6 calibration parameter. The posterior distribution of θ_6 is concentrated in the first half of the possible values. Values smaller than 118 for the particle diameter are more possible than values larger than 118.



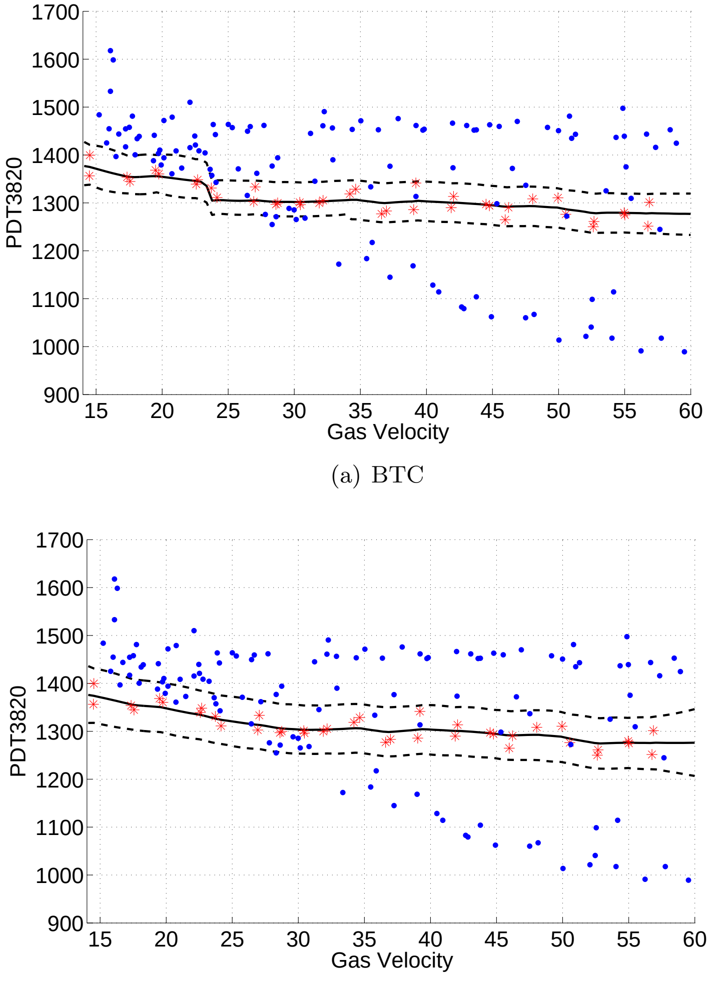
(a) Posterior of θ_1 (b) Posterior of θ_2 (c) Posterior of θ_3

(d) Posterior of θ_4 (e) Posterior of θ_5 (f) Posterior of θ_6

Figure 11: MCMC for the calibration parameter $(\theta_1, \ldots, \theta_6)$ for the SBGPC calibration.

Following the same settings in the MCMC algorithm, we calculate the posterior distributions of the calibration parameters using SBGPC (Figure 11). The prior distribution also is noted with blue curves. SBGPC gives almost exact posterior distributions with BTC for Res.PP, θ_1, and Res.PW, θ_2. When using the SBGPC method, the posterior distribution of θ_3 and θ_4 slightly deviate from the prior distribution. Minor differences between the two methods are observed in these posterior distributions. Both methods show slight deviation of the posterior from the prior for θ_5. However, in the BTC this deviation is more distinct than that observed with SBGPC. Finally, the difference between the two methods estimated posterior distribution of the calibration parameters is more apparent in the particle size, θ_6. The posterior distribution of θ_6 using SBGPC has two modes (one for 121μm and one for 115μm) and appears to have larger variance.

In a real calibration problem, there is no direct evidence to support whether or not one calibration value is better than the other. Finding an effective particle size applied in the simulations from a distribution is a difficult task. Among many compelling factors in the multiphysics multiphase flow, it is believed that the most important factor is the drag because the drag on the solid particles determines how the solid float and mixed in the gas, and thus the fluidized bed height and other important CFD quantities of interest. The drag which results from the relative velocity between the gas and solid provides a lifting force balancing. The downward gravity is approximately proportional to the surface area of the particle. Therefore if the drag is the primary concern, the effective particle size would be close to the SMD. Considering the fact that drag being proportional to the surface area is a rough approximation and other factors such as the particles are not necessary in a uniform sphere form, SMD only provides a reasonable estimate and it is highly valuable to calibrate the effective particle size for CFD simulations. When the posterior distribution of particle size is used, the MFIX simulation results are more in line with the experiment data. When the sorbent particle size falls beyond the posterior distribution, the gas-solid drag force is either too large or too small, resulting in the fluidized bed being either too high or too low. The SMD in our example is equal to 116μm which is closer to the BTC posterior mean for the particle size, θ_6.



;

(a) BTC

(b) SBGPC

Figure 12: Prediction with confidence intervals for the experimental data as a function of the Gas Velocity using BTC (a) and SBGPC (b). Experimental observations are noted by red cross, and the computer model observations are denoted by blue stars.

An important tool for comparing the two methods is the prediction of the real system for each observable input. Figures 12(a) and 12(b) show the prediction mean (solid black line), the 95% prediction bands (two black dashed lines), real system output observation (red stars), and computer model observation as a function of observable input (Gas Velocity) for the BTC and SBGPC, correspondingly. Differences in the prediction means are observed when possible discontinuity has been captured from BTC around Gas Velocity 20 and 30, while in the rest of the observable input the prediction means are very similar. BTC better represents the experimental data on the part where the two models disagree. More difference are observed on the 95% prediction bands (two black dashed lines). SBGPC gives larger prediction bands for the real system as a function of the observable input, while the BTC gives a more realistic representation of the variance as a function of the observable input. BTC seems to split the observable input in two parts where it applies different mean and covariance functions for the surrogate model of the computer model and the discrepancy. The discrepancy function changes as a function of the observable input. For values of Gas Velocity smaller than 24 SLPM, the discrepancy is negative. For Gas Velocity greater than 24 SLPM the discrepancy seems to be close to zero. The variance of the computer model changes also as a function of Gas Velocity.

(a)(b)(c)(d)(e)(f)
MAPMAPMAPMAPMAPMAP
treetreetreetreetreetree
marginal-marginal-marginal-marginal-marginal-marginal-
izedizedizedizedizedized
onononononon
$(x_1, t_1)(x_1, t_2)(x_1, t_3)(x_1, t_4)(x_1, t_5)(x_1, t_6)$

Figure 13: MAP estimation of tree partitioning (dashed lines) for the proposed TGP calibration with the locations and the output of the computer model as a function of the observable input x_1 and each of the six calibration inputs $(t_1, \dots, t_6)$.

From the prediction, it is evident that the BTC model has captured a type of non-staitonarity of the output across the observable input. To better understand the BTC model's usefulness, we present the output as a function of the observable input and one calibration input. Figure 13 shows six different graphs of the computer model output as a function of the observable input and one calibration input. We also show the maximum a posterior (MAP) Bayesian tree calibration partitions with black dashed lines. Two partitions are obvious in this binary tree, which lead to three subregions. For the MAP Bayesian tree, no other partitions occur as a function of two calibration parameters. From these graphs, it is clear that there is a non-stationarity in the computer model. The computer model has different mean and variance as a function of observable input (x_1) Gas Velocity the particle size (t_6). When the Gas Velocity is below 24 SLPM, the fluidized bed is more smooth. When it gets pass that value, the fluidized bed becomes more turbulent. Higher gas flow make the two-phase flow more dynamic, and the particle size will have greater impact on the fluidized bed condition. This is possibly the reason on why at higher inlet flow rate, the bifurcation becomes more apparent and there is a type of discontinuity.

7 Conclusions and Further Work

In this article, we have developed the Bayesian treed calibration (BTC) method for model calibration, which can deal with non-stationarity in the mean, variation, and variance of the computer model and/or its discrepancy function from the true system. The method also is adequate for non-stationarity in the variance of nugget terms. The proposed BTC can be seen as an extension of the standard Gaussian process Bayesian calibration (SGPBC) (Kennedy and O'Hagan, 2001) using Bayesian tree techniques (Gramacy and Lee, 2008; Chipman et al., 1998). Appropriate formulation of the problem, in terms of defining the likelihood and prior distributions, lead to a well-defined Bayesian hierarchical model. Both the observational and calibration inputs are subject to partition. Each output of the binary tree's external nodes represents a model calibration problem where prior distributions are updated in the presence of real data. In the Bayesian inference, we can explore the whole parametric space of the calibration parameters, regardless of the number and the positions of the subregions. We also integrate out all the linear terms associated with the model calibration inside each external node. This, combined with appropriate proposals for the other parameters, leads to more efficient (in terms of acceptance ratio) local proposals in the *Grow* and *Prune* operations. The proposed model is also suitable for parallel computing to speed up computations as suggested in Bayesian treed operations.

A special case of the proposed model involves using many linear terms, such as basis function for the mean, and model the spatial error as independent. The linear terms will explicitly model the large-scale variation and have been proven to work well in relatively smooth response surfaces. The Bayesian smoothing spline analysis of variance (BSS) ANOVA and polynomial chaos are two such examples. One more observation in our examples is the fact that, without the right combination of the proposal for the calibration parameter theta and the GP parameters, the posterior distribution of the calibration parameter $\boldsymbol{\theta}$ may be trapped to local maxima. We leave this for future work. Despite the observed sensitivity of the posterior distribution in the calibration parameter, the posterior of the predictions tends to be less sensitive (more robust). Another possible extension of the proposed model is to the multivariate setting. We can model the covariance function of computer code and discrepancy similar to the one proposed by Konomi et al. (2014a).

In several artificial examples, we have shown that the proposed BTC performs better than the SGPBC (Kennedy and O'Hagan, 2001; Higdon et al., 2004) when the output of the computer model or the real experiment has discontinuity or localized features. Finally, we apply the proposed method to the multiphase flow simulations of the adsorber within a carbon capture system. The CFD model did show some discrepancy to the reality of experimental data, but it also captures the trend of the physical reality reasonably well. The discrepancy term depends on the value of the Gas Velocity. Gas Velocity values smaller than 24 have a negative discrepancy, while those greater than 24 seems to be close to zero with bigger variance. The proposed BTC can capture these different behaviors, as well as those of the output when we change the calibration parameter associated with the particle size (*PSize*) θ_6.

Source code

Matlab-code for BTC: The BTC_JASA file contains the main code (written in Matlab) to perform the proposed BTC described in the article. The file also contains the three examples described in this paper. Details on using

Acknowledgments This work was supported by the Applied Mathematics Program within the Department of Energy Office of Advanced Scientific Computing Research as part of the Collaboratory on Mathematics for Mesoscopic Modeling of Materials. Computations were performed using the computational resources of Pacific Northwest National Laboratory (PNNL) Institutional Computing cluster systems and the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory. The PNNL is operated by Battelle for the US Department of Energy under Contract DE-AC05-76RL01830. We also thank Dr. C. Storlie for providing his pre-print paper about BSS-ANOVA calibration. We thank the reviewers and the editors for their valuable comments.

References

Asegehegn, T., Schreiber, M., and Krautz, H. (2011), "Investigation of bubble behavior in fluidized beds with and without immersed horizontal tubes using a digital image analysis technique," *Powder Technology*, 3, 248–260.

— (2012), "Influence of two- and three dimensional simulations on bubble behavior in gas-solid fluidized beds with and without immersed horizontal tubes," *Powder Technology*, 219, 919.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy,

R. J., Paulo, R., Sacks, J., and Walsh, D. (2007), “Computer model validation with functional output,” *Annals of Statistics*, 35, 1874–1906.

Chao, Z., Wang, Y., Jakobsen, J., Fernandino, M., and Jakobsen, H. (2011), “Derivation and validation of a binary multifluid eulerian model for fluidized beds,” *Chemical Engineering Science*, 66, 3605–3616.

Chen, M.-H. and Schmeiser, B. (1998), “Monte Carlo Methods on Bayesian Analysis of Constrained Parameter Problems with Normalizing Constants,” *Journal of Computational and Graphical Statistics*, 7, 1–22.

Chipman, H., George, E., and McCulloch, R. (1998), “Bayesian CART Model Search,” *Journal of the American Statistical Association*, 93, 935–960.

— (2002), “Bayesian Treed Models,” *Machine Learning*, 48, 303–324.

Gelfand, A. E. and Smith, A. F. M. (1990), “Sampling-Based Approaches to Calculating Marginal Densities,” *Journal of the American Statistical Association*, 85, 398–409.

Gramacy, R. B. and Lee, H. K. H. (2008), “Bayesian treed Gaussian process Models with an application to computer modeling,” *Journal of the American Statistical Association*, 103, 1119–1130.

— (2009), “Adaptive Design and Analysis of Supercomputer Experiments,” *Technometrics*, 51, 130–145.

— (2012), “Cases for the nugget in modeling computer experiments,” *Statistics and Computing*, 22, 713–722.

Green, P. (1995), “Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination,” *Biometrika*, 82, 711–732.

Hastings, W. K. (1970), “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, 57, 97–109.

Herzog, N., Schreiber, M., Egbers, C., and Krautz, H. (2012), “A comparative study of different cfd-codes for numerical simulation of gas-solid fluidized bed hydrodynamics,” *Computers & Chemical Engineering*, 39, 41–46.

Higdon, D. (1998), “A process-convolution approach to modeling temperatures in the North Atlantic Ocean,” *Journal of Environmental and Ecological Statistics*, 5, 173–190.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer model calibration using high-dimensional output," *Journal of the American Statistical Association*, 103, 570–583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J., and Ryne, J. (2004), "Combining field data and computer simulations for calibration and prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999), "Bayesian model averaging: a tutorial," *Statistical science*, 382–401.

Kennedy, M. and O'Hagan, A. (2001), "Bayesian calibration of computer models (with discussion)," *Journal of the Royal Statistical Society (Series B)*, 68, 425–464.

Konomi, B., Karagiannis, G., Sarkar, A., and Lin, G. (2014a), "Bayesian Treed Multivariate Gaussian Process with Adaptive Design: Application to a Carbon Capture Unit," *Technometrics*, 56, 145–158.

Konomi, B., Sang, H., and Mallick, B. (2014b), "Adaptive Bayesian nonstationary modeling for large spatial datasets using covariance approximations," *Journal of Computational and Graphical Statistics*, 23, 802–829.

Krutka, H. and Sjostrom, S. (2013), "Evaluation of Solid Sorbents as a Retrofit Technology for CO2 Capture from Coal-fired Power Plants," Tech. rep., Technical Report, DOE Award Number DE-NT0005649, Report Number 05649FR01.

Lee, H., Taddy, M., Gramacy, R., and Gray, G. (2010), "Designing and Analyzing a Circuit Device Experiment Using Treed Gaussian Processes." in *The Oxford Handbook of Applied Bayesian Analysis*, eds. O'Hagan, A. and West, M., Oxford University Press.

Li, T., Dietiker, J., Zhang, Y., and Shahnam, M. (2011), "Cartesian grid simulations of bubbling fluidized beds with a horizontal tube bundle," *Chemical Engineering Science*, 66, 6220–6231.

McKay, M., Beckman, R., and Conover, W. (1979), "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, 21, 239–245.

Mueller, P. (1993), "Alternatives to the Gibbs sampling scheme," Tech. rep., Institute Statistics and Decision Sciences, Duke University.

Paciorek, C. and Schervish, M. (2006), "Spatial modelling using a new class of nonstationary covariance functions," *Environmetrics*, 17, 483–506.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York, NY : Springer-Verlag, 1st ed.

Stein, M. L. (1999), *Interpolation of Spatial Data: Some Theory for Kriging. 2nd edition*, New York: Springer.

Storlie, C., Lane, W., Ryan, E., Gattiker, J., and Higdon, D. (2013), "Calibration of Computational Models with Categorical Parameters and Correlated Outputs via Bayesian Smoothing Spline ANOVA," *Journal of the American Statistical Association*, 1, (in review).

Vernon, I., Goldsteiny, M., and Bowerz, R. (2010), "Galaxy Formation: a Bayesian Uncertainty Analysis," *Bayesian Analysis*, 4, 619–670.

Wen, C. Y. and Yu, Y. H. (1966), "A generalized method for predicting the minimum fluidization velocity," *American Institute of Chemical Engineers Journal*, 12, 610–612.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., and Keller-McNult, S. (2006), "Combining Experimental Data and Computer Simulations, With an Application to Flyer Plate Experiments," *Bayesian Analysis*, 1, 765–792.

Yusuf, R., Halvorsen, B., and Melaaen, M. (2012), "An experimental and computational study of wall to bed heat transfer in a bubbling gassolid fluidized bed," *International Journal of Multiphase Flow*, 42, 923.

A Modeling with two independent trees for the computer model output and the discrepancy

Let assume that we use two independent trees for the computer model output and the discrepancy. One further assumption which will facilitate the model, in this case, is that the error term of the computer model output depends on the tree structure of $\eta(\cdot,\cdot)$ and the error term of the experiment depends on the tree structure associated with $\delta(\cdot)$. Both these assumptions are made such that we do not construct other trees for these quantities. Additionally, assume that $\mathcal{T}_\eta$ and $\mathcal{T}_\delta$ are the two trees associated with the computer code and the discrepancy respectively.

The mean of z_i is $E(z_i) = \beta_{\eta,k}^T h_\eta(\boldsymbol{x}_i;\boldsymbol{\theta}) + \beta_{\delta,k'}^T h_\delta(\boldsymbol{x}_i)$ where k and k' represent the k^{th} and k'^{th} external node of the computer code and the discrepancy respectively. The mean of y_j is $E(y_j) = \beta_{\eta,k}^T h_\eta(\boldsymbol{x}_i;\boldsymbol{\theta})$.

The variance for the experimental and the computer code output are:

$$Var(z_i) = c_{\eta,k}((\boldsymbol{x}_i, \boldsymbol{\theta}), (\boldsymbol{x}_i, \boldsymbol{\theta})) + c_{\delta,k'}(\boldsymbol{x}_i, \boldsymbol{x}_i) + \sigma_\epsilon^2$$

$$Var(y_j, y_{j'}) = c_{\eta,k}((\boldsymbol{x}_j, \boldsymbol{t}_j), (\boldsymbol{x}_{j'}, \boldsymbol{t}_{j'})) + \sigma_v^2$$

For the covariance function we have to distinguish among four cases which depend on the location of the input points.

- First case: Two different inputs belong at the same external node k of the tree for the computer code ($\mathcal{T}_\eta$) and at the same external node k' of the tree for the discrepancy function ($\mathcal{T}_\delta$). This means that for the observation of experiment the input $(\boldsymbol{x}_i, \boldsymbol{\theta})$ and $(\boldsymbol{x}_{i'}, \boldsymbol{\theta})$ belong to the external node k of $\mathcal{T}_\eta$ and the external node k' of $\mathcal{T}_\delta$.

- Second case: Two different inputs belong to the same external node k of the tree for the computer code ($\mathcal{T}_\eta$) but different external nodes of the tree for the discrepancy function ($\mathcal{T}_\delta$).

- Third case: Two different inputs belong to different external nodes of $\mathcal{T}_\eta$ but same external node k' of $\mathcal{T}_\delta$.

- Forth case: Two different inputs belong to different external nodes of $\mathcal{T}_\eta$ and different external node of $\mathcal{T}_\delta$.

Given the form of the mean and the covariance function, the Bayesian inference is more challenging. It is not straightforward how unknown parameters, such as the linear coefficients the GP mean function, can be integrated out. Therefore, the dimension of the sampling space in the RJ moves can be large and cause the RJ to perform poorly. Also, the number of the parameters increases significantly.

B Metropolis within Gibbs Sampler for the GP Parameters at External Node k

Let $\boldsymbol{\chi}_k = (\boldsymbol{\phi}_k, \boldsymbol{\sigma}_k^2, \boldsymbol{\tau}_k) = (\phi_{\eta,k,1}, \ldots, \phi_{\eta,k,(q+p)}, \phi_{\delta,k,1}, \ldots, \phi_{\eta,k,q}, \sigma_\eta^2, \sigma_\delta^2, \tau_{e,k}^2, \tau_{\ni,k}^2)$. For each component of $\boldsymbol{\chi}_k$, $\chi_{k,j}$ for $j = 1, \ldots, (2q+p+4)$, we perform Metropolis within Gibbs as in (Mueller, 1993). For any step of the Gibbs sampler that does not have a close form conditional posterior distribution $p(\chi_{k,j}|\boldsymbol{d}_j, \chi_{k,1}, \ldots, \chi_{k,j-1}, \chi_{k,j-1}, \ldots, \chi_{j,(2q+p+4)})$, substitute a MH sampler.

For $j = 1, \ldots, (2q+p+4)$, given $\boldsymbol{\chi}_{k,(-j)}^{(t)} = (\chi_{k,1}^{t+1}, \ldots, \chi_{k,j-1}^{t+1}, \chi_{k,j-1}^{t}, \ldots, \chi_{k,(2q+p+4)}^{t})$:

1. Generate $\chi^*_{k,j} \sim q_j(\chi^*_{k,j}|\chi^{(t)}_{k,j}) \equiv \log N(\chi^*_{k,j}|\chi^{(t)}_{k,j})$ from a log Normal distribution.

2. Calculate:

$$r_k = \frac{p(\chi^*_{k,j}|\boldsymbol{d}_k, \boldsymbol{\chi}^{(t)}_{k,(-j)})q_j(\chi^{(t)}_{k,j}|\chi^*_{k,j})}{p(\chi^{(t)}_{k,j}|\boldsymbol{d}_k, \boldsymbol{\chi}^{(t)}_{k,(-j)})q_j(\chi^*_{k,j}|\chi^{(t)}_{k,j})}. \tag{10}$$

3. Set $\chi^{(t+1)}_{k,j} = \chi^*_{k,j}$ with probability $\min{(1, r_k)}$ and $\chi^{(t+1)}_{k,j} = \chi^{(t)}_{k,j}$ with the remaining probability.

In this algorithm, the MH step is performed only once at each iteration. Chen and Schmeiser (1998) note that multiple MH steps are unnecessary. A precise approximation of the conditional probability does not automatically lead to a better approximation of the join distribution, and a single step may be beneficial to the sampler's speed.