

Multi-block Parameter Calibration in Computer Models

Cheoljoon Jeong, Ziang Xu, Albert S. Berahas, Eunshin Byon

Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109
cjeong@umich.edu; xuziang@umich.edu; aberahas@umich.edu; ebyon@umich.edu

Kristen Cetin

Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824
cetinkri@msu.edu

Parameter calibration aims to estimate unobservable parameters employed in a computer model by utilizing physical process responses and computer model outputs. In the literature, existing studies calibrate all parameters simultaneously using an entire dataset. However, in certain applications, some parameters are associated with only a subset of data. For example, in the building energy simulation, cooling (heating) season parameters should be calibrated using data collected during the cooling (heating) season only. This study provides a new multi-block calibration approach that considers such heterogeneity. Unlike existing studies that build emulators for the computer model response, such as the widely used Bayesian calibration approach, we consider multiple loss functions to be minimized, each for a block of parameters that use the corresponding dataset, and estimate the parameters using a nonlinear optimization technique. We present the convergence properties under certain conditions and quantify the parameter estimation uncertainties. The superiority of our approach is demonstrated through numerical studies and a real-world building energy simulation case study. Supplementary materials for this article are available online.

Key words: Building energy, nonlinear optimization, simulation

History: This paper was submitted on March 28, 2022, revised on October 31, 2022, and accepted on March 1, 2023

1. Introduction

Recent advancements in computing have enabled the adoption of computer models as an essential tool to design and control systems. In general, a computer model consists of a set of mathematical functions that are based on complex physics-based first principals in order to closely resemble a real-world system. A computer model is usually utilized with a simulator. A simulator receives input variables that are typically observable or controllable, such as operational conditions, and it generates outputs through simulation. Besides the input variables, additional parameters need to be specified *a priori* to run simulation. These parameters are oftentimes not observable, thus their values should be estimated using either physical laws or data. When physical laws that identify

appropriate parameter values are not available, one can use a data-driven approach using data collected from the physical system. Such a data-driven procedure is called *parameter calibration* in the literature (Kennedy and O’Hagan 2001).

Over the last couple of decades, parameter calibration has been studied in many fields such as biology, chemistry, climatology, and engineering (Santner et al. 2018). We provide a detailed literature review in Section 3. Existing studies, by and large, utilize limited datasets when physical trials are limited and computer models are expensive to run. To accommodate data scarcity, they build surrogate models for computer models and calibration is performed with surrogate models. Typically, they generate data from a computer model *a priori* through a fixed set of simulation runs. With the resulting data, they build a response surface (or emulator) for the computer model output using Gaussian processes (GPs) or other statistical models (Higdon et al. 2004, Tuo et al. 2021). Consequently, calibration accuracy highly depends on the emulator accuracy. When the computer model data are generated at less informative design points, e.g., values far from (unknown) true parameter values, the emulator may not accurately characterize the response surface near the true parameter values, leading to inaccurate calibration results.

Recently, Liu et al. (2021) discussed a necessity to develop a new calibration procedure for big data settings. In many operational systems, thanks to the recent advances in sensing technology and data acquisition systems, massive amounts of observational data from an actual physical system have become available. Further, a sufficiently large number of simulation data can be generated through a medium- or low-fidelity simulator or using advanced computing facilities, e.g., supercomputers. When a large number of field observations and simulation data are available, Liu et al. (2021) presented a new approach to directly utilize data without building emulators.

The aforementioned existing literature calibrates all parameters in a similar manner despite of heterogeneous data requirement. That is, they calibrate all parameters at once with the same dataset. In some applications, however, some parameters are associated with certain specific operating conditions or a subset of datasets. Consider a building energy model (BEM). Among many parameters that need to be calibrated, some parameters are season-dependent, and thus they should be employed for simulating building energy operations only in specific seasons (Xu et al. 2021). For example, “cooling supply air flow rate” is a cooling-season parameter, whereas “heating supply air flow rate” is associated with a heating season. On the other hand, other parameters, e.g., “fan total efficiency”, are global parameters that need to be employed throughout the year. This BEM parameter calibration, as our motivating application, will be explained in more detail in Section 2.

From a data science point of view, these season-dependent (or block-dependent) parameters should be calibrated using data (both field operational and simulation-generated data) associated with the corresponding season (or block) only. That is, one needs to use a subset of data. When

the required dataset for each group of parameters is exclusive from one another, one can calibrate parameters separately for each group. However, when the subsets for different groups overlap, prior methods become inappropriate.

This study develops a new calibration approach where parameters can be divided into multiple groups and each group of parameters is associated with a subset (or block) of the entire dataset. Specifically, we estimate the parameters by minimizing multiple loss functions, each of which is associated with each group of parameters. Our proposed approach is useful when large computer experiments can be conducted, e.g., from a low- to medium-fidelity computer model.

We summarize and highlight the key contribution of this paper as follows.

- Our approach estimates unobservable parameters employed in a computer model where parameters are associated with different subsets of observational data that possibly overlap with one another. To the best of our knowledge, this is the first calibration study that takes this special problem structure into consideration in the calibration procedure.
- One salient feature of the proposed approach is that it adaptively generates data *on the fly* from the computer model during the calibration procedure so that the most appropriate data can be produced from the computer model. This is different from the traditional frameworks that generate simulation data at pre-selected data points *a priori*.
- To minimize multiple loss functions, we design a nonlinear optimization algorithm and derive its associated convergence guarantees. Under certain (reasonable) conditions, our analysis provides strong guarantees for our algorithm for different classes of functions.
- We conduct numerical experiments on a wide range of settings and a case study with real data to demonstrate the performance of the proposed approach. The results indicate its superiority with respect to multiple criteria, including estimation accuracy, uncertainty quantification, computational efficiency, and scalability, over alternative approaches.

A preliminary version of this work is presented in a short conference paper (Xu et al. 2021). In this paper, we substantially extend the analysis to connect the calibration procedure with the statistical parameter estimation, derive and present convergence guarantees, quantify estimation uncertainties, and provide extensive empirical results via numerical examples and a case study.

The rest of this paper is organized as follows. Section 2 discusses the BEM application as our motivating example and other examples from various applications. Section 3 provides a literature review of related methods. Section 4 gives a formulation of the problem. Section 5 develops a new algorithm to solve the problem and studies the convergence properties. We evaluate the performance of our methodology using numerical examples and building energy simulation case study in Sections 6 and 7, respectively. Finally, Section 8 provides a brief conclusion.

2. Applications

In this section, we introduce the motivating building energy application and discuss the wide applicability of our proposed method.

2.1. Motivating Application: Building Energy Model

A physics-based BEM, which simulates the building energy operations, can play an important role in optimizing building design and operations. To effectively use the BEM, several parameters that specify building characteristics need to be estimated using operational data. Table 1 lists the parameters employed in the BEM simulator, called EnergyPlus, which is developed and maintained by the U.S. Department of Energy’s National Renewable Energy Laboratories (U.S. Department of Energy 2019).

A unique aspect of the BEM is the seasonal dependency of the parameters, which is the focus of our study. Among the parameters, some parameters related to lighting, domestic hot water, window material, and ventilation can be considered as global parameters that are associated with the entire year-long operational data and simulation. On the contrary, cooling- and heating-related parameters are only associated with their seasonal portions of the year-long data. Interestingly, it is possible that different seasons may overlap in some areas. For example, in Texas, buildings operate heating devices from November to April, whereas cooling systems run from March to November. Hence, during some periods of time, they operate both heating and cooling devices. Motivated by the BEM parameter calibration, this study develops a new calibration approach when parameters need to be calibrated with different portions of data.

Table 1 Seasonal dependency of the BEM parameters

Parameter group	Description	Schedule in Texas
global parameters θ_g	lighting level ceiling fan design level maximum supply air temperature heater thermal efficiency fan total efficiency ventilation design flow rate	January - December
cooling-season parameters θ_c	solar transmittance gross rated cooling COP gross rated total cooling capacity cooling supply air flow rate	March - November
heating-season parameters θ_h	fuel nominal capacity burner efficiency heating supply air flow rate	January - April, November, December

2.2. Other Applications

There are various applications where our proposed method can be utilized. First, consider the hydrological computer model that simulates water flow to estimate natural system behaviors affected by climate change, land use, etc. This model can be used to investigate the rainfall-runoff relationship for the windrow compositing pad to estimate the amount of runoff (Duncan et al. 2013, Bhattacharjee et al. 2019). To accurately simulate rainfall-runoff dynamics, several parameters, including the depth of surface, depth of subsurface, saturated hydraulic conductivity of the gravel media, and saturated hydraulic conductivity of the supporting soil below the media, need to be calibrated. Among them, the depth of surface and depth of subsurface can be regarded as global parameters that are associated with the entire year-long data, whereas others are possibly season-dependent parameters associated only with their seasonal portion of data due to the fact that they are generally affected by weather.

Another application is the wind flow model that characterizes spatially heterogeneous wind patterns within a wind farm due to the interactions among turbines (You et al. 2017, 2018). Recently, Howland et al. (2022) presented a flow model that consists of two sub-models: the power-yaw model and wake effect model. The power-yaw model predicts the power production of upwind turbines (yawed turbines) with a parameter λ_p that adjusts the power generation amount in a yaw-misalignment setting. The wake effect model predicts the wind speed deficit at downstream turbines (waked turbines) due to the shading effect of upwind turbines (Liu et al. 2021) so that it estimates the power production of the waked turbines. This wake effect model takes the wake spreading coefficient k_w and the proportionality constant of the modeled Gaussian wake σ_0 as parameters. Among the parameters, λ_p can be considered as a global parameter that is employed in all wind conditions. On the other hand, k_w and σ_0 depend on atmospheric conditions (Howland et al. 2022), which are calibrated using a portion of data collected under specific conditions.

Lastly, consider a system with a hierarchical block structure. One example could be a pandemic simulation model for infectious diseases. When an outbreak, such as COVID-19, occurs, it is important to predict the spread of infectious diseases. In the simulation models, some parameters such as infection rate, reinfection rate, and positive test rate should be calibrated to achieve accurate predictions and they are usually associated with specific geographical areas. These areas overlap hierarchically from a country level, followed by state, county, city levels, etc. and each parameter should be calibrated using the data obtained from the associated area of interest. For instance, data to predict the infection rate at a city level overlap with that at a county level.

3. Literature Review

In this section, we provide a literature review of several relevant methodologies.

3.1. Statistical Calibration Studies

The seminal work in modern parameter calibration is Bayesian calibration (Kennedy and O’Hagan 2001, Higdon et al. 2004). It uses a linear linkage model to connect the responses of physical processes with computer model’s outputs, and the calibration parameters and hyperparameters in mean vectors and covariance matrices of the GP emulators are estimated from posterior distributions. The Bayesian calibration methodology has gained popularity due to its ability to quantify estimation uncertainties with limited field observations and/or scant simulation data (Kennedy and O’Hagan 2001, Higdon et al. 2004, Gramacy et al. 2015). Recently, Tuo and Wu (2016) suggested that the Bayesian approach could result in unreasonable estimates for imperfect computer models. Here, an imperfect computer model implies that the outputs of the model, even with the optimally calibrated parameters, are different from the expected outcomes of a physical process. To address this limitation, Tuo and Wu (2015) presented a new calibration approach from a frequentist point of view.

Among these statistical methods, studies in the building energy literature have mainly adopted the Bayesian calibration approach (Chong and Menberg 2018, Coakley et al. 2014). However, the Bayesian approach has several limitations. First, it is computationally intensive, so studies mostly use aggregated data, e.g., monthly (Heo et al. 2012, 2013, 2015, Kim and Park 2016, Kristensen et al. 2017b, Li et al. 2016, Lim and Zhai 2017, Sokol et al. 2017, Tian et al. 2016) or annual (Booth et al. 2013) data. Some studies use hourly data, but they use a small subset of hourly data, instead of using the whole dataset collected over a sufficiently long period of time (Chong et al. 2017, Chong and Lam 2017, Manfren et al. 2013, Menberg et al. 2017, Kristensen et al. 2017a). In Chong et al. (2017), only 80 hourly data samples are selected from the dataset collected for three months. To relieve the computational burden, a lightweight Bayesian calibration that uses a linear regression emulator is presented in Li et al. (2016). Menberg et al. (2017) employ the Hamiltonian Monte Carlo sampling to obtain the posterior distribution more efficiently. Despite these advances, the Bayesian calibration approach still remains computationally demanding, which limits the use of large-size datasets.

Second, the results from Bayesian calibration are sensitive to the prior specification (Liu et al. 2021). Our analysis, which will be discussed in more detail in Section 7, indicates that the posterior density is heavily affected by the prior specification. When a non-informative prior is employed, the resulting posterior tends to become non-informative, showing a relatively flat posterior distribution. We believe this is because the use of limited data (either aggregated monthly or yearly data, or a small number of selected hourly data) provides insufficient information to generate a meaningful posterior. While a more informative prior leads to a sharper posterior, it requires domain knowledge for the appropriate prior setting.

Further, the Bayesian approach in the BEM calibration studies limits the number of calibration parameters to 2 to 5. This is because the hyperparameter estimation of the covariance function becomes computationally prohibitive as the problem size grows. Further, compounded with the aforementioned issues of limited data, when multiple parameters are calibrated, overparameterization may occur, resulting in a relatively flat posterior, which does not generate meaningful calibration results.

Most importantly, existing approaches cannot handle multiple blocks of overlapping data. In particular, when the block sizes are different, one might suggest that the problem is similar to the class imbalance problem where the data size of each class is substantially different (Byon et al. 2010). In such cases, one possible solution could be to impose different weights on each data block. However, how to differentiate weights across blocks is not straightforward. In particular, when data blocks overlap, if the algorithm calibrates the parameters with their small portion of data with a higher weight, it would also affect other blocks with larger portions of data. Thus, the weight should consider the overlapping portion as well as the data size of each block; devising a proper weighting scheme with rigorous justification is challenging.

3.2. Other Related Studies

We also review two other streams of work that are relevant to this study, namely, adaptive sampling and multi-task learning. First, adaptive sampling, also known as sequential design in the statistical literature and active learning in the machine learning literature, has been actively studied. The idea is to collect informative data points sequentially in order to construct surrogates and/or minimize a black box function by accounting for the trade-off between exploration and exploitation (Liu et al. 2018). First, in surrogate modeling studies that often use GPs, adaptive sampling strategies choose the next design point that maximizes predictive variance or maximizes the average reduction in variance (Gramacy 2020, Liu et al. 2018). These studies aim to construct accurate surrogates.

Second, Bayesian optimization (BO) has received much attention recently for minimizing expensive black box function. BO proceeds by modeling the objective function via a GP, optimizing an acquisition function to find the next design point and then updating the posterior distribution of the GP. Acquisition functions, e.g., expected improvement (Jones et al. 1998) and lower confidence bound (Srinivas et al. 2010), attempt to strike a balance between the exploration of a new design point with high uncertainty and exploitation with a low objective function value. More details on BO can be found in Shahriari et al. (2015) and Frazier (2018). Typically, BO aims to minimize a single objective. Although the multi-task BO (Swersky et al. 2013) is proposed to handle multiple objective functions, it uses the same training data for all objectives to perform optimization. Our proposed method solves a multi-objective problem with multiple blocks of data that possibly overlap with one another.

On the other hand, multi-task learning aims to learn multiple tasks simultaneously by leveraging knowledge learned from one or some tasks while exploiting commonalities and differences (Zhang and Yang 2021). It helps to improve prediction accuracy and computational efficiency for learning multiple tasks, compared to individual task learning. In principle, the proposed multi-block calibration methodology can be regarded as a special case of multi-task learning, if we consider calibrating each block of parameters as each task; updated parameters associated with one block affect the loss function in other blocks so that loss function values benefit from or are hurt by others. However, multi-task learning is typically performed in a supervised learning setting with a fixed dataset, whereas our proposed method exploits adaptive sampling. Moreover, in multi-task learning, each task is learned with its own dataset which is disjoint from other tasks' data, however, our approach accounts for overlapping blocks of data.

4. Problem Formulation

We first present a calibration problem in a general single-block calibration setting in Section 4.1 and extend it to a multi-block parameter calibration problem in Section 4.2.

4.1. Single-block Parameter Calibration

Let $\mathbf{x} \in \Omega \subseteq \mathbb{R}^d$ denote the vector of physically observable input variables of dimension $d \in \mathbb{Z}_+$ in a system, where Ω is a convex and compact region of input variables that is a subset of \mathbb{R}^d . Let $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ denote a set of calibration parameters of dimension $p \in \mathbb{Z}_+$. Let $y(\mathbf{x}) \subseteq \mathbb{R}$ denote the noisy field observational data of its true physical process $\zeta(\mathbf{x})$ at input \mathbf{x} . Let $\eta(\mathbf{x}, \boldsymbol{\theta}) \subseteq \mathbb{R}$ denote the response from the computer model, simulating the true process $\zeta(\mathbf{x})$ at input \mathbf{x} . For instance, in the building energy simulation, \mathbf{x} can be hourly environmental condition such as temperature, $y(\mathbf{x})$ the actual hourly electricity consumption, and $\eta(\mathbf{x}, \boldsymbol{\theta})$ the electricity consumption generated from BEM. The unknown calibration parameter vector $\boldsymbol{\theta}$ needs to be estimated using $y(\mathbf{x})$ and $\eta(\mathbf{x}, \boldsymbol{\theta})$.

Assuming the simulator represents the physical process accurately, Higdon et al. (2004) present the regression model to connect physical process responses with computer model outputs as

$$y(\mathbf{x}_j) = \eta(\mathbf{x}_j, \boldsymbol{\theta}) + \epsilon_j, \quad j = 1, 2, \dots, n, \quad (1)$$

where n denotes the number of observations and ϵ_j an observation error that follows an independently and identically distributed (iid) normal distribution, $\epsilon_j \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$.

The likelihood function of $\boldsymbol{\theta}$ is then

$$\mathcal{L}(\boldsymbol{\theta}, \sigma^2 | y(\mathbf{x})) = \prod_{j=1}^n \mathcal{N}(\eta(\mathbf{x}_j, \boldsymbol{\theta}), \sigma^2), \quad (2)$$

or more explicitly,

$$\mathcal{L}(\boldsymbol{\theta}, \sigma^2 | y(\mathbf{x})) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}))^2 \right\}. \quad (3)$$

To estimate $\boldsymbol{\theta}$, let us consider the maximum likelihood estimation (MLE) method. The log-likelihood function is

$$\ell(\boldsymbol{\theta}, \sigma^2 | y(\mathbf{x})) = -\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{j=1}^n (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}))^2. \quad (4)$$

We now maximize the log-likelihood to obtain $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ given σ^2 as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{MLE}} &= \arg \max_{\boldsymbol{\theta} \in \Theta} \frac{1}{n} \ell(\boldsymbol{\theta} | y(\mathbf{x})) \\ &= \arg \min_{\boldsymbol{\theta} \in \Theta} \frac{1}{n} \sum_{j=1}^n (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}))^2 := F(\boldsymbol{\theta}). \end{aligned} \quad (5)$$

Note that maximizing the likelihood function is equivalent to minimizing the empirical loss function that quantifies the discrepancy between the computer model output $\eta(\mathbf{x}, \boldsymbol{\theta})$ and physical process data $y(\mathbf{x})$ with L_2 norm. This calibration procedure that minimizes the empirical loss aligns with the idea of L_2 calibration studied in Tuo and Wu (2015). Tuo and Wu (2015) aimed to find the parameters that minimize the L_2 distance between the physical process responses and computer model outputs. Under the limited data setting, they built emulators for both computer model outputs and physical process responses. However, we consider a situation where a sufficiently large number of physical observational data is available and a computer model is relatively cheap to run.

With data availability, instead of generating data from a computer model at pre-defined design points, we want to generate computer model data more effectively so more useful data can be used for the calibration purpose and directly use it to estimate the parameters without building emulators. To this end, we minimize the loss functions in (5) with respect to $\boldsymbol{\theta}$. As the computer model output $\eta(\mathbf{x}, \boldsymbol{\theta})$ is likely nonlinear, we consider nonlinear optimization techniques. Consider a first-order optimization method, i.e., gradient descent (GD). The GD method works by iteratively updating $\boldsymbol{\theta}$ with $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla F(\boldsymbol{\theta})$ until a suitable termination criterion is satisfied, where $F(\boldsymbol{\theta})$ is the empirical loss function at $\boldsymbol{\theta}$, $\nabla F(\boldsymbol{\theta})$ represents its gradient, and α is a step size. Here, because $\eta(\mathbf{x}, \boldsymbol{\theta})$, as the output from a black box computer model, does not have a closed-form expression, $\nabla F(\boldsymbol{\theta})$ can be approximated using the central finite difference

$$[\nabla F(\boldsymbol{\theta})]_i \approx \frac{F(\boldsymbol{\theta} + h\mathbf{e}_i) - F(\boldsymbol{\theta} - h\mathbf{e}_i)}{2h}, \quad (6)$$

where $h > 0$ is a small disturbance to the i^{th} component of θ , e_i denotes a $p \times 1$ vector with i^{th} element being one and others zero.

One key advantage of using nonlinear optimization is that it can adaptively generate computer model data $\eta(\mathbf{x}, \theta)$. In other words, as θ is updated through the iterative GD process, we can run simulations to get $\eta(\mathbf{x}, \theta)$ at the newly refined θ . It allows us to get more informative data toward minimizing the empirical loss (or maximizing the likelihood).

4.2. Multi-block Parameter Calibration

The single-block parameter calibration presented in Section 4.1 treats all parameters equally using the entire dataset. We propose a new block-wise calibration approach when some parameters are associated with a specific subset of the entire dataset.

Recall that in the BEM application, we have three distinct groups of parameters, denoted by θ_g , θ_c , and θ_h in Table 1. The parameter set θ_g consists of global parameters so it should be globally employed with the entire year-long simulation. On the other hand, θ_c and θ_h include season-specific parameters, associated with only cooling-season and heating-season portions of the entire simulation, respectively. Let us generalize the problem structure, as illustrated in Figure 1. Consider that the entire parameter set θ consists of B groups of parameters, i.e., $\theta = [\theta_1, \theta_2, \dots, \theta_B]$, where $\theta_b \in \mathbb{R}^{p_b}$ denotes the b^{th} group of parameters. For instance, in BEM, we have $\theta_1 = \theta_g$, $\theta_2 = \theta_c$, and $\theta_3 = \theta_h$ with $B = 3$. Suppose further that a parameter vector θ_b is associated with only a subset of entire dataset for each $b = 1, 2, \dots, B$ (see Figure 1). Note that these subsets are not necessarily disjoint.

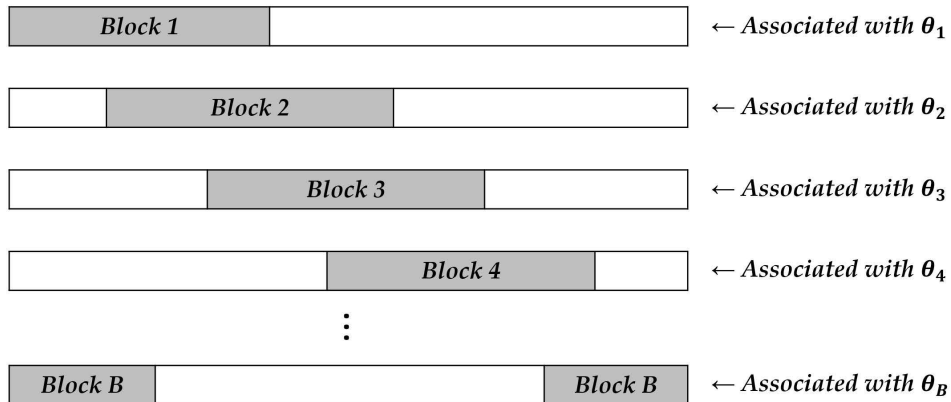


Figure 1 Illustration of multiple data blocks (Here, the data include both physical process data $(x_j, y(x_j))$ and generated data $(x_j, \eta(x_j, \theta))$)

With this problem structure, we calibrate θ_b using its corresponding portion of dataset, i.e., the b^{th} data block. By using the right data block, it is expected to obtain more accurate estimates, compared to the single-block calibration that does not differentiate the block-dependent heterogeneity.

To account for the block dependency, we consider multiple likelihood functions, each associated with the b^{th} block of datasets, as follows:

$$\mathcal{L}_b(\boldsymbol{\theta}_b, \sigma^2 | y(\mathbf{x})) = \prod_{j \in \mathcal{I}_b} \mathcal{N}(\eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}), \sigma^2), \quad \forall b, \quad (7)$$

where \mathcal{I}_b is an index set in the b^{th} block of the dataset, associated with $\boldsymbol{\theta}_b$, and $\boldsymbol{\theta}_{-b}$ represents the remaining parameters. $\eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})$ is the computer model output at input \mathbf{x}_j when the b^{th} block parameters are at $\boldsymbol{\theta}_b$ and remaining parameters at $\boldsymbol{\theta}_{-b}$.

While \mathcal{I}_b does not have to be disjoint with other blocks, $\boldsymbol{\theta}_b$ and $\boldsymbol{\theta}_{-b}$ are disjoint. Then the corresponding log-likelihood function becomes

$$\ell_b(\boldsymbol{\theta}_b, \sigma^2 | y(\mathbf{x}), \boldsymbol{\theta}_{-b}) = -\frac{n_b}{2} \log 2\pi - \frac{n_b}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{j \in \mathcal{I}_b} (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}))^2, \quad \forall b, \quad (8)$$

where $n_b = |\mathcal{I}_b|$ is the size of \mathcal{I}_b . To maximize the log-likelihood, we consider the corresponding empirical loss function as

$$F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}) = \frac{1}{n_b} \sum_{j \in \mathcal{I}_b} (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}))^2, \quad \forall b, \quad (9)$$

and obtain the estimates for the b^{th} block of parameters $\boldsymbol{\theta}_b$ by

$$\boldsymbol{\theta}_b^* = \arg \min_{\boldsymbol{\theta}_b \in \Theta_b} F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}), \quad \forall b. \quad (10)$$

We refer to the proposed block-wise calibration approach as multi-block calibration (shortly, M-BC) in the subsequent discussion.

It should be noted that the simulation output $\eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})$ could possibly depend on $\boldsymbol{\theta}_{-b}$, as well as $\boldsymbol{\theta}_b$, when the b^{th} block of data overlaps with other blocks associated with some parameters in $\boldsymbol{\theta}_{-b}$. For instance, let us consider the cooling-season parameter $\boldsymbol{\theta}_c$. The simulation output during the cooling season (March–November) also depends on the setting of global parameters $\boldsymbol{\theta}_g$. Also, during March to April when the cooling and heating seasons overlap, heating-season parameters also affect the energy consumption output during a part of the cooling season. Therefore, in solving (10), the resulting $\boldsymbol{\theta}_b^*$ may also depend on $\boldsymbol{\theta}_{-b}$. Our ultimate goal is to find $\boldsymbol{\theta}_b^*$ that minimizes F_b , while at the same time, other block parameters minimize their own loss functions. In the next section, we provide an iterative algorithm and its convergence properties under certain conditions.

5. Parameter Estimation and Convergence Properties in Multi-block Calibration

In this section, we propose a new algorithm to calibrate parameters in a block-wise manner. We also provide convergence guarantees for the algorithm under certain conditions. Finally, given convergence, we employ the MLE's asymptotic properties and construct confidence intervals to quantify estimation uncertainties.

5.1. Multi-block Calibration Algorithm

Recall that existing studies typically pre-generate data and build emulators for $\eta(\cdot)$ using the generated data. This approach is not directly applicable to the multi-block parameter calibration problem. One may attempt to build B emulators, say, $\hat{\eta}_1(\mathbf{x}, \boldsymbol{\theta}_1; \boldsymbol{\theta}_{-1}), \dots, \hat{\eta}_B(\mathbf{x}, \boldsymbol{\theta}_B; \boldsymbol{\theta}_{-B})$, using each block of data and try to calibrate $\boldsymbol{\theta}_b$ with $\hat{\eta}_b(\mathbf{x}, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})$. However, as discussed earlier, the computer model response at the b^{th} block can be affected by the setting of $\boldsymbol{\theta}_{-b}$, making $\hat{\eta}_b(\cdot)$ a function of both $\boldsymbol{\theta}_b$ and $\boldsymbol{\theta}_{-b}$. If one calibrates $\boldsymbol{\theta}_b$ and $\boldsymbol{\theta}_{-b}$ simultaneously using $\hat{\eta}_b(\mathbf{x}, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})$, the resulting calibrated values might differ across multiple blocks. Further, instead of generating data from a computer model at pre-defined design points, we want to generate computer model outputs more effectively so more informative computer model data can be used for the calibration purpose and directly use the data to calibrate parameters without building emulators.

To this end, we design a new algorithm to solve the block-wise calibration problem. The fundamental idea is to devise a block-wise optimization algorithm so that the parameters in different blocks can be calibrated with their own objective functions over iterations. Specifically, we propose cyclically optimizing one block of parameters each time while keeping the parameters in other blocks fixed at their most up-to-date values. For instance, in the BEM calibration, we have $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_g$, $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_c$, and $\boldsymbol{\theta}_3 = \boldsymbol{\theta}_h$ with $B = 3$. We iteratively optimize $\boldsymbol{\theta}_g$ that minimizes $F_g(\boldsymbol{\theta}_g; \boldsymbol{\theta}_{-g})$, while other parameters in $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_h$ are fixed at their current iterates. Then we optimize $\boldsymbol{\theta}_c$ for $F_c(\boldsymbol{\theta}_c; \boldsymbol{\theta}_{-c})$ with the previously optimized $\boldsymbol{\theta}_g$ and continue with a similar procedure for $F_h(\boldsymbol{\theta}_h; \boldsymbol{\theta}_{-h})$ with the previously optimized $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_c$. This procedure is repeated until pre-specified stopping criteria are satisfied.

Let $\boldsymbol{\theta}_b^k$ denote the iterate of $\boldsymbol{\theta}_b$ after finishing the k^{th} iteration for $b = 1, 2, \dots, B$. At the $(k+1)^{\text{th}}$ iteration, let $F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})$ in (9) denote the loss function for the b^{th} block of parameters $\boldsymbol{\theta}_b$, given the remaining parameter iterates $\boldsymbol{\theta}_{-b}^{k+1}$. Without loss of generality, we optimize each block in an ascending order, starting from the first block, i.e., $b = 1$. Therefore, in optimizing $\boldsymbol{\theta}_b$ that minimizes $F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})$, the remaining parameters become $\boldsymbol{\theta}_{-b}^{k+1} = [\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_{b-1}^{k+1}, \boldsymbol{\theta}_{b+1}^k, \dots, \boldsymbol{\theta}_B^k]$. Here, note that the parameters before the b^{th} block are previously updated, whereas those after the b^{th} block are not refined yet. In this scheme, the loss function for the b^{th} block in (9) at the $(k+1)^{\text{th}}$ iteration can be written as

$$F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1}) = \frac{1}{n_b} \sum_{j \in \mathcal{I}_b} (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1}))^2. \quad (11)$$

The computer model output $\eta(\cdot)$ is likely nonlinear, as most computer models consist of a set of complex mathematical functions. Hence, we consider nonlinear optimization methods to find $\boldsymbol{\theta}_b$ that minimizes $F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})$, given $\boldsymbol{\theta}_{-b}^{k+1}$. Among various optimization methods, we use the most

commonly used method, which is GD. Our approach can be easily extended to other second-order optimization methods such as Newton's method or quasi-Newton methods.

In minimizing $F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})$ over $\boldsymbol{\theta}_b$, we further consider the inner iteration, given $\boldsymbol{\theta}_{-b}^{k+1}$. Here, the inner iteration is for the GD-based parameter updates in each block, and the outer iteration implies the cycle for all blocks. Let $\boldsymbol{\theta}_b^{k,m}$ denote the iterate of $\boldsymbol{\theta}_b$ at the m^{th} inner iteration of the outer $(k+1)^{\text{th}}$ iteration. We update $\boldsymbol{\theta}_b^{k,m}$ by

$$\boldsymbol{\theta}_b^{k,m+1} = \boldsymbol{\theta}_b^{k,m} - \alpha_b^{k,m} \nabla F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1}), \quad (12)$$

where $\alpha_b^{k,m} > 0$ is the step size and the i^{th} element of $\nabla F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1})$ is

$$[\nabla F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1})]_{b,i} = \left. \frac{\partial F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})}{\partial \theta_{b,i}} \right|_{\theta_{b,i} = \theta_{b,i}^{k,m}}, \quad (13)$$

where $\theta_{b,i}$ is the i^{th} parameter in $\boldsymbol{\theta}_b$ for $i = 1, 2, \dots, p_b$, with p_b being the number of parameters in $\boldsymbol{\theta}_b$ for $b = 1, 2, \dots, B$. Since the loss functions have no mathematical closed-form expressions due to the black box nature of the simulator, similar to (6), we use a finite-differencing to approximate the gradient for each loss function as follows:

$$\left. \frac{\partial F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})}{\partial \theta_{b,i}} \right|_{\theta_{b,i} = \theta_{b,i}^{k,m}} \approx \frac{F_b(\boldsymbol{\theta}_b^{k,m} + h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}^{k+1}) - F_b(\boldsymbol{\theta}_b^{k,m} - h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}^{k+1})}{2h}, \quad (14)$$

where $h > 0$ is a small perturbation to the i^{th} parameter of each $\boldsymbol{\theta}_b$ and $\mathbf{e}_{b,i}$ is a $p_b \times 1$ vector with its i^{th} element being one and others zero.

We repeat the inner iteration for each block until some termination criteria are met. Once we obtain $\boldsymbol{\theta}_b$ that minimizes the b^{th} loss function $F_b(\boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b}^{k+1})$, denoted by $\boldsymbol{\theta}_b^{k+1}$, we move to the next $(b+1)^{\text{th}}$ block to minimize its loss function $F_{b+1}(\boldsymbol{\theta}_{b+1}; \boldsymbol{\theta}_{-(b+1)}^{k+1})$ over $\boldsymbol{\theta}_{b+1}$, given $\boldsymbol{\theta}_{-(b+1)}^{k+1} = [\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_b^{k+1}, \boldsymbol{\theta}_{b+2}^k, \dots, \boldsymbol{\theta}_B^k]$. After we finish the cycle with all blocks, we set $k = k+1$ and repeat the procedure. We finish the iteration when either the norm of gradient of each loss function is less than a small tolerance τ_1 , the relative difference of loss function values is less than a small tolerance τ_2 , or the maximum number of iteration is larger than some value τ_3 . Algorithm 1 summarizes the proposed M-BC procedure.

We would like to highlight that the proposed cyclic calibration approach is different from the block coordinate descent (BCD) algorithm. The major difference is that our approach handles multiple objective functions, each of which is associated with its corresponding subset of the dataset. On the other hand, BCD considers a single objective function.

As discussed earlier, an important feature of our approach is that it adaptively generates computer model outputs during the calibration procedure, unlike existing approaches that use pre-collected samples. As the optimization proceeds, newer parameters that make the computer model

Algorithm 1 Multi-block Calibration (M-BC)

-
- 1: **Input:** $\mathcal{D} = \{y(\mathbf{x}_j), \mathbf{x}_j\}_{j=1}^n$ with an index set \mathcal{I}_b for each b^{th} block of \mathcal{D} , $b = 1, 2, \dots, B$
 - 2: **Initialization:** $\boldsymbol{\theta}^0 = (\boldsymbol{\theta}_1^0, \boldsymbol{\theta}_2^0, \dots, \boldsymbol{\theta}_B^0)$ and $k \leftarrow 0$
 - 3: **while** convergence criterion not met **do**
 - 4: **for** $b \leftarrow 1$ to B **do**
 - 5: Set $m \leftarrow 1$ and $\boldsymbol{\theta}_b^{k,m} \leftarrow \boldsymbol{\theta}_b^k$
 - 6: **while** convergence criterion not met **do**
 - 7: Run simulations for $\mathbf{x}_j, \forall j \in \mathcal{I}_b$, at $(\boldsymbol{\theta}_b^{k,m}, \boldsymbol{\theta}_{-b}^{k+1})$ to obtain $F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1})$
 - 8: Run simulations for $\mathbf{x}_j, \forall j \in \mathcal{I}_b$, at $(\boldsymbol{\theta}_b^{k,m} + h\mathbf{e}_{b,i}, \boldsymbol{\theta}_{-b}^{k+1})$ and $(\boldsymbol{\theta}_b^{k,m} - h\mathbf{e}_{b,i}, \boldsymbol{\theta}_{-b}^{k+1})$ to obtain $F_b(\boldsymbol{\theta}_b^{k,m} + h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}^{k+1})$ and $F_b(\boldsymbol{\theta}_b^{k,m} - h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}^{k+1})$, respectively
 - 9: Calculate $G_b^{k,m} = \nabla F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1})$ using (13)-(14) and set a step size $\alpha_b^{k,m}$
 - 10: Update $\boldsymbol{\theta}_b^{k,m+1} \leftarrow \boldsymbol{\theta}_b^{k,m} - \alpha_b^{k,m} G_b^{k,m}$
 - 11: Set $m \leftarrow m + 1$
 - 12: **end while**
 - 13: Set $\boldsymbol{\theta}_b^{k+1} = \boldsymbol{\theta}_b^{k,m}$
 - 14: **end for**
 - 15: Set $k \leftarrow k + 1$
 - 16: **end while**
 - 17: **Output:** $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2^k, \dots, \boldsymbol{\theta}_B^k)$
-

outputs closer to physical process responses are obtained. That is, once $\boldsymbol{\theta}_b^{k,m}$ is updated to $\boldsymbol{\theta}_b^{k,m+1}$ through the GD procedure in (12), we run the simulator at the refined $\boldsymbol{\theta}_b^{k,m+1}$ to get data that leads to smaller loss. Through this iterative data generation procedure, more informative data can be produced toward minimizing the discrepancy between the computer model output and physical process response.

As a remark, in multi-block calibration one can randomly choose the next block, but in this section we consider the fixed ordering that proceeds in a cyclical way. In Section 6, we empirically compare the cyclic ordering with stochastic ordering. Additionally, we note that, depending on the problem structure, the order of blocks may matter. However, in our numerical examples in a wide range of settings, we obtain comparable results in different orderings.

5.2. Convergence Analysis

This section presents the convergence properties when each loss function F_b , $b = 1, 2, \dots, B$, is a general nonconvex, strongly convex, or convex function. While we utilize the standard analysis procedure of GD, the block-wise treatment with multiple loss functions poses substantial challenges in analyzing the convergence properties of the proposed M-BC approach. In each of the three settings, we impose conditions on F_b and possibly the iterates generated by Algorithm 1 to achieve

certain convergence properties. In the special case where the number of blocks is equal to 1, our results recover the standard results for GD. The detailed proofs of all lemma and theorems are available in the online supplementary document.

Throughout the convergence analysis, we use the notations summarized in Table 2. We understand that the notations are complex, mainly due to the double (inner and outer) iteration nature in M-BC. For ease of understanding, Figure 2 presents the essence of the notations. For the number of inner iterations, denoted by $M_b(k)$, we consider the fixed number of inner iterations, i.e., $M_b(k) = M_b$, but the analysis can be extended to variable number of inner iterations. Moreover, we use a fixed step size α_b for the b^{th} block for $b = 1, 2, \dots, B$, but this could also be extended to utilizing a backtracking line search.

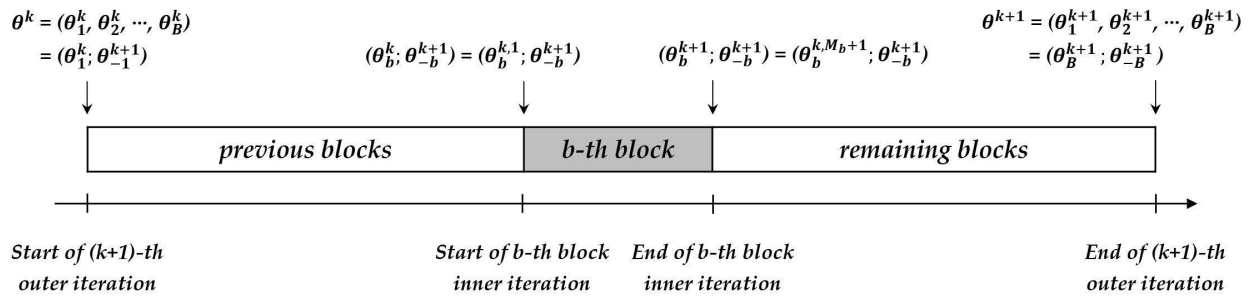


Figure 2 Illustration of notations at the $(k+1)^{\text{th}}$ outer iteration

In our analysis, we make the following assumption.

ASSUMPTION 1. Let F_b be a continuously differentiable function in $\text{dom}(F_b)$ for $b = 1, 2, \dots, B$. Assume that F_b has a block-wise Lipschitz continuous gradient with a Lipschitz constant $L_b < \infty$, with respect to θ_b for any given θ_{-b} , i.e., $\|\nabla F_b(\theta_b; \theta_{-b}) - \nabla F_b(\theta'_b; \theta_{-b})\|_2 \leq L_b \|\theta_b - \theta'_b\|_2, \forall \theta_b, \theta'_b \in \mathbb{R}^{p_b}$. Further, assume that F_b is bounded below so that the infimum of F_b, F_b^{inf} , exists and F_b has at least one stationary point for all b .

We first show a descent lemma, a fundamental ingredient for the convergence analysis. Lemma 1 shows that the function value $F_b(\theta)$ decreases when the block parameter is updated from θ_b^k to θ_b^{k+1} with θ_{-b}^{k+1} fixed.

LEMMA 1 (**Descent lemma**). Suppose Assumption 1 holds with constants L_b for $b = 1, 2, \dots, B$. Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all $k \geq 0$ and all b ,

$$F_b(\theta_b^{k+1}; \theta_{-b}^{k+1}) \leq F_b(\theta_b^k; \theta_{-b}^{k+1}) - \frac{1}{2} \alpha_b \sum_{m=1}^{M_b} \|\nabla F_b(\theta_b^{k,m}; \theta_{-b}^{k+1})\|_2^2. \quad (15)$$

Table 2 Nomenclature for convergence analysis

Category	Symbol	Meaning
Superscript	k	The iteration number for the outer iteration.
	m	The iteration number for the inner iteration, i.e., $m = 1, 2, \dots, M_b(k)$.
	$M_b(k)$	The number of inner iterations for the b^{th} block at the k^{th} outer iteration.
	M_b	Fixed number of $M_b(k)$ regardless of the k^{th} outer iteration.
Iterate	$\boldsymbol{\theta}^k \in \mathbb{R}^p$	The iterate after all blocks are updated at the k^{th} outer iteration, i.e., $\boldsymbol{\theta}^k \equiv (\boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2^k, \dots, \boldsymbol{\theta}_B^k)$.
	$\boldsymbol{\theta}^* \in \mathbb{R}^p$	The unique global minimizer of F_b in the strongly convex case; a point at which the minimum of F_b is attained in the convex case, i.e., $\boldsymbol{\theta}^* \equiv (\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \dots, \boldsymbol{\theta}_B^*)$.
	$\boldsymbol{\theta}_b^k \in \mathbb{R}^{p_b}$	The iterate of the b^{th} block parameters after finishing the inner iteration of the b^{th} block at the k^{th} outer iteration.
	$\boldsymbol{\theta}_b^* \in \mathbb{R}^{p_b}$	The unique global minimizer of F_b given $\boldsymbol{\theta}_{-b}^*$ in the strongly convex case; a point at which the minimum of F_b given $\boldsymbol{\theta}_{-b}^*$ is attained in the convex case, i.e., $\boldsymbol{\theta}^* \equiv (\boldsymbol{\theta}_b^*; \boldsymbol{\theta}_{-b}^*)$.
	$\boldsymbol{\theta}_b^{k,m} \in \mathbb{R}^{p_b}$	The iterate of the b^{th} block parameter at the m^{th} inner iteration of the $(k+1)^{\text{th}}$ outer iteration, given $\boldsymbol{\theta}_{-b}^{k+1}$. Note $\boldsymbol{\theta}_b^{k,1} \equiv \boldsymbol{\theta}_b^k$ and $\boldsymbol{\theta}_b^{k,M_b+1} \equiv \boldsymbol{\theta}_b^{k+1}$.
	$\boldsymbol{\theta}_b^{k,*} \in \mathbb{R}^{p_b}$	The unique global minimizer of F_b given $\boldsymbol{\theta}_{-b}^{k+1}$ in the strongly convex case; a point at which the minimum of F_b given $\boldsymbol{\theta}_{-b}^{k+1}$ is attained in the convex case.
	$\boldsymbol{\theta}_{-b}^k \in \mathbb{R}^{p-p_b}$	$\boldsymbol{\theta}_{-b}^k \equiv [\boldsymbol{\theta}_1^k, \dots, \boldsymbol{\theta}_{b-1}^k, \boldsymbol{\theta}_{b+1}^{k-1}, \dots, \boldsymbol{\theta}_B^{k-1}]$.
	$(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}^p$	$(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) \equiv (\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_{b-1}^{k+1}, \boldsymbol{\theta}_b^k, \boldsymbol{\theta}_{b+1}^k, \dots, \boldsymbol{\theta}_B^k)$. Note $(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) = (\boldsymbol{\theta}_b^{k,1}; \boldsymbol{\theta}_{-b}^{k+1})$, where $\boldsymbol{\theta}_b^{k,1}$ is the initial value of $\boldsymbol{\theta}_b^k$ at the $(k+1)^{\text{th}}$ outer iteration with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed.
	$(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}^p$	$(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) \equiv (\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_{b-1}^{k+1}, \boldsymbol{\theta}_b^{k+1}, \boldsymbol{\theta}_{b+1}^k, \dots, \boldsymbol{\theta}_B^k)$ with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed. Note $(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) = (\boldsymbol{\theta}_b^{k,M_b+1}; \boldsymbol{\theta}_{-b}^{k+1})$, where $\boldsymbol{\theta}_b^{k,M_b+1}$ is the final value of $\boldsymbol{\theta}_b^k$ at the $(k+1)^{\text{th}}$ outer iteration with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed.
	$(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}^p$	$(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_{-b}^{k+1}) \equiv (\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_{b-1}^{k+1}, \boldsymbol{\theta}_b^{k,m}, \boldsymbol{\theta}_{b+1}^k, \dots, \boldsymbol{\theta}_B^k)$ with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed.
$(\boldsymbol{\theta}_b^{k,*}; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}^p$	$(\boldsymbol{\theta}_b^{k,*}; \boldsymbol{\theta}_{-b}^{k+1}) \equiv (\boldsymbol{\theta}_1^{k+1}, \dots, \boldsymbol{\theta}_{b-1}^{k+1}, \boldsymbol{\theta}_b^{k,*}, \boldsymbol{\theta}_{b+1}^k, \dots, \boldsymbol{\theta}_B^k)$ with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed.	
Function	$F_b(\boldsymbol{\theta}^k) \in \mathbb{R}$	The function value of F_b evaluated at $\boldsymbol{\theta}^k = (\boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2^k, \dots, \boldsymbol{\theta}_B^k)$.
	$F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}$	The function value of F_b evaluated at $(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1})$ to stress $\boldsymbol{\theta}_{-b}^{k+1}$ is fixed.
	F_b^{inf}	A lower bound on the function value of F_b .
Gradient	$\nabla F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) \in \mathbb{R}^{p_b}$	The gradient of F_b evaluated at $\boldsymbol{\theta}_b^k$ with $\boldsymbol{\theta}_{-b}^{k+1}$ being fixed.

The result above is analogous to that of GD. Namely, the progress made over a sequence of iterations (for M-BC, this is over the inner iterations) is proportional to the sum of the norms of the gradient squared. Using Lemma 1, we first analyze the convergence properties of M-BC on the general nonconvex case in Theorem 1.

Theorem 1 shows that under certain conditions, the iterate $\boldsymbol{\theta}^k$ converges to a stationary point $\boldsymbol{\theta}^*$ as $k \rightarrow \infty$, i.e., the norm of the gradient of all objective functions vanishes in the limit.

THEOREM 1 (General nonconvex case). *Suppose Assumption 1 holds. Further, suppose that*

$$\begin{aligned} F_b(\boldsymbol{\theta}^k) &\geq F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}), \\ F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) &\geq F_b(\boldsymbol{\theta}^{k+1}), \end{aligned} \tag{16}$$

for all $k \geq 0$ and $b = 1, 2, \dots, B$. Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all b ,

$$\lim_{k \rightarrow \infty} \|\nabla F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1})\|_2 = 0. \tag{17}$$

We make a few remarks about the conditions in (16). From Lemma 1, it follows that $F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) \geq F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1})$. Thus, the inequalities (16) imply that $F_b(\boldsymbol{\theta}^k) \geq F_b(\boldsymbol{\theta}^{k+1})$, i.e., the function value for the b^{th} block decreases after all blocks have updated their associated parameters (across outer iterations). This seems to be a strong assumption in that the blocks do not compete over the course of optimization. However, this does not require that when every other block updates, the function value F_b of the b^{th} block is non-increasing. Instead, F_b is non-increasing across outer iterations.

The result in Theorem 1 has an important implication. When we eventually find the stationary point $\boldsymbol{\theta}_b^*$ for the parameters in the b^{th} block, other blocks' parameters also converge to a stationary point $\boldsymbol{\theta}_{-b}^*$ as well and $F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^k)$ eventually converges to $F_b(\boldsymbol{\theta}_b^*, \boldsymbol{\theta}_{-b}^*)$, $\forall b$. It implies that even though we calibrate parameters in a block-wise manner, our approach obtains a concurrent convergence of the whole parameter vector, that is, $\boldsymbol{\theta}^k = (\boldsymbol{\theta}_1^k, \boldsymbol{\theta}_2^k, \dots, \boldsymbol{\theta}_B^k)$ converges to a stationary point $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \dots, \boldsymbol{\theta}_B^*)$ under the non-competing condition in (16) as k increases using M-BC. Accordingly, if F_b 's are strongly convex, F_b 's have a unique global minimizer $\boldsymbol{\theta}^*$. If F_b 's are convex, F_b 's have a global minimizer $\boldsymbol{\theta}^*$, $\forall b$.

We make the possible extension of conditions in (16) at the end of this section. Additionally, we relax the conditions in (16) and prove the similar result in Corollary 1 by instead assuming the weaker technical condition in (18).

COROLLARY 1. *Suppose Assumption 1 holds. Further, suppose that the infinitely cumulative sum of the decreases of the function values during the inner iterations is finite, i.e.,*

$$\lim_{K \rightarrow \infty} \sum_{k=0}^K \{F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}) - F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1})\} < \infty, \tag{18}$$

for all $b = 1, 2, \dots, B$. Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all b ,

$$\lim_{k \rightarrow \infty} \|\nabla F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1})\|_2 = 0. \tag{19}$$

Corollary 1 states that as long as the sum of loss function decreases is finite, M-BC converges to a stationary point. It allows other block calibrations to increase the b^{th} block loss, so the non-competing condition in (16) is not needed. Thus, this result is more general than that in Theorem 1.

Next, we show the convergence rates for the strongly convex (Theorem 2) and convex (Theorem 3) cases.

THEOREM 2 (Strongly convex case). *Suppose Assumption 1 holds. Further, assume that the function F_b is μ_b -strongly convex over θ_b for any fixed θ_{-b} for $b = 1, 2, \dots, B$. Moreover, suppose that*

$$\begin{aligned} |F_b(\theta^{k+1}) - F_b(\theta^*)| &\leq c_1 \{F_b(\theta_b^{k+1}; \theta_{-b}^{k+1}) - F_b(\theta_b^{k,*}; \theta_{-b}^{k+1})\}, \\ F_b(\theta_b^k; \theta_{-b}^{k+1}) - F_b(\theta_b^{k,*}; \theta_{-b}^{k+1}) &\leq c_2 |F_b(\theta^k) - F_b(\theta^*)|, \end{aligned} \quad (20)$$

for all $k \geq 0$ and b , where $c_1, c_2 > 0$ and $c_1 c_2 < \frac{1}{(1 - \alpha_b \mu_b)^{M_b}}$. Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all k and b ,

$$|F_b(\theta^k) - F_b(\theta^*)| \leq \rho^k |F_b(\theta^0) - F_b(\theta^*)|, \quad (21)$$

where $\rho = c_1 c_2 (1 - \alpha_b \mu_b)^{M_b} < 1$.

Both conditions in (20) bound the optimality gap between the progress made in inner iterations versus the progress made in outer iterations. Since the objective function of each block is changing across outer iterations, the absolute value safeguards against the possibility of a current iterate having lower function value than the optimal solution. We note that this is possible in our setting since the optimal solution of $\theta^* = (\theta_b^*; \theta_{-b}^*)$ is defined as the point where all objectives are optimized and thus can potentially have a higher function value than the point $(\theta_b^{k,*}; \theta_{-b}^k)$ has.

Note that the conditions for Theorem 2 do not require that the function of the b^{th} block is monotonically non-increasing across outer iterations. That being said, these assumptions are relatively weak and can be tightened so that the result still shows the linear convergence rate at possibly a slower rate. We provide the corollary about this.

COROLLARY 2. *Suppose Assumption 1 holds. Further, assume that the function F_b is μ_b -strongly convex over θ_b for any fixed θ_{-b} for $b = 1, 2, \dots, B$. Moreover, suppose that*

$$\begin{aligned} |F_b(\theta^{k+1}) - F_b(\theta^*)| &\leq c_1 \{F_b(\theta_b^{k+1}; \theta_{-b}^{k+1}) - F_b(\theta_b^{k,*}; \theta_{-b}^{k+1})\}, \\ F_b(\theta_b^k; \theta_{-b}^{k+1}) - F_b(\theta_b^{k,*}; \theta_{-b}^{k+1}) &\leq c_2 |F_b(\theta^k) - F_b(\theta^*)|, \end{aligned} \quad (22)$$

for all $k \geq 0$ and b , where $c_1, c_2 > 0$ and $c_1 c_2 < \frac{1}{(1 - \alpha_b \mu_b)^{M_b}}$. Additionally, suppose that

$$\begin{aligned} F_b(\theta^k) &\geq F_b(\theta_b^k; \theta_{-b}^{k+1}), \\ F_b(\theta_b^{k+1}; \theta_{-b}^{k+1}) &\geq F_b(\theta^{k+1}), \end{aligned} \quad (23)$$

for all k and b . Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all k and b ,

$$F_b(\theta^k) - F_b(\theta^*) \leq \rho^k \{F_b(\theta^0) - F_b(\theta^*)\}, \quad (24)$$

where $\rho = c_1 c_2 (1 - \alpha_b \mu_b)^{M_b} < 1$.

Theorem 2 and Corollary 2 show linear convergence of the iterates generated by the M-BC algorithm, analogous to that of GD (in the special case where $b = 1$, we recover the result of GD). While convergence is derived across outer iterations, the rate of convergence depends on the number of inner iterations. Note that $\rho = c_1 c_2 (1 - \alpha_b \mu_b)^{M_b} < 1$, thus, the larger the number of inner iterations, the faster the rate (constant), or alternatively, the larger the errors that are acceptable in (22) (larger constants c_1 and c_2). Note that compared to Theorem 2, Corollary 2 has somewhat stronger result that shows outer function value $F_b(\boldsymbol{\theta}^k)$ keep non-increasing with the linear rate $\mathcal{O}(\rho^k)$.

Additionally, Theorem 3 shows the convergence rate for the convex case with certain assumptions.

THEOREM 3 (Convex case). *Suppose Assumption 1 holds. Further, assume that the function F_b is convex over $\boldsymbol{\theta}_b$ for any fixed $\boldsymbol{\theta}_{-b}$ for $b = 1, 2, \dots, B$. Moreover, suppose that*

$$\begin{aligned} |F_b(\boldsymbol{\theta}^{k+1}) - F_b(\boldsymbol{\theta}^*)| &\leq c_1 \{F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) - F_b(\boldsymbol{\theta}_b^{k,*}; \boldsymbol{\theta}_{-b}^{k+1})\}, \\ \|\boldsymbol{\theta}_b^{k+1} - \boldsymbol{\theta}_b^{k+1,*}\|_2^2 &\leq \|\boldsymbol{\theta}_b^{k+1} - \boldsymbol{\theta}_b^{k,*}\|_2^2, \end{aligned} \quad (25)$$

for all $k \geq 0$ and b , where $c_1 > 0$. Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all $k \geq 1$ and b ,

$$\frac{1}{k} \sum_{l=1}^k |F_b(\boldsymbol{\theta}^l) - F_b(\boldsymbol{\theta}^*)| \leq \frac{c_1 \|\boldsymbol{\theta}_b^0 - \boldsymbol{\theta}_b^{0,*}\|_2^2}{2\alpha_b k M_b}. \quad (26)$$

The first inequality in (25) is the same as the first condition in the strongly convex setting (see (20)). The second bounds the distance between the difference from the current iterate's function value to the optimal solution's function value across two outer iterations.

Note that the first assumption in (25) is relatively weak and can be tightened so that the result still shows the sublinear convergence rate at possibly a slower rate. We provide the result about this in Corollary 3.

COROLLARY 3. *Suppose Assumption 1 holds. Further, assume that the function F_b is convex for $b = 1, 2, \dots, B$ over $\boldsymbol{\theta}_b$ for any fixed $\boldsymbol{\theta}_{-b}$. Moreover, assume that*

$$\begin{aligned} |F_b(\boldsymbol{\theta}^{k+1}) - F_b(\boldsymbol{\theta}^*)| &\leq c_1 \{F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) - F_b(\boldsymbol{\theta}_b^{k,*}; \boldsymbol{\theta}_{-b}^{k+1})\}, \\ \|\boldsymbol{\theta}_b^{k+1} - \boldsymbol{\theta}_b^{k+1,*}\|_2^2 &\leq \|\boldsymbol{\theta}_b^{k+1} - \boldsymbol{\theta}_b^{k,*}\|_2^2, \end{aligned} \quad (27)$$

for all $k \geq 0$ and b , where $c_1 > 0$. Additionally, assume that

$$\begin{aligned} F_b(\boldsymbol{\theta}^k) &\geq F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}), \\ F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) &\geq F_b(\boldsymbol{\theta}^{k+1}), \end{aligned} \quad (28)$$

for all k and b . Let $\alpha_b \in (0, 1/L_b]$ for all b . Then, for all k and b ,

$$F_b(\boldsymbol{\theta}^k) - F_b(\boldsymbol{\theta}^*) \leq \frac{c_1 \|\boldsymbol{\theta}_b^0 - \boldsymbol{\theta}_b^{0,*}\|_2^2}{2\alpha_b k M_b}. \quad (29)$$

Theorem 3 and Corollary 3 show that in the convex setting, the iterates generated by the M-BC algorithm converge to a minimizer at a sublinear rate that depends on both the number of inner and outer iterations. Again, this is analogous to GD, and in the special case with $b = 1$, we recover the result of GD. Note that Corollary 3 has somewhat stronger result that shows outer iterates $\boldsymbol{\theta}^k$ keep non-increasing with the sublinear rate $\mathcal{O}(1/(kM_b))$.

We make a comment about the number of iterations required for our algorithm to converge (i.e., number of inner iterations for a given block and number of outer iterations) and provide the computational complexity of M-BC. Of course, the number of iterations highly depends on the convergence criterion used in Algorithm 1. To give a concrete example, consider the case of strongly convex functions, where the convergence criterion in the inner loop is based on the optimality gap and suppose the inner and outer termination tolerances for relative differences are set as $0 < \epsilon_{\text{outer}} \leq \epsilon_{\text{inner}} < 1$. In this case, one can show that the number of inner iterations required for each block is $\mathcal{O}(\log(1/\epsilon_{\text{inner}}))$, and the total number of iterations is $\mathcal{O}(\log_{\epsilon_{\text{inner}}}(1/\epsilon_{\text{outer}}))$. Thus, combining the time for data generation and for parameter estimation, the computational complexity becomes $\mathcal{O}(n \log_{\epsilon_{\text{inner}}}(1/\epsilon_{\text{outer}}))$, assuming that the simulation running time is proportional to the data size n . Similar results can be shown for the convex and nonconvex cases under appropriate termination conditions.

Finally, we would like to make additional remarks about the conditions in our analysis. While the conditions we impose in deriving convergence properties, for example, that the objective functions do not compete for all $\boldsymbol{\theta}_b$, may seem to be strong, we make these assumptions for the analysis of the M-BC algorithm to be analogous to that of GD. Furthermore, the conditions in our analysis are less restrictive than that as they are over the course of the iterates generated by the algorithm, and not for the entire space. These conditions may not be verified except in relatively simple settings, as they require information about the optimal solution and/or problem specific parameters. In practice, after running the method on a specific problem, one can observe the trajectories to see if the conditions are satisfied. In our numerical examples in a wide range of settings and the BEM case study, we did not observe the situation that the non-competing conditions were severely violated.

That being said, we conclude this section by discussing possible extensions of the proposed approach and convergence results. Suppose that in the general nonconvex case, instead of assuming (16), we can relax the conditions as

$$\begin{aligned} F_b(\boldsymbol{\theta}^k) + \delta_{b,1}^k &\geq F_b(\boldsymbol{\theta}_b^k; \boldsymbol{\theta}_{-b}^{k+1}), \\ F_b(\boldsymbol{\theta}_b^{k+1}; \boldsymbol{\theta}_{-b}^{k+1}) &\geq F_b(\boldsymbol{\theta}^{k+1}) - \delta_{b,2}^k, \end{aligned} \tag{30}$$

for all $k \geq 0$ and $b = 1, 2, \dots, B$ with $\delta_{b,1}^k > 0$ and $\delta_{b,2}^k > 0$. That is, we allow the objective function of the block to increase by some levels ($\delta_{b,1}^k$ and $\delta_{b,2}^k$) after calibrating parameters in other blocks.

Under this assumption, if $\delta_{b,1}^k = \delta_1$ and $\delta_{b,1}^k = \delta_2$ for all $b = 1, 2, \dots, B$, then we conjecture that one can only show convergence to a neighborhood of the solution that depends on δ_1 and δ_2 . On the other hand, if $\delta_{b,1}^k$ and $\delta_{b,1}^k$ are summable, then a similar result to Theorem 1 can be proven. Similarly, the conditions for the results in the strongly convex and convex cases can be relaxed, and under appropriate conditions, convergence (and rates) can be proven. Our BEM case study in Section 7 demonstrates that the non-competing condition in (16) is satisfied. However, we plan to further investigate competing cases with (30) in our future study, as it may arise in other applications.

5.3. Asymptotic Properties of Calibration Parameters

When θ_b converges to a minimizer θ_b^* for each b based on the results in the previous section, we obtain the MLE for the calibration parameters. Thus, we can entertain its appealing theoretical properties such as consistency and normality. From the likelihood function in (8), the consistency property says $\hat{\theta}_{b,\text{MLE}}$ converges in probability to a true parameter vector, denoted by $\theta_{b,0}$, as $n_b \rightarrow \infty$, i.e., $\hat{\theta}_{b,\text{MLE}} \xrightarrow{p} \theta_{b,0}$ for each b . Further, the asymptotic normality indicates that the estimator $\sqrt{n_b}(\hat{\theta}_{b,\text{MLE}} - \theta_{b,0})$ converges in distribution to a (multivariate) normal distribution $\mathcal{N}(\mathbf{0}, I(\theta_{b,0})^{-1})$ as $n_b \rightarrow \infty$, i.e., $\sqrt{n_b}(\hat{\theta}_{b,\text{MLE}} - \theta_{b,0}) \xrightarrow{d} \mathcal{N}(\mathbf{0}, I(\theta_{b,0})^{-1})$, where $I(\theta_{b,0})$ is an expected Fisher information matrix. We use the MLE's asymptotic properties to construct the confidence intervals of calibration parameters for uncertainty quantification.

Based on (8), the expected Fisher information matrix is represented by

$$I(\theta_{b,0}) = -\mathbb{E} \left[\frac{\partial^2 \tilde{\ell}_b(\theta_b | y(\mathbf{x}), \theta_{-b})}{\partial \theta_b^2} \right] \Bigg|_{\theta_b = \theta_{b,0}}, \quad (31)$$

where $\tilde{\ell}_b(\theta_b | y(\mathbf{x}), \theta_{-b})$ is a log-likelihood function from a single observation for the b^{th} block as

$$\tilde{\ell}_b(\theta_b | y(\mathbf{x}), \theta_{-b}) = -\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y(\mathbf{x}) - \eta(\mathbf{x}, \theta_b; \theta_{-b}))^2. \quad (32)$$

The expected Fisher information matrix can be approximated by its empirical counterpart,

$$I(\theta_{b,0}) \approx -\frac{1}{n_b} \sum_{j \in \mathcal{I}_b} \frac{\partial^2 \tilde{\ell}_b(\theta_b | y(\mathbf{x}_j), \theta_{-b})}{\partial \theta_b^2} \Bigg|_{\theta_b = \theta_{b,0}}. \quad (33)$$

Using the chain rule, we obtain the (i, i') -entry of $\partial^2 \tilde{\ell}_b(\theta_b | y(\mathbf{x}_j), \theta_{-b}) / \partial \theta_b^2$ as follows.

$$\frac{\partial^2 \tilde{\ell}_b(\theta_b | y(\mathbf{x}_j), \theta_{-b})}{\partial \theta_{b,i} \partial \theta_{b,i'}} = -\frac{1}{\sigma^2} \frac{\partial \eta(\mathbf{x}_j, \theta_b; \theta_{-b})}{\partial \theta_{b,i'}} \frac{\partial \eta(\mathbf{x}_j, \theta_b; \theta_{-b})}{\partial \theta_{b,i}} + \frac{1}{\sigma^2} (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \theta_b; \theta_{-b})) \times \frac{\partial^2 \eta(\mathbf{x}_j, \theta_b; \theta_{-b})}{\partial \theta_{b,i} \partial \theta_{b,i'}}. \quad (34)$$

Here, since the first- and second-order partial derivatives of $\eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})$ are not analytically available, we obtain them numerically using the central finite difference (Abramowitz and Stegun 1972) as follows:

$$\frac{\partial \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})}{\partial \theta_{b,i}} \approx \frac{\eta(\mathbf{x}_j, \boldsymbol{\theta}_b + h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b - h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b})}{2h}, \quad (35)$$

$$\begin{aligned} \frac{\partial^2 \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})}{\partial \theta_{b,i}^2} \approx & \frac{-\eta(\mathbf{x}_j, \boldsymbol{\theta}_b + 2h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}) + 16\eta(\mathbf{x}_j, \boldsymbol{\theta}_b + h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}) - 30\eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})}{12h^2} \\ & + \frac{16\eta(\mathbf{x}_j, \boldsymbol{\theta}_b - h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b}) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b - 2h\mathbf{e}_{b,i}; \boldsymbol{\theta}_{-b})}{12h^2}, \quad (36) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \eta(\mathbf{x}_j, \boldsymbol{\theta}_b; \boldsymbol{\theta}_{-b})}{\partial \theta_{b,i} \partial \theta_{b,i'}} \approx & \frac{\eta(\mathbf{x}_j, \boldsymbol{\theta}_b + h\mathbf{e}_{b,i} + h'\mathbf{e}_{b,i'}; \boldsymbol{\theta}_{-b}) - \eta(\mathbf{x}_j, \boldsymbol{\theta}_b + h\mathbf{e}_{b,i} - h'\mathbf{e}_{b,i'}; \boldsymbol{\theta}_{-b})}{4hh'} \\ & + \frac{-\eta(\mathbf{x}_j, \boldsymbol{\theta}_b - h\mathbf{e}_{b,i} + h'\mathbf{e}_{b,i'}; \boldsymbol{\theta}_{-b}) + \eta(\mathbf{x}_j, \boldsymbol{\theta}_b - h\mathbf{e}_{b,i} - h'\mathbf{e}_{b,i'}; \boldsymbol{\theta}_{-b})}{4hh'}, \quad (37) \end{aligned}$$

for $i, i' = 1, 2, \dots, p_b$, where $h, h' > 0$ are small perturbation values. Further, assuming the physical process data $y(\mathbf{x})$ have the same variance σ^2 across multiple blocks, we can find its MLE as

$$\hat{\sigma}_{\text{MLE}}^2 = \frac{1}{n} \sum_{j=1}^n (y(\mathbf{x}_j) - \eta(\mathbf{x}_j, \hat{\boldsymbol{\theta}}_{\text{MLE}}))^2. \quad (38)$$

Then we can obtain the asymptotic $100(1 - \alpha)\%$ Wald confidence interval (CI) for each component of $\hat{\boldsymbol{\theta}}_{b, \text{MLE}}$ as follows:

$$\hat{\theta}_{b, \text{MLE}, i} \pm z_{1-\alpha/2} \frac{1}{\sqrt{n_b}} \sqrt{I_{ii}^{-1}(\hat{\boldsymbol{\theta}}_{b, \text{MLE}})}, \quad i = 1, 2, \dots, p_b, \quad (39)$$

for $b = 1, 2, \dots, B$, $z_{1-\alpha/2}$ is a critical value of the normal distribution, and I_{ii}^{-1} denotes the i^{th} diagonal entry of the inverse of the Fisher information matrix I .

5.4. Implementation Details

We provide guidance on defining the algorithmic parameters in M-BC (Algorithm 1).

- $\alpha_b^{k,m} > 0$ (step size of the b^{th} block at the $(k+1)^{\text{th}}$ outer iteration and m^{th} inner iteration):
The step size $\alpha_b^{k,m}$ can be obtained by a backtracking line search, as described in Algorithm 2 below. In the backtracking line search, the parameters can be set as $\bar{\alpha} = 1$, $\rho = 1/2$, $c = 10^{-4}$ (or $1/2$) (Boyd and Vandenberghe 2004, Nocedal and Wright 2006). Note that, in principle, the step size $\alpha_b^{k,m}$ could also be set to a sufficiently small constant or a diminishing sequence (e.g., $1/m$).

Algorithm 2 Backtracking Line Search

- 1: **Choose:** $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$
 - 2: **while** $F_b(\boldsymbol{\theta}_b^{k,m} - \alpha G_b^{k,m}; \boldsymbol{\theta}_b^{k+1}) > F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_b^{k+1}) - c\alpha \|\nabla F_b(\boldsymbol{\theta}_b^{k,m}; \boldsymbol{\theta}_b^{k+1})\|_2^2$ **do**
 - 3: Update $\alpha \leftarrow \rho\alpha$
 - 4: **end while**
 - 5: **Output:** $\alpha_b^{k,m} \leftarrow \alpha$
-

- $h > 0$ (*bandwidth of finite-differencing when approximating the gradient*): The bandwidth h is set to 10^{-8} , which is the square root of machine precision. We note that in the setting where exact function values can be computed, the choice of bandwidth is optimal up to constants (Moré and Wild 2012). In practice, we note that a sufficiently small value h works well in most cases (e.g., 10^{-4}).
- $\tau_1 > 0$ (*termination tolerance: norm of the gradient*): We suggest that τ_1 would be set as any sufficiently small value (e.g., 10^{-4}).
- $\tau_2 > 0$ (*termination tolerance: relative difference of loss function values*): We suggest that τ_2 would be also set as any sufficiently small value (e.g., 10^{-4}).
- $\tau_3 > 0$ (*termination tolerance: maximum iterations*): The maximum number of iterations is used to safeguard the algorithm from running forever. If the algorithm reaches this value, it exits the loop. We suggest that τ_3 would be set as sufficiently large value as long as computing resources are available. In our implementation, we set 2000 for the outer loop and 1000 for the inner loops of the algorithm.

6. Numerical Study

In this section, we evaluate the calibration accuracy of our proposed method in comparison with other alternatives through numerical studies. We also report the half length of CI and coverage rate of each parameter to quantify estimation uncertainties. All experiments, except those for alternative methods in Section 6.4, are conducted with MATLAB 2014a. For the alternative methods in Section 6.4, R is used. All experiments use the following computing environment: 64-bit Windows OS with the Intel Xeon CPU E5-2697 @ 2.60 GHz processor and 128 GB RAM.

6.1. Problem Settings

We consider three numerical examples with different settings of the following physical processes and computer models.

(a) **Example I** (perfect computer model, block-wise convex loss functions)

- Physical process

$$y(x) = \mathbb{I}\left(x < \frac{2}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \cos(x) + \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \varepsilon.$$

- Computer model

$$\begin{aligned} \eta(x, \boldsymbol{\theta}) = & \mathbb{I}\left(x < \frac{2}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \cos(x) + \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) \\ & - \mathbb{I}\left(x < \frac{2}{5}\pi\right) \frac{(\theta_1 + 1)\sqrt{x^2 - x + 1}}{2} - \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{4}{5}\pi\right) \frac{(\theta_1 + 1)(\theta_2 - 10)\sqrt{x^2 - x + 1}}{10} \\ & - \mathbb{I}\left(\frac{4}{5}\pi \leq x < \frac{6}{5}\pi\right) \frac{(\theta_2 - 10)\sqrt{x^2 - x + 1}}{10} - \mathbb{I}\left(\frac{6}{5}\pi \leq x < \frac{8}{5}\pi\right) \frac{(\theta_2 - 10)(\theta_3 - 5)\sqrt{x^2 - x + 1}}{10} \\ & - \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \frac{(\theta_3 - 5)\sqrt{x^2 - x + 1}}{20}. \end{aligned}$$

(b) **Example II** (perfect computer model, nonconvex loss functions)

- Physical process

$$y(x) = \mathbb{I}\left(x < \frac{2}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \cos(x) + \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \varepsilon.$$

- Computer model

$$\begin{aligned} \eta(x, \boldsymbol{\theta}) = & \mathbb{I}\left(x < \frac{2}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \cos(x) + \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \exp\left(\frac{x}{10}\right) \sin(x) \\ & - 2\mathbb{I}\left(x < \frac{2}{5}\pi\right) (\theta_1 + 1) \cos\left(\frac{\theta_1 x}{10}\right) - \mathbb{I}\left(\frac{2}{5}\pi \leq x < \frac{4}{5}\pi\right) \frac{(\theta_1 + 1)(\theta_2 - 10)}{2} \cos\left(\frac{\theta_2 x}{20}\right) \\ & - \mathbb{I}\left(\frac{4}{5}\pi \leq x < \frac{6}{5}\pi\right) \frac{\theta_2 - 10}{2} \cos\left(\frac{\theta_2 x}{10}\right) - \mathbb{I}\left(\frac{6}{5}\pi \leq x < \frac{8}{5}\pi\right) \frac{(\theta_2 - 10)(\theta_3 - 5)}{2} \cos\left(\frac{\theta_2 x}{20}\right) \\ & - \mathbb{I}\left(x \geq \frac{8}{5}\pi\right) \frac{\theta_3 - 5}{2} \cos\left(\frac{\theta_3 x}{10}\right). \end{aligned}$$

(c) **Example III** (imperfect computer model, nonconvex loss functions)

- Physical process

$$y(x) = \exp\left(\frac{x}{10}\right) + \mathbb{I}(x < \pi) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}(x \geq \pi) \exp\left(\frac{x}{10}\right) \cos(x) + \varepsilon.$$

- Computer model

$$\begin{aligned} \eta(x, \boldsymbol{\theta}) = & \exp\left(\frac{x}{10}\right) + \mathbb{I}(x < \pi) \exp\left(\frac{x}{10}\right) \sin(x) + \mathbb{I}(x \geq \pi) \exp\left(\frac{x}{10}\right) \cos(x) \\ & - 5\sqrt{\theta_1^2 - \theta_1 + 1} \left\{ \sin\left(\frac{\theta_1 x}{10}\right) + \cos\left(\frac{\theta_1 x}{10}\right) \right\} - \mathbb{I}(x < \pi) \frac{\sqrt{\theta_2^2 - \theta_2 + 1}}{2} \left\{ \sin\left(\frac{\theta_2 x}{5}\right) + \cos\left(\frac{\theta_2 x}{5}\right) \right\} \\ & - \mathbb{I}(x \geq \pi) \frac{\sqrt{\theta_3^2 - \theta_3 + 1}}{2} \left\{ \sin\left(\frac{\theta_3 x}{10}\right) + \cos\left(\frac{\theta_3 x}{10}\right) \right\}. \end{aligned}$$

In all examples, $x \sim \mathcal{U}(0, 2\pi)$ and $\varepsilon \sim \mathcal{N}(0, 0.01)$. In Examples I and II, the terms can be divided into three input domains $x < 4\pi/5$, $2\pi/5 \leq x < 8\pi/5$, and $x \geq 6\pi/5$, respectively, to represent different blocks (e.g., seasonality in BEM). Hence, θ_1 , θ_2 , and θ_3 are the block parameters that need to be calibrated with dataset associated with the first, second, and third blocks of data collected when $x < 4\pi/5$, $2\pi/5 \leq x < 8\pi/5$, and $x \geq 6\pi/5$, respectively. Note that these data blocks are not disjoint.

For example, θ_1 and θ_2 together affect the computer model response for $2\pi/5 \leq x < 4\pi/5$. Thus, one should not calibrate each θ_i separately. In Example III, the first term is globally employed regardless of the input value. The second and third terms are employed when $x < \pi$ and $x \geq \pi$, respectively, which mimic the block dependency. Thus, θ_1 should be calibrated with the entire dataset, whereas the block parameters θ_2 and θ_3 need to be calibrated with the second and third blocks of data collected when $x < \pi$ and $x \geq \pi$, respectively. Figure 3 illustrates the multiple data blocks that are employed in Examples I, II and III.

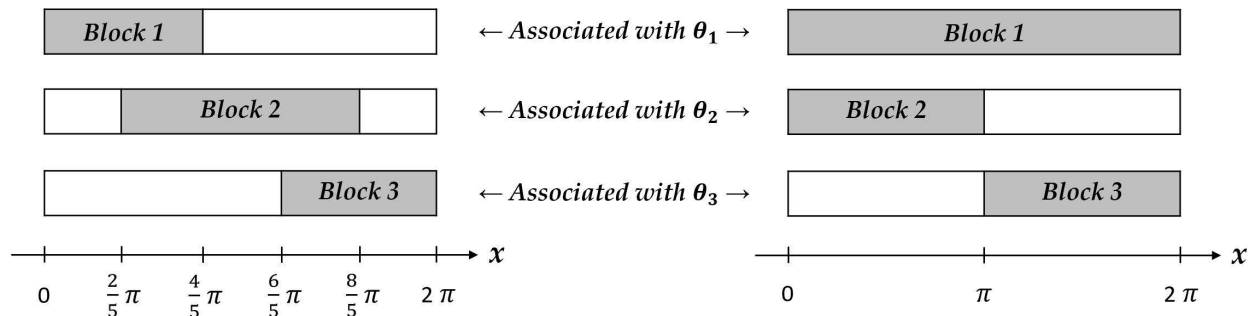


Figure 3 Illustration of multiple data blocks for Examples I & II (left) and Example III (right)

In Examples I and II, the computer models represent perfect computer models with the true parameters $\theta = (\theta_1, \theta_2, \theta_3) = (-1, 10, 5)$, implying that the computer models produce exactly same simulation outcomes as the expected physical process response when their parameters are correctly calibrated. However, Example III exhibits an imperfect computer model where there are no true parameter values. Further, the loss functions in Example I for three blocks, F_1, F_2 , and F_3 , are block-wise convex, i.e., the block loss functions are convex in terms of their associated parameters, given other parameters being fixed. On the other hand, Examples II and III generate nonconvex loss functions.

In implementing the proposed approach, we choose the step size ($\alpha_b^{k,m}$ in (12)) using the backtracking line search in Algorithm 2. The termination condition is set as the norm of the gradient of the loss functions is less than 10^{-4} in both inner and outer iterations.

6.2. Implementation Results

We present the implementation results in comparison with the following two alternatives.

(a) **Hybrid-block Calibration (H-BC):** In this alternative, the computer model correctly simulates its response according to underlying block properties. Specifically, similar to M-BC, it considers the block dependency, using the corresponding subset of data for each block, rather than

using the entire data. However, unlike M-BC that uses multiple loss functions, H-BC uses a single loss function in (5) and optimizes $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ altogether, namely,

$$\boldsymbol{\theta} \leftarrow \arg \min_{\boldsymbol{\theta} \in \Theta} \frac{1}{n} \sum_{j=1}^n (y(x_j) - \eta(x_j, \boldsymbol{\theta}))^2. \quad (40)$$

In solving (40), H-BC uses the correct block of data to compute the gradient of the loss function, but instead of exploiting a block-wise minimization, it updates the parameters all at once, i.e., it updates $\theta_1, \theta_2, \theta_3$ altogether at each GD iteration.

(b) **Single-block Calibration (S-BC):** This alternative ignores the block dependency of parameters. It treats all parameters as global parameters. Thus, when the computer model response is generated, it employs all terms, ignoring $\mathbb{I}(\cdot)$ in the above examples. For example, in Example III, when $\eta(x, \boldsymbol{\theta})$ is simulated for $x < \pi$, it should employ the first global term and term with $\mathbb{I}(x < \pi)$ only, but it also employ the other term with $\mathbb{I}(x \geq \pi)$. Then it uses the single loss function, shown in (5), to calibrate parameters. We include S-BC because in the BEM simulation, seasonality can be ignored and one may blindly run the BEM simulator without correct scheduling.

When loss functions are convex, we can start from an arbitrarily selected starting point to attain a global minimum. In practice, however, we do not know the form of the loss functions, since the computer model is a black box. Thus, we use multiple initial points and select the best estimates that produce the smallest mean squared error (MSE) in a training set consisting of 1000 data samples. We then evaluate the performance using a test set with another 1000 data samples.

Table 3 summarizes the average of calibrated parameters (and standard deviation) in the third to fifth columns, obtained from 200 experiments, as well as MSEs in the test set in the last four columns. Each F_i represents the MSE at the i^{th} data block, and F_{all} in the last column is overall MSE with all test data samples. Clearly, each calibrated value for θ_1 , θ_2 and θ_3 using M-BC in Examples I and II is closer to the true parameter values $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) = (-1, 10, 5)$.

We summarize some observations from Table 3.

- Since M-BC optimizes each block of parameters with the correct subset of data by minimizing the corresponding loss functions, it provides the most accurate calibration results with the lowest MSEs.
- H-BC also uses the correct portion of data, so its MSEs tend to be smaller than those from S-BC. But, its calibrated values deviate from the true values in some cases, e.g., θ_2 and θ_3 in Example I. It is because its single loss function for all three groups of parameters is quite complicated and highly nonconvex with interactions between parameters (e.g., $(\theta_1 + 1)(\theta_2 - 10)$). On the contrary, in the proposed M-BC, each loss function, given fixed parameters in other blocks, is simpler than the H-BC's loss function. For instance, in Example I, even though $\eta(\cdot)$

Table 3 Results of calibrated parameter values and MSE in test sets from 200 experiments (Note: F_i is the empirical loss associated with i^{th} block for $i = 1, 2, 3$. The values inside parentheses are standard deviations)

Ex	Method	Calibrated parameter values			MSE			
		θ_1	θ_2	θ_3	F_1	F_2	F_3	F_{all}
	True value	-1	10	5	-	-	-	-
I	M-BC	-1.00 (0.01)	9.99 (0.02)	5.00 (0.03)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	H-BC	-1.00 (0.02)	9.87 (0.21)	5.36 (1.86)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	S-BC	4.69 (0.83)	-2.05 (1.03)	0.55 (1.37)	42.35 (8.31)	72.3 (5.68)	600.36 (123.71)	3.17 (0.08)
II	M-BC	-1.00 (0.00)	10.00 (0.01)	5.00 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	H-BC	-1.00 (0.00)	10.00 (0.01)	5.00 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	S-BC	-0.15 (1.64)	9.77 (1.61)	5.18 (4.16)	3.78 (16.94)	3.08 (5.59)	6.72 (26.02)	1.55 (0.35)
III	M-BC	0.38 (0.00)	-5.75 (0.44)	8.18 (0.05)	6.89 (0.67)	11.35 (1.25)	2.47 (0.16)	6.89 (0.67)
	H-BC	0.40 (0.01)	7.70 (1.08)	7.96 (1.46)	18.99 (2.43)	35.03 (2.15)	3.07 (3.94)	18.99 (2.43)
	S-BC	0.34 (0.15)	4.52 (2.62)	2.81 (4.23)	41.48 (12.33)	38.95 (7.38)	17.94 (11.08)	41.48 (12.33)

is nonconvex over $(\theta_1, \theta_2, \theta_3)$, the loss function for each parameter becomes block-wise convex. This local convexification helps the M-BC procedure find the solution more effectively.

- The different performance of H-BC in Examples I and II is related to the problem structure where different weights are assigned to each term. In Example I, for the term including $\mathbb{I}(x < 2\pi/5)$ in the computer model, we assigned 1/2, while we assigned 1/10 and 1/20 as weights, respectively, for the terms including $\mathbb{I}(2\pi/5 \leq x < 8\pi/5)$ and $\mathbb{I}(x \geq 8\pi/5)$. Thus, θ_1 is the most influential parameter and the loss function changes rapidly as θ_1 varies. With this problem structure, H-BC weighs θ_1 more, leading to the correct calibration of θ_1 relative to θ_2 and θ_3 . In Example II, the advantage of M-BC, compared to that of H-BC, in terms of calibration accuracy, is less clear. We believe it is because the weights for each term are not substantially different. These results indicate that the performance of H-BC is sensitive to the problem structure. All things considered, M-BC achieves the best accuracy in all examples.
- Obviously, some of the parameter estimates from S-BC largely deviate from their true values in many cases, leading to larger loss values. Recall that S-BC ignores the block property (by ignoring $\mathbb{I}(\cdot)$ terms in the computer models), while observational data from the physical process are collected in different blocks. This mismatch between the physical process responses and simulation outputs leads to poor calibration results with large MSEs.

M-BC also has a computational advantage over the two benchmarks, H-BC and S-BC, in general. This is mainly because M-BC calibrates each block of parameters using its associated data only, unlike H-BC and S-BC that update all parameters simultaneously. Table 4 summarizes the average computation time for each run with Examples I, II, and III. Hereafter, the unit of computation time is seconds, and the average computation time for each run is reported. In Example I, S-BC slightly runs faster than M-BC, but recall that S-BC does not provide accurate results. Overall, M-BC is

computationally more efficient than H-BC and S-BC. Further, its running time is similar in all three examples, whereas the computation time of H-BC and S-BC varies significantly depending on the problem structure.

Table 4 Average computation time for each experiment

Ex	Computation time (sec)		
	M-BC	H-BC	S-BC
I	0.8	12.4	0.7
II	0.7	1.5	5.7
III	0.8	2.1	1.1

We additionally note that comparing computation performance in terms of number of iterations between the three methods is far from trivial. This is because M-BC has the double-loop structure with inner and outer iterations, whereas H-BC and S-BC does not have inner iterations. Moreover, another difficulty is related to the way the data is utilized by the methods. Specifically, as discussed in Section 5.2, assuming that the simulation running time is proportional to the data size n , computational complexity for M-BC can be expressed as $\mathcal{O}(n \log_{\epsilon_{\text{inner}}}(1/\epsilon_{\text{outer}}))$ in the strongly convex case. The analogue for H-BC and S-BC (assuming a tolerance of ϵ_{outer}) can be expressed as $\mathcal{O}(n \log(1/\epsilon_{\text{outer}}))$. While the terms in order notation look similar, hidden are condition number dependencies. For M-BC the dependence is on $\sum_b \kappa_b$ where κ_b is the condition number for each block problem, whereas for H-BC and S-BC the dependence is on κ , the condition number of the full problem. Thus, their computational complexity cannot be readily and easily compared.

6.3. Uncertainty Quantification

Table 5 summarizes the average half length of 95% CI for each parameter and an empirical coverage rate from the 200 experiments in each method. Coverage rates show how often the derived confidence intervals include true parameter values. Ideally, they should be close to the nominal rate, 95% in this case. Since Example III represents an imperfect computer model where the true values do not exist, we do not report the coverage rates. Overall, the results suggest that M-BC provides its empirical coverage rates close to the nominal rate of 95%. On the contrary, the coverage rates of H-BC deviate from the nominal rates in some cases, e.g., those for θ_2 or θ_3 in Example I. The low coverage rates in S-BC are related to inaccurate estimations shown in Table 3.

6.4. Performance Comparison with Other Methods

We compare the performance of M-BC with two common calibration approaches in the literature, including L_2 calibration (Tuo and Wu 2015) and Bayesian calibration (Kennedy and O’Hagan 2001, Higdon et al. 2004).

Table 5 Uncertainty quantification results from 200 experiments (values inside parentheses are standard deviations)

Ex	Method	Average half length of 95% CI			Empirical coverage rate (%)		
		θ_1	θ_2	θ_3	θ_1	θ_2	θ_3
I	M-BC	0.029 (0.001)	0.050 (0.005)	0.055 (0.008)	94.5	93.0	94.0
	H-BC	0.030 (0.002)	0.045 (0.016)	0.064 (0.030)	96.0	81.0	77.6
	S-BC	0.004 (0.001)	0.016 (0.017)	0.107 (0.052)	0.0	0.0	0.0
II	M-BC	0.007 (0.000)	0.030 (0.002)	0.029 (0.001)	95.0	95.5	93.5
	H-BC	0.007 (0.000)	0.030 (0.002)	0.030 (0.002)	95.0	95.0	95.5
	S-BC	0.038 (0.011)	0.189 (0.150)	0.508 (0.727)	0.0	1.6	4.9
III	M-BC	0.036 (0.015)	0.194 (0.162)	0.175 (0.208)	N/A	N/A	N/A
	H-BC	0.097 (0.231)	0.147 (0.121)	0.186 (0.172)	N/A	N/A	N/A
	S-BC	0.065 (0.061)	0.153 (0.150)	0.151 (0.093)	N/A	N/A	N/A

We briefly explain the two benchmark approaches. More details are available in Tuo and Wu (2015), Kennedy and O’Hagan (2001), and Higdon et al. (2004). Given physical observations $\{y(\mathbf{x}_j), \mathbf{x}_j\}_{j=1}^n$, L_2 calibration first obtains the estimated true physical process response $\hat{\zeta}$, where $y(\mathbf{x}) = \zeta(\mathbf{x}) + \epsilon$ with an observation error ϵ , using kernel ridge regression (Wahba 1990) in the reproducing kernel Hilbert space. Similarly, using computer model responses $\{y(\mathbf{x}_{j'}), (\mathbf{x}_{j'}, \boldsymbol{\theta}_{j'})\}_{j'=1}^{n'}$ generated at pre-specified design points, it builds an emulator $\hat{\eta}(\cdot, \cdot)$, often constructed by a GP (Santner et al. 2018), for the computer model. Then, $\boldsymbol{\theta}$ is calibrated by solving the following optimization problem:

$$\hat{\boldsymbol{\theta}}^{L_2} = \arg \min_{\boldsymbol{\theta} \in \Theta} \|\hat{\zeta}(\cdot) - \hat{\eta}(\cdot, \boldsymbol{\theta})\|_{L_2(\Omega)}, \quad (41)$$

Next, the Bayesian calibration approach formulates the physical observation using the linear linkage model as $y(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \gamma(\mathbf{x}) + \epsilon$, where $\gamma(\mathbf{x})$ represents the model discrepancy to correct a model bias between the physical process and the computer model responses, and ϵ is an observation error. Typically, $\eta(\mathbf{x}, \boldsymbol{\theta})$ and $\gamma(\mathbf{x})$ are modeled by GPs, and the calibration parameters and other hyperparameters in mean vectors and covariance functions are estimated from posterior distributions. To explore posterior distributions, Markov chain Monte Carlo (MCMC) is utilized.

We compare the performance between M-BC, L_2 calibration, and Bayesian calibration for the imperfect computer model represented in Example III. For both M-BC and L_2 calibration, we conduct 200 experiments with different training and test sets with 8 starting points. Due to heavy computation time, Bayesian calibration only runs 10 experiments with two prior specifications: uniform and Gaussian. Here, since the computer model is imperfect, we do not know the true parameter values. Instead, we compare MSE values in test sets for each block among three methods.

Table 6 reports the comparison results. With $n = 50$, M-BC achieves the best performance with much smaller MSE values for all blocks, F_1 , F_2 , and F_3 . There is also a computational advantage when using M-BC, taking 0.05 sec, compared to 24.63 sec for L_2 calibration and about 41400 sec

Table 6 Comparison between M-BC, L_2 calibration, and Bayesian calibration in Example III (Note: the values inside parentheses are standard deviations, and “-” means that the running time of the algorithm is excessive so the results are not obtained)

Ex	Method	n	Calibrated parameter values			MSE		
			θ_1	θ_2	θ_3	F_1	F_2	F_3
	M-BC	50	0.38 (0.02)	-3.29 (4.61)	8.22 (0.22)	10.20 (5.47)	18.09 (11.57)	2.61 (0.78)
	L_2 calibration	50	-0.08 (1.36)	-7.12 (2.55)	7.80 (3.30)	59.78 (172.99)	50.06 (91.32)	70.71 (269.67)
	Bayesian (uniform)	50	0.00 (0.00)	-0.61 (1.93)	0.78 (2.48)	21.65 (13.52)	25.19 (13.15)	18.14 (17.13)
	Bayesian (Gaussian)	50	-0.14 (0.43)	-0.78 (2.45)	0.60 (1.88)	25.15 (12.56)	33.29 (15.48)	16.47 (12.67)
III	M-BC	200	0.38 (0.01)	-4.80 (2.65)	7.71 (2.49)	7.99 (2.99)	13.51 (5.88)	2.42 (0.37)
	L_2 calibration	200	0.30 (0.18)	-6.08 (2.16)	8.01 (0.71)	9.28 (11.24)	15.14 (19.93)	3.39 (3.38)
	Bayesian (uniform/Gaussian)	200	-	-	-	-	-	-
	M-BC	1000	0.38 (0.00)	-5.75 (0.44)	8.18 (0.05)	6.89 (0.67)	11.35 (1.25)	2.47 (0.16)
	L_2 calibration	1000	-	-	-	-	-	-
	Bayesian (uniform/Gaussian)	1000	-	-	-	-	-	-

for Bayesian calibration, on average for each run. With a larger sample size of $n = 200$, we could not get Bayesian calibration results because it is computationally prohibitive with our computing resources (each experiment takes 5+ days). While MSE values from L_2 calibration decreases with the larger training data, M-BC still outperforms for all blocks. Meanwhile, computation time for L_2 calibration grows very fast; it takes 50.46 sec, while M-BC only runs for 0.76 sec for each run. Clearly, it demonstrates the advantage of M-BC in both performance metrics.

With the training size of $n = 1000$, M-BC performs very well in terms of MSE values with smallest standard deviations of calibrated parameter values and MSEs. Note that small standard deviations indicate robustness and precision of the respective method. The computation time of L_2 calibration and Bayesian calibration is excessive and they do not produce results in 5 days for each run. In summary, M-BC elicits the superiority of calibration accuracy as well as computational efficiency and scalability, compared to the two existing methods.

We make comments about the difficulty in theoretically comparing computational complexity between the M-BC and Bayesian approach. First, their data generation mechanisms are different. The Bayesian approach generates a *fixed* set of data from pre-specified design points *a priori*, whereas M-BC collects computer response data *on the fly*. Next, in terms of computation after data is generated, in Bayesian calibration, the main computational issue is the numerical integration with respect to the posterior distribution of calibration parameters θ and matrix inversions for each θ value when using GPs in this numerical integration. If the number of observational data and computer experiments are small, the matrix inversion can be done with $\mathcal{O}((n + n')^3)$ in an iteration where n denotes the number of observational data and n' the number of computer experiments (Kennedy and O’Hagan 2001). However, when the data size gets larger, the matrix inversion becomes challenging. Further, it uses MCMC methods to explore posterior distributions. It is known that the MCMC computation generally depends on the number of parameters θ , proposal

distribution, and number of simulation runs (or iterations). Chains' mixing time to its stationary distribution also affect the total computational complexity of the Bayesian approach. On the contrary, the computational complexity for M-BC is $\mathcal{O}(n \log_{\epsilon_{\text{inner}}} (1/\epsilon_{\text{outer}}))$ in the strongly convex case as discussed in Section 5.2. With these reasons, the direct comparison of the computational complexity between the M-BC and Bayesian approach does not seem to be adequate. Further, the termination conditions of the two approaches are different. However, their empirical performance shows the computational efficiency and scalability of M-BC over the Bayesian approach.

6.5. Performance Evaluation of M-BC Variants

We consider some variants of the proposed calibration approach. The first variant removes the inner iteration in M-BC. We refer to this variant as M-BC-V. Even with no inner iterations, it still updates parameters in a block-wise manner, but unlike M-BC, it does not perform the block-wise minimization. It may seem to be similar to H-BC. However, the main difference is that H-BC updates all parameters simultaneously at once, whereas M-BC-V sequentially updates each parameter block based on the most up-to-date information of other blocks. We also consider another variant using a random block selection. This algorithm is the same as M-BC-V but the block is selected randomly, instead of sequentially (or cyclically). This algorithm is referred to as M-BC-VS with 'S' standing for 'stochastic'.

Table 7 summarizes calibration results from these variants. In Examples I and II, M-BC-V and M-BC-VS provide results comparable to the original M-BC. However, they yield inaccurate results in terms of MSE and unstable estimates with higher standard deviations in Example III. It appears that the block-wise minimization (i.e., the inner iteration for each block) in M-BC brings appealing improvements in some cases and provides consistently better results.

Table 7 Results of M-BC variants from 200 experiments (values inside parentheses are standard deviations)

Ex	Method	Calibrated parameter values			MSE			
		θ_1	θ_2	θ_3	F_1	F_2	F_3	F_{all}
I	M-BC	-1.00 (0.01)	9.99 (0.02)	5.00 (0.03)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	M-BC-V	-1.00 (0.02)	9.99 (0.02)	5.00 (0.03)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	M-BC-VS	-1.00 (0.02)	10.00 (0.03)	5.00 (0.03)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
II	M-BC	-1.00 (0.00)	10.00 (0.01)	5.00 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	M-BC-V	-1.00 (0.00)	10.00 (0.01)	5.00 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
	M-BC-VS	-1.00 (0.00)	10.00 (0.02)	5.00 (0.02)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
III	M-BC	0.38 (0.00)	-5.75 (0.44)	8.18 (0.05)	6.89 (0.67)	11.35 (1.25)	2.47 (0.16)	6.89 (0.67)
	M-BC-V	0.36 (0.15)	0.12 (6.76)	-0.99 (7.05)	15.03 (12.43)	23.33 (16.15)	6.80 (18.80)	15.03 (12.43)
	M-BC-VS	0.36 (0.14)	-0.92 (6.30)	1.83 (6.89)	13.15 (10.15)	21.31 (13.98)	5.07 (11.96)	13.15 (10.15)

Further, we investigate whether the block ordering affects the estimation performance in M-BC. With three blocks, there are six pathways in the cyclic block ordering. We do not notice

any significant differences in calibration results among the six pathways in all three examples. In the future, we plan to explore other variants and study their theoretical properties and empirical performance.

6.6. Scalability Analysis and Block Dominance Case

Additionally, we examine the scalability of the method in various settings such as the number of samples, the extent of block overlaps, and the number of blocks/parameters, using a newly designed problem structure. Due to the limited space, we omit the details of the problem structure and results but we discuss them in the online supplementary document. The results show that M-BC yields consistently good results in all settings in terms of calibration accuracy and uncertainty quantification. In particular, it demonstrates that the proposed method is suitable even for the problems with a significant data overlap. In the online supplementary document, we further examine the performance of M-BC when block domination occurs. Our approach provides superior results in terms of both calibration accuracy and uncertainty quantification across different levels of block dominance.

7. Case Study: Building Energy Simulation

We evaluate the performance of our proposed approach in the BEM application. We use the hourly electrical energy consumption data and the simulation outputs obtained from the simulator, EnergyPlus 9.3.0 (U.S. Department of Energy 2019) on a municipal building of Mueller, Austin in Texas. EnergyPlus is a widely used BEM simulator to model both energy consumption—for heating, cooling, ventilation, lighting, plug and process loads—and water use in buildings. Each run for year-long simulation takes about 2 minutes with a standard desktop computer.

Among a large number of parameters employed in EnergyPlus, we choose important parameters in the building energy use, based on domain knowledge and previous studies (Manfredi et al. 2013, Chong et al. 2017), including lighting, ventilation, domestic hot water, window material (optical properties), and heating and cooling systems, as summarized in Table 1. In this case study, we let $\boldsymbol{\theta}_g = (\theta_{g,1}, \dots, \theta_{g,6})^T$ denote the vector of year-long global parameters and $\boldsymbol{\theta}_c = (\theta_{c,1}, \dots, \theta_{c,4})^T$ and $\boldsymbol{\theta}_h = (\theta_{h,1}, \dots, \theta_{h,3})^T$, respectively, represent cooling- and heating-season parameter vectors.

7.1. Implementation Results

We use the implementation setting similar to that in numerical examples. We terminate iterations when the relative difference between consecutive loss function values becomes smaller than a termination tolerance 10^{-4} . We also use a small perturbation $h = 10^{-4}$ in computing the gradient. The step size is decided using the backtracking line search.

Figure 4 shows the example of loss function values for each seasonal block from the proposed M-BC approach. Each dashed vertical line indicates the iteration index where the inner iteration

terminates, and each solid line marks the iteration index for the outer iteration. In the inner iteration, we first minimize the year-long loss F_g , followed by F_c and F_h sequentially. Each plot in Figure 4 illustrates how these loss functions change over iterations. Interestingly, Figure 4(a) shows that F_g decreases mostly when θ_c is updated. This is because the cooling season during March to November (see Table 1) largely overlaps with the global block (January to December) in the studied Texas region. Similarly, the loss F_h during the heating season also decreases when θ_c is calibrated. It is because heating and cooling seasons overlap during three months (see Table 1). Overall, all of the three loss values converge within a few outer iterations, which indicates the important role of inner iterations that calibrates each block of parameters.

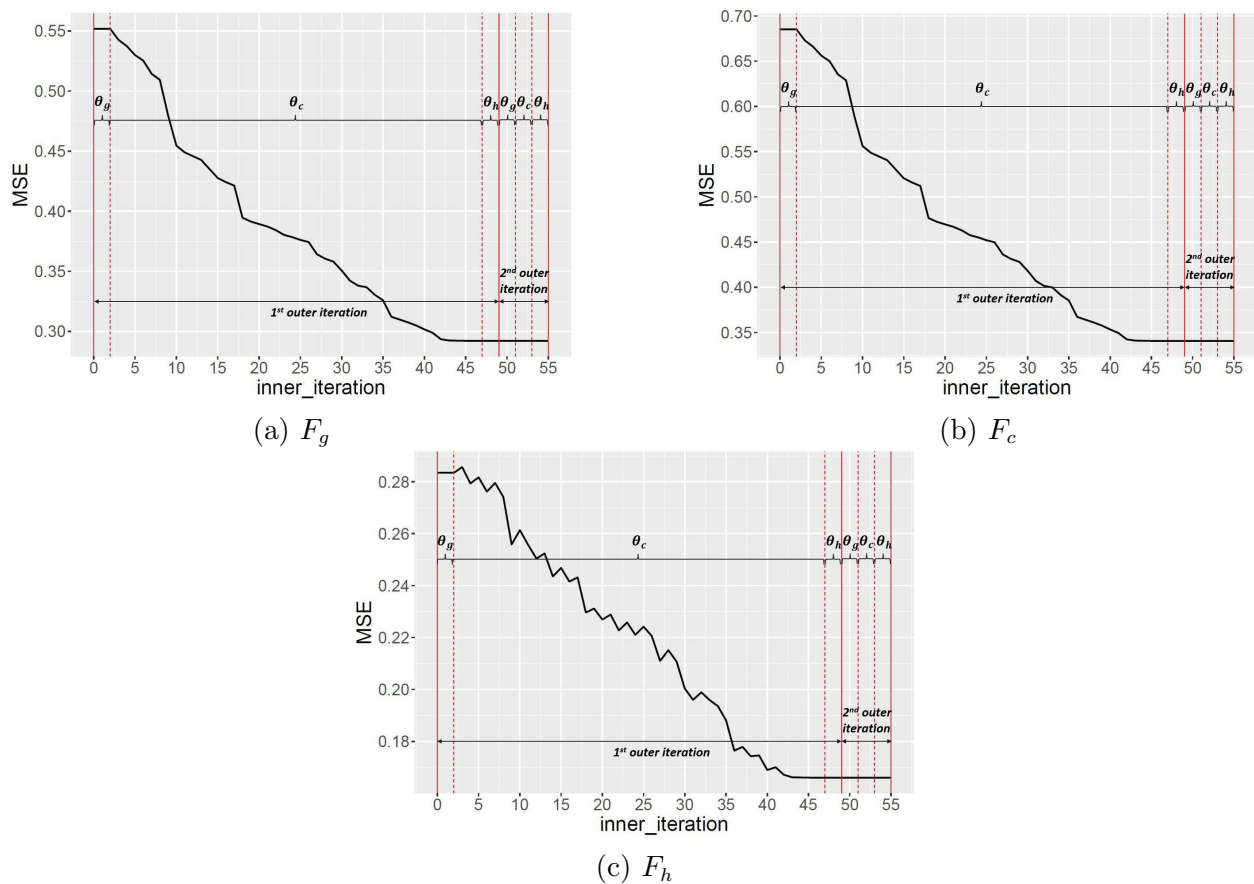


Figure 4 Examples of the trajectories of loss function values (MSEs) from M-BC

Next, we compare the performance of M-BC with H-BC and S-BC. To initialize the parameters in implementing these methods, we use five different sets of starting points obtained from the Latin hypercube sampling (LHS). For each initial setting, we calibrate parameters using a training set consisting of 70% of data points and evaluate the performance with the test set from the remaining 30% of data points. Table 8 summarizes the results from five different starting points, referred to as

LHS 1 through LHS 5. In all cases, M-BC achieves the smallest global (F_g), cooling-season (F_c), and heating-season (F_h) loss values (MSEs). In some cases (e.g., with LHS 3 and 4), the performance of H-BC is comparable to M-BC, but M-BC outperforms H-BC in other initial settings. S-BC performs worse than M-BC and H-BC in most cases.

Table 8 MSEs with 5 different initializations using Latin hypercube design

Method	LHS 1			LHS 2			LHS 3			LHS 4			LHS 5		
	F_g	F_c	F_h	F_g	F_c	F_h	F_g	F_c	F_h	F_g	F_c	F_h	F_g	F_c	F_h
M-BC	0.29	0.34	0.16	0.35	0.39	0.23	0.33	0.36	0.22	0.33	0.39	0.18	0.31	0.36	0.18
H-BC	0.32	0.38	0.16	0.36	0.41	0.24	0.33	0.36	0.22	0.33	0.39	0.18	0.37	0.44	0.23
S-BC	0.36	0.37	0.25	0.40	0.40	0.32	0.40	0.42	0.32	0.36	0.37	0.26	0.39	0.40	0.29

To further compare the approaches, we conduct additional experiments with 10 different training and test sets. Because we get the lowest MSEs with LHS 1 in three methods in Table 8, we fix the starting point at the LHS 1 setting. Table 9 reports the average MSEs and standard deviations in the test sets from the 10 experiments. M-BC achieves the lowest MSEs in all three blocks. Notably, it also consistently produces similar results in 10 experiments, resulting in the small standard deviations. On the contrary, the ranges of the final loss values in H-BC are much wider than those in M-BC, implying that its performance is sensitive to the choice of data. This result is consistent with what we have observed in Section 6 (see Ex III results in Table 3).

7.2. Comparison with Bayesian Calibration Approach

We further compare our approach with Bayesian calibration. To obtain posterior distributions, we exploit the No-U-Turn Sampler (NUTS), which is known to be useful for sampling from a high-dimensional posterior distribution as suggested in Chong and Menberg (2018). The NUTS is developed based on Hamiltonian Monte Carlo (HMC) which allows MCMC to converge faster in that it explores the high-dimensional posterior distribution by optimizing the tuning parameters such as a scaling factor and the number of leapfrog steps. More details are available in Chong and Menberg (2018). Even with NUTS, Bayesian calibration is computationally intensive, as discussed in Section 2. With the computational resources available to us, it takes an unrealistically long time if we use hourly year-long data. Thus, we use weekly data in the Bayesian analysis as in Kristensen et al. (2017b). Even with weekly data, one experiment requires about five days to calibrate the BEM parameters.

We consider two prior specifications in Bayesian approach: non-informative priors with uniform distributions and informative priors with Gaussian distributions, as suggested in Chong and Menberg (2018). In both prior settings, we use the settings in LHS 1 as the prior means for the

Table 9 Comparison results from 10 experiments under LHS 1 initialization (values inside parentheses are standard deviations)

Method	MSE		
	F_g	F_c	F_h
M-BC	0.29 (0.01)	0.34 (0.01)	0.16 (0.01)
H-BC	0.30 (0.02)	0.35 (0.03)	0.17 (0.01)
S-BC	0.35 (0.01)	0.37 (0.01)	0.24 (0.01)

parameters. We set the prior standard deviation as 0.2 in the informative prior settings, as in Chong and Menberg (2018).

Table 10 shows the MSEs in the test sets from 10 independent experiments where we use the posterior means as the calibrated estimates. Compared to M-BC, Bayesian calibration in both prior settings generates larger MSE values especially in the global and cooling-season blocks. We note that the MSE from M-BC in the heating-season block is slightly larger than that of the Bayesian approach with the Gaussian prior. One possible reason is that the underlying heating-season loss is nonconvex and M-BC may attain the local minimum. Additionally, in the attempt to minimize all three losses simultaneously, calibrating the global and cooling-season parameters may negatively affect the loss function in the heating season, because the three data blocks overlap one another and global and cooling season blocks dominate the data portion.

Table 10 Comparison with the Bayesian calibration approach from 10 experiments (values inside parentheses are standard deviations)

Method (Prior)	MSE		
	F_g	F_c	F_h
M-BC	0.29 (0.01)	0.34 (0.01)	0.16 (0.01)
Bayesian (uniform)	0.36 (0.01)	0.45 (0.01)	0.15 (0.00)
Bayesian (Gaussian)	0.33 (0.01)	0.40 (0.01)	0.14 (0.01)

To further investigate, Figure 5 shows the posterior distributions of one of the parameters—lightening level—when non-informative uniform and informative Gaussian priors are used. It turns out that the resulting posterior distribution when using the non-informative prior is still close to its prior distribution, showing large uncertainties. It implies that the calibrated parameter value from the posterior density does not deliver meaningful information. Meanwhile, when the informative prior is employed, the posterior distribution changes from its prior. Still, its MSEs are much larger in the global and cooling-season blocks (see Table 10).

8. Conclusion

In this study, we present a new multi-block parameter calibration methodology when computer model parameters need to be calibrated using different, possibly non-disjoint, subsets of data. We

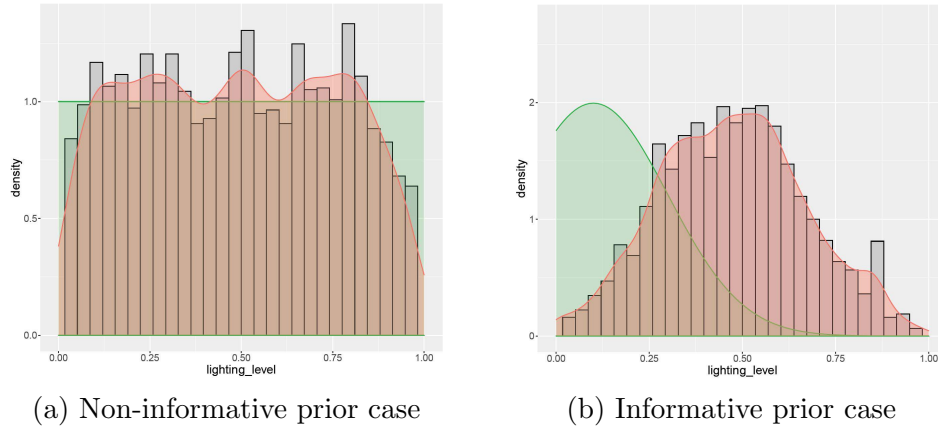


Figure 5 Results from the Bayesian approach for the normalized lightning level parameter (green area: prior, red area: posterior, and gray histogram: posterior samples)

consider multiple loss functions, each for their associated block of parameters that use the corresponding dataset. In addition to the block property, our approach is different from the existing literature that generates the computer model responses at pre-selected parameter settings and builds emulators using the fixed dataset. Although being widely adopted, such traditional approach decouples the data generation stage and calibration stage. On the contrary, our approach adaptively runs the simulation at new parameter settings recommended by the nonlinear optimization procedure. As such, more instructive data is generated from the computer model, which makes the calibration procedure more effective.

Our implementation results in both numerical examples and BEM case study demonstrate that the discrepancy between the physical process and the computer model outputs can be significantly reduced, when the multi-block dependency of parameters is taken into consideration. Further, in our case study, the performance of the proposed M-BC algorithm is more robust, compared to the alternatives. We also construct CIs using asymptotic properties of ML estimators. Our approach yields more stable performance with narrower CIs while maintaining empirical coverage rates close to the nominal rates, compared to other methods. The comparison with the widely used Bayesian calibration elicits the advantage of our approach in terms of improved calibration accuracy.

In the future, we would like to extend the approach under more general settings, e.g., competing cases and time-variant processes (Byon et al. 2016). Developing multi-task learning and BO approaches that can handle overlapping data would also be a possible future research task. Additionally, we plan to use the well-calibrated BEM computer model to control building energy end use such as demand control (Jang et al. 2020, Li et al. 2020).

Acknowledgments

The authors thank the Editor-in-Chief, Senior Editor, Associate Editor, and reviewers for suggestions that helped us greatly improve this manuscript. This work was partially supported by the National Science Foundation under Grants CMMI-1662553, CMMI-2226348, and CBET-1804321. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards, U.S. Department of Commerce, 1972.
- N. V. Bhattacharjee, P. Ranjan, A. Mandal, and E. W. Tollner. A history matching approach for calibrating hydrological models. *Environmental and Ecological Statistics*, 26(1):87–105, 2019.
- A. Booth, R. Choudhary, and D. Spiegelhalter. A hierarchical Bayesian framework for calibrating micro-level models with macro-level data. *Journal of Building Performance Simulation*, 6(4):293–318, 2013.
- S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- E. Byon, A. K. Shrivastava, and Y. Ding. A classification procedure for highly imbalanced class sizes. *IIE Transactions*, 42(4):288–303, 2010.
- E. Byon, Y. Choe, and N. Yampikulsakul. Adaptive learning in time-variant processes with application to wind power systems. *IEEE Transactions on Automation Science and Engineering*, 13(2):997–1007, 2016.
- A. Chong and K. P. Lam. A comparison of MCMC algorithm for the Bayesian calibration of building energy models. *Proceedings of the 15th IBPSA Building Simulation Conference*, 2017.
- A. Chong and K. Menberg. Guidelines for the Bayesian calibration of building energy models. *Energy and Buildings*, 174:527–547, 2018.
- A. Chong, K. P. Lam, M. Pozzi, and J. Yang. Bayesian calibration of building energy models with large datasets. *Energy and Buildings*, 154:343–355, 2017.
- D. Coakley, P. Raftery, and M. Keane. A review of methods to match building energy simulation models to measured data. *Renewable & sustainable energy reviews*, 37:123–141, 2014.
- O. J. Duncan, E. W. Tollner, and H. Ssegane. An instantaneous unit hydrograph for estimating runoff from windrow composting pads. *Applied Engineering in Agriculture*, 29(2):209–223, 2013.
- P. Frazier. A tutorial on bayesian optimization. *ArXiv*, abs/1807.02811, 2018.
- R. B. Gramacy. *Surrogate: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman and Hall/CRC, 2020.
- R. B. Gramacy, D. Bingham, J. P. Holloway, M. J. Grosskopf, C. C. Kuranz, E. Rutter, M. Trantham, and R. P. Drake. Calibrating a large computer experiment simulating radioactive shock hydrodynamics. *Annals of Applied Statistics*, 9:1141–1168, 2015.

- Y. Heo, R. Choudhary, and G. Augenbroe. Calibration of building energy models for retrofit analysis under uncertainty. *Energy and Buildings*, 47:550–560, 2012.
- Y. Heo, G. Augenbroe, and R. Choudhary. Quantitative risk management for energy retrofit projects. *Journal of Building Performance Simulation*, 6(4):257–268, 2013.
- Y. Heo, D. J. Graziano, L. Guzowski, and R. T. Muehleisen. Evaluation of calibration efficacy under different levels of uncertainty. *Journal of Building Performance Simulation*, 8(3):135–144, 2015.
- D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafo, and R. D. Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.
- M. F. Howland, J. B. Quesada, J. J. P. Martinez, F. P. Larrañaga, Ne. Yadav, J. S Chawla, V. Sivaram, and J. O. Dabiri. Collective wind farm operation based on a predictive model increases utility-scale energy production. *Nature Energy*, 2022.
- Y. Jang, E. Byon, E. Jahani, and K. Cetin. On the long-term density prediction of peak electricity load with demand side management in buildings. *Energy and Buildings*, 228:110450, 2020.
- D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
- M. C. Kennedy and A. O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63:425–464, 2001.
- Y. J. Kim and C. S. Park. Stepwise deterministic and stochastic calibration of an energy simulation model for an existing building. *Energy and Buildings*, 133:455–468, 2016.
- M. H. Kristensen, R. Choudhary, R. H. Pedersen, and S. Petersen. Bayesian calibration of residential building clusters using a single geometric building representation. *Proceedings of the 15th IBPSA Building Simulation Conference*, pages 1294–1303, 2017a.
- M. H. Kristensen, R. Choudhary, and S. Petersen. Bayesian calibration of building energy models: comparison of predictive accuracy using metered utility data of different temporal resolution. *Energy Procedia*, 122:277–282, 2017b.
- D. Li, Carol C. Menassa, V. R. Kamat, and E. Byon. Heat-human embodied autonomous thermostat. *Building and Environment*, 178:106879, 2020.
- Q. Li, G. Augenbroe, and J. Brown. Assessment of linear emulators in lightweight Bayesian calibration of dynamic building energy models for parameter estimation and performance prediction. *Energy and Buildings*, 124:194–202, 2016.
- H. Lim and Z. J. Zhai. Comprehensive evaluation of the influence of meta-models on Bayesian calibration. *Energy and Buildings*, 155:66–75, 2017.

- B. Liu, X. Yue, E. Byon, and R. Kontar. Parameter calibration in wake effect simulation model with stochastic gradient descent and stratified sampling. *Annals of Applied Statistics*, 2021.
- H. Liu, Y. Ong, and J. Cai. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, 57(1):393–416, 2018.
- M. Manfren, N. Aste, and R. Moshksar. Calibration and uncertainty analysis for computer models: a meta-model based approach for integrated building energy simulation. *Applied Energy*, 103:627–641, 2013.
- K. Menberg, Y. Heo, and R. Choudhary. Efficiency and reliability of Bayesian calibration of energy supply system models. *Proceedings of the 15th IBPSA Building Simulation Conference*, 2017.
- J. J. Moré and S. M. Wild. Estimating derivatives of noisy simulations. *ACM Transactions on Mathematical Software (TOMS)*, 38(3):1–21, 2012.
- J. Nocedal and J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, 2nd edition, 2018.
- B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.
- J. Sokol, C. C. Davila, and C. F. Reinhart. Validation of a bayesian-based method for defining residential archetypes in urban building energy models. *Energy and Buildings*, 134:11–24, 2017.
- N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, page 1015–1022, 2010.
- K. Swersky, J. Snoek, and R. P. Adams. Multi-task bayesian optimization. *Advances in neural information processing systems*, 26, 2013.
- W. Tian, S. Yang, Z. Li, S. Wei, W. Pan, and Y. Liu. Identifying informative energy data in Bayesian calibration of building energy models. *Energy and Buildings*, 119:363–376, 2016.
- R. Tuo and C. J. Wu. Efficient calibration for imperfect computer models. *Annals of Statistics*, 43:2331–2352, 2015.
- R. Tuo and C. J. Wu. A theoretical framework for calibration in computer models: parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795, 2016.
- R. Tuo, S. He, A. Pourhabib, Y. Ding, and J. Z. Huang. A reproducing kernel hilbert space approach to functional calibration of computer models. *Journal of the American Statistical Association*, pages 1–15, 2021.
- U.S. Department of Energy. Eneplus essentials. Technical Report, 2019.

- G. Wahba. *Spline models for observational data*. SIAM, 1990.
- Z. Xu, C. Jeong, E. Byon, and K. Cetin. Season-dependent parameter calibration in building energy simulation. *Proceedings of 2021 IISE Annual Conference*, pages 423–428, 2021.
- M. You, E. Byon, J. Jin, and G Lee. When wind travels through turbines: A new statistical approach for characterizing heterogeneous wake effects in multi-turbine wind farms. *IISE Transactions*, 49(1):84–95, 2017.
- M. You, B. Liu, E. Byon, S. Huang, and J. Jin. Direction-dependent power curve modeling for multiple interacting wind turbines. *IEEE Transactions on Power Systems*, 33(2):1725–1733, 2018.
- Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.