A history matching approach for calibrating hydrological models

Natalia V. Bhattacharjee[1,3] · Pritam Ranjan[2] · Abhyuday Mandal[3] ·
Ernest W. Tollner[4]

Abstract

Calibration of hydrological time-series models is a challenging task since these models give a wide spectrum of output series and calibration procedures require significant amount of time. From a statistical standpoint, this model parameter estimation problem simplifies to finding an inverse solution of a computer model that generates pre-specified time-series output (i.e., realistic output series). In this paper, we propose a modified history matching approach for calibrating the time-series rainfall-runoff models with respect to the real data collected from the state of Georgia, USA. We present the methodology and illustrate the application of the algorithm by carrying a simulation study and the two case studies. Several goodness-of-fit statistics were calculated to assess the model performance. The results showed that the proposed history matching algorithm led to a significant improvement, of 30% and 14% (in terms of root mean squared error) and 26% and 118% (in terms of peak percent threshold statistics), for the two case-studies with Matlab-Simulink and SWAT models, respectively.

Keywords Contour estimation · Gaussian process model · History matching · Hydrology · Inverse problem · Prediction

---

✉ Abhyuday Mandal
amandal@stat.uga.edu

1   Institute for Health Metrics and Evaluation, University of Washington, Seattle, USA

2   OM&QT, Indian Institute of Management Indore, Indore, MP, India

3   Department of Statistics, University of Georgia, Athens, USA

4   College of Engineering, University of Georgia, Athens, USA

1 Introduction

Hydrological models are commonly used in environmental studies to estimate the water cycle elements in an area of interest. These models use basic principles of mass balance, energy conservation and other principles of physics. The input parameters of these models are often unknown and correspond to physical properties that are difficult to measure. Tuning/calibration of these parameters is required to obtain realistic outputs (Montanari and Toth 2007). This calibration problem is also referred to as the inverse problem in computer experiments literature. This research deals with obtaining the set of input parameters of a computer model that corresponds to a pre-specified target response, which is the observed field data in our application.

In this paper, we focus on calibrating two time-series valued hydrological models that simulate rainfall-runoff dynamics. The input parameters of these models are high dimensional, and the outputs can be very sensitive to small changes in the inputs. Realistic computer models can also be computationally and/or financially expensive, which prohibits numerous evaluation of the simulator. As a result, the calibration of these time-series models is a challenging problem, and an efficient approach to find the inverse solution is extremely important. Several researchers have attempted to solve the inverse problem for hydrological models using different methods via both manual and automated approaches, such as, the Genetic Algorithms, Maximum Likelihood Estimator, Markov Chain Monte Carlo, and Shuffled Complex Evolution (Boyle et al. 2000; Chu et al. 2010; Duan et al. 1992; Franchini and Galeati 1997; Montanari and Toth 2007; Tigkas et al. 2015).

In an unrelated endeavour, Ranjan et al. (2016) and Zhang et al. (2018) proposed a sequential design strategies for estimating the inverse solution, and Vernon et al. (2010) proposed an iterative approach called history matching (HM) for calibrating a galaxy formation model called GALFORM. HM algorithm intelligently eliminates the implausible points from the input (or parameter) space and returns a set of plausible candidates for the inverse solution. However, there are a few aspects of the HM algorithm by Vernon et al. (2010) that differ from our objective. First, the end result of the HM algorithm may be an empty set if there does not exist a plausible inverse solution, and second, the HM algorithm requires a large number of simulator runs which is undesirable in several applications like ours, where the simulator is expensive to evaluate.

We propose a modification in the HM algorithm which allows us to find the inverse solution in fewer simulator runs, and gives us a perfect match if possible, otherwise, the best approximation instead of returning an empty set of inverse solutions. We carry out a simulation study and two case studies of rainfall-runoff models to apply the proposed algorithm in solving this inverse mapping problem. To the best of our knowledge, the HM algorithms have not been applied yet for calibration of hydrological models with time series response.

The case studies refer to the calibration of two rainfall-runoff computer simulators for two target data sets collected at different locations in the state of Georgia, USA, which contains forty to fifty windrow composting systems. The management of the composting pad is crucial since the pad runoff is highly regulated and researchers have tried to estimate runoff in order to provide guidance for retention pond design (Kalaba

et al. 2007; Wilson et al. 2004). The first case study focusses on the calibration of
Matlab-Simulink compartmental dynamic model that estimates the amount of runoff
from the windrow composting pad (Duncan et al. 2013). We wish to calibrate this model
with respect to the composting pad data from the Bioconversion center, University of
Georgia, Athens. The second case study considers the calibration of Soil and Water
Assessment Tool (SWAT) model, a complex hydrological model that simulates runoff
from watershed areas based on climate variables, soil types, elevation and land use
data (Arnold et al. 1994). We use the Middle Oconee River data for calibrating this
model. SWAT is an internationally accepted simulator and used in modeling of the
rainfall-runoff processes across various watersheds and river basins to address climate
changes, water quality, land use and water resources management practices (Dile
et al. 2013; Jayakrishnan et al. 2005; Krysanova and Srinivasan 2015; Srinivasan et al.
2005).

The rest of the manuscript is organized as follows. Section 2 presents the method-
ology for the proposed history matching algorithm for solving the inverse problems.
Section 3 presents a simulation study. The implementation of the proposed strategy
is shown for the two case studies in Sect. 4. Section 5 concludes the article with a
summary and important remarks.

2 Methodology

Let $g(\boldsymbol{x}) := \{g(\boldsymbol{x}, t_i), i = 1, 2, \ldots, L\}$ denote the time-series valued simulator
response for a given input $\boldsymbol{x} \in [0, 1]^d$ (scaled to an unit hypercube for convenience).
Then the objective of the inverse problem is to find the $\boldsymbol{x}$ (or set of $\boldsymbol{x}$'s) that gen-
erate the desired (pre-specified) output $g_0 := \{g_0(t_i), i = 1, 2, \ldots, L\}$ (say). For
many complex phenomena, the realistic computer models are also computationally
and/or financially expensive to run. As a result, standard mathematical techniques
and algorithms cannot be used for solving the inverse problems. Ranjan et al. (2008)
proposed a sequential design approach for efficiently finding the inverse problem for
scalar-valued simulators. However, for this research, the complexity due to time-series
response makes the problem more challenging. Section 2.1 briefly reviews the history
matching (HM) algorithm proposed by Vernon et al. (2010), and then we discuss the
proposed modifications to the HM algorithm in Sect. 2.2.

2.1 History matching algorithm

The history matching algorithm proposed by Vernon et al. (2010) begins by discretizing
the time-series response on T_k time points, say, at $t_1^*, t_2^*, \ldots, t_{T_k}^*$, such that T_k is much
smaller than L. These T_k time points are chosen in such a way that they capture
the defining features of the target response. Then, the HM method finds a common
set of plausible solutions to these T_k inverse problems for scalar-valued simulators,
and declares it as a solution to the general inverse problem. Mathematically, the HM
algorithm finds $\boldsymbol{x} \in [0, 1]^d$ such that $g(\boldsymbol{x}, t_j^*) = g_0(t_j^*)$ for all $j = 1, 2, \ldots, T_k$.

Assuming that the computer model is expensive, the inverse solution must be estimated using the minimal number of model runs. A common practice in computer experiments literature is to build up the methodologies using a flexible statistical surrogate trained on carefully chosen model runs. Vernon et al. (2010) used the most popular surrogate, Gaussian process (GP) model. For simplicity, let us assume that $y(\boldsymbol{x}_i) = g(\boldsymbol{x}_i, t_j^*)$. Then, the n training points, $(\boldsymbol{x}_i, y(\boldsymbol{x}_i))$, $i = 1, 2, \ldots, n$, are modelled as $y(\boldsymbol{x}_i) = \mu + Z(\boldsymbol{x}_i)$, where μ is the mean and $\{Z(\boldsymbol{x}), \boldsymbol{x} \in [0, 1]^d\}$ is a GP, denoted by $Z(\boldsymbol{x}) \sim GP(0, \sigma^2 R)$. This implies that $E(Z(\boldsymbol{x})) = 0$ and the spatial covariance structure defined as $Cov(Z(\boldsymbol{x}_i), Z(\boldsymbol{x}_j)) = \Sigma_{ij} = \sigma^2 R(\theta; \boldsymbol{x}_i, \boldsymbol{x}_j)$. (Notation: We use bold $\boldsymbol{x}_i$ to denote a d-dimensional point in $[0, 1]^d$ and un-bold x_{ik} to denote the kth coordinate of $\boldsymbol{x}_i$.)

For any given input $\boldsymbol{x}^*$ in the design space, the fitted GP surrogate gives the predicted simulator reponse as,

$$\hat{y}(\boldsymbol{x}^*) = \mu + \mathbf{r}(x^*)^T \mathbf{R}^{-1}(\mathbf{y} - \mu \mathbf{1}_n), \tag{1}$$

where $\mathbf{r}(\boldsymbol{x}^*) = [\mathrm{corr}(z(\boldsymbol{x}^*), z(\boldsymbol{x}_1)), \mathrm{corr}(z(\boldsymbol{x}^*), z(\boldsymbol{x}_2)), \ldots, \mathrm{corr}(z(\boldsymbol{x}^*), z(\boldsymbol{x}_n))]^T$, $\mathbf{1}_n$ is a vector of ones of length n, $\mathbf{R}$ is the $n \times n$ correlation matrix for $(Z(\boldsymbol{x}_1), \ldots, Z(\boldsymbol{x}_n))$, $\mathbf{y}$ is the response vector $(y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n))$, and the associated uncertainty estimate is,

$$s^2(\boldsymbol{x}^*) = \sigma^2 \left(1 - \mathbf{r}(\boldsymbol{x}^*)^T \mathbf{R}^{-1} \mathbf{r}(\boldsymbol{x}^*)\right). \tag{2}$$

In practice, the parameters μ, σ^2 and θ in Eqs. (1) and (2) are replaced by their estimates (see Vernon et al. 2010 for details). We used the R package GPfit (MacDonald et al. 2015) for obtaining $\hat{y}(\boldsymbol{x}^*)$ and $s^2(\boldsymbol{x}^*)$ for any arbitrary $\boldsymbol{x}^*$ and a given training data.

The driving force behind the HM algorithm is the implausibility function

$$I_{(j)}(\boldsymbol{x}) = \frac{|\hat{g}(\boldsymbol{x}, t_j^*) - g_0(t_j^*)|}{s_{t_j}(\boldsymbol{x})}, \tag{3}$$

where $\hat{g}(\boldsymbol{x}, t_j^*)$ is the predicted response in Eq. (1), and $s_{t_j}(\boldsymbol{x})$ is the associated uncertainty estimate in Eq. (2). The main idea is to label the design points implausible if $I_{max}(\boldsymbol{x}) > c$, where

$$I_{max}(\boldsymbol{x}) = \max\{I_{(1)}(\boldsymbol{x}), I_{(2)}(\boldsymbol{x}), \ldots, I_{(T_k)}(\boldsymbol{x})\},$$

and c is a pre-determined cutoff (e.g., $c = 3$ as per 3σ rule of thumb). Vernon et al. (2010) further proposed an iterative approach to refine the plausible subset of points from the input space. However, the algorithm is designed to find the set of all plausible inverse solutions and not only the perfect solution. For the Galaxy formation model (GALFORM) application with input dimension $d = 17$, Vernon et al. (2010) used a large training set to start with ($n_1 = 1000$) and ended up with $N = 2011$ points after four iterations.

2.2 Modified history matching algorithm

We propose a few modifications in the history matching algorithm described above. We aim to find only the best possible approximation of the inverse solution instead of the entire plausible set, and prefer to use a reasonably small space-filling design instead of a large design in $[0, 1]^d$ for building the initial surrogate. The optimal choice for the size of design, n_1, is discussed in Section 3.2. The key steps of the proposed modified HM algorithm are summarized as follows:

1. *Choose a discretization-point-set (DPS),* $t_1^*, t_2^*, \ldots, t_{T_k}^*$.
2. *Set* $i = 1$. *Assume* $D_0 = \phi$ *(empty set).*
3. *Choose a training set,* $D_1 = \{x_1, x_2, \ldots, x_{n_1}\} \subset [0, 1]^d$, *using a space-filling design, and evaluate the simulator* $g(x)$ *over* D_1.
4. *Fit* T_k *scalar-response GP-based surrogate to* $g(x, t_j^*)$ *over the training set* $D = D_i \cup D_{i-1}$. *We used the* R *package* GPfit *for surrogate fitting.*
5. *Evaluate the implausibility criteria* $I_{(j)}(x)$ *for* $j = 1, 2, \ldots, T_k$ *over a randomly generated test set* χ_i *of size M (via a space-filling design) in* $[0, 1]^d$ *and combine them via*

$$I_{max}(x) = \max\{I_{(1)}(x), I_{(2)}(x), \ldots, I_{(T_k)}(x)\},$$

   for screening the plausible set of points $D_{i+1} = \{x \in \chi_i \ : \ I_{max}(x) \leq c\}$.
6. *Stop if* $D_{i+1} = \phi$, *otherwise, set* $i = i + 1$, *evaluate the simulator on* D_i *and go to Step 4.*

Instead of using the entire D_{i+1} from Step 5 to Step 6, one can use a space-filling design to find a representative subset of D_{i+1} and then augment it in Step 4 for the next iteration. This will further reduce the total computer model evaluation in solving the inverse problem. Since we assume that the target response is a realization of the simulator output, one can find the best possible approximation of the inverse solutions by minimizing the discrepancy $\delta(x) = \|g(x) - g_0\|$, where $\| \cdot \|$ is the Euclidean distance or L_2 norm. Assuming N is the total number of points at the end of the proposed HM algorithm, the desired inverse solution is given by

$$\hat{x}_{opt} = \underset{1 \leq i \leq N}{\operatorname{argmin}} \|g(x_i) - g_0\|.$$

Instead of minimizing $\delta(x)$ over the training set, one can develop an extraction technique using the final fitted surrogate and/or the DPS.

In summary, we need to identify the following elements to implement the proposed history matching algorithm:

(a) *a computer model* $(g(\cdot))$ *that takes a* d-*dimensional input vector and returns a time-series output,*
(b) *input parameters* (x) *that need to be calibrated,*
(c) *a target response* (g_0) *for calibrating the computer model, and*
(d) *algorithmic parameters:* $n_1, c, T_k, (t_1^*, \ldots, t_{T_k}^*)$ *and M.*
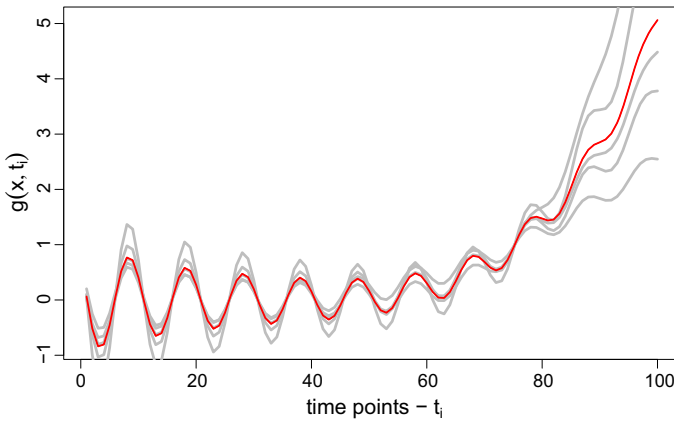
Fig. 1 The illustrative example: a few model outputs (dashed curves) and the target response (solid curve)

Next, we present a simulation study for a comprehensive understanding of the calibration problem and investigate different aspects of the proposed algorithm. Two real-life case studies are presented in Sect. 4.

3 Simulation study

The objective of this simulation study is to discuss the implementation details of the proposed algorithm, and investigate the sensitivity of the algorithmic parameters on the performance efficiency. We consider a simple test function as a computer simulator with two calibration parameters. Specifically, the inputs are $x = (x_1, x_2) \in [0, 1]^2$, which return the following time-series output:

$$g(x, t_i) = \frac{\sin(10\pi t_i)}{(2x_1 + 1)t_i} + |t_i - 1|^{(4x_2 + 2)}, \tag{4}$$

where $t_i = 0.5, 0.52, 0.54, \ldots, 2.50$ (equidistant time points of length $L = 101$ in $[0.5, 2.5]$). We further assume that the true value of the calibration parameter is $x_0 = (0.5, 0.5)$, which generates the target response g_0 in the inverse problem context. Fig. 1 presents the model outputs for a few random input combinations (gray curves) and the target response series (red curve).

Our objective is to find $x \in [0, 1]^2$ such that $g(x) \approx g_0$. We now apply the proposed HM algorithm for solving the inverse problem.

3.1 Application of the proposed algorithm

The implementation procedure stats with choosing the algorithmic parameters. Since the computer simulator, as shown in Fig. 1, appears to be quite simple and $d = 2$, we wish to start with $n_1 = 10$ points for fitting the initial surrogate (note that the choice of n_1 is somewhat arbitrary at this point). The cutoff for selecting the plausible points is
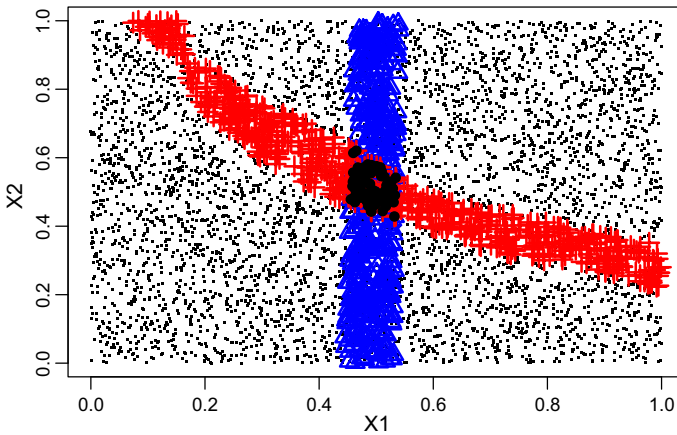
Fig. 2 The illustrative example: selection of the training points according to the implausibility function with cutoff $c = 3$ at the discretization-point-set $DPS = (33, 67)$ in the first iteration of the modified HM algorithm

chosen as $c = 3$, which is guided by the 3σ rule of thumb for normal distributions. We randomly selected $T_k = 2$ and then used $L/3$ and $2L/3$ for discretizing the response, i.e., $DPS = (33, 67)$, since $L = 101$. Finally, we used a randomly chosen large dense sets of size $M = 5000$ for thoroughly searching the follow-up points in the subsequent iterations. That is, the algorithmic parameters are: $n_1 = 10, c = 3, T_k = 2$, $DPS = (33, 67)$ and $M = 5000$.

Figure 2 provides the selection of points in the first iteration, where the points in (blue) triangle and (red) plus correspond to $I_{(j)}(x) \leq 3$ for $t_1^* = 33$ and $t_2^* = 67$ respectively, and the (black) solid circle represents $D_2 = \{I_{max}(x) \leq 3\}$. The iterative procedure gives $|D_2| = 69$.

Subsequently, the augmented training set is of size 79. Now, for the second iteration, Fig. 3 shows the implausibility value of the candidate points. It turns out that D_3 is an empty set, i.e., there are no black solid dots in this figure. This happens because individually $\{x : I_{(j)}(x) \leq 3\}$ are non-empty for both $j = 1, 2$, but $I_{max}(x) \nleq 3$. Thus, the iterative procedure terminates.

As a result, the final training set is of size $N = 79$, and the minimized $\log[\delta(x_i)]$ over the training set is -4.2290, with the estimated inverse solution $\hat{x}_{opt} = (0.4992, 0.5007)$. It turns out that the simulator output at $\hat{x}_{opt}$ is very similar to the target response (see Fig. 4).

3.2 Sensitivity of algorithmic parameters

We now investigate the sensitivity of the algorithmic parameters, n_1, c, T_k and M, with respect to the accuracy of the estimated inverse solution measured by $\log[\delta(\hat{x}_{opt})]$, which is the minimized value of $\delta(x)$ over the augmented training data at the end of the proposed HM algorithm. That is, the lower the value of $\log[\delta(\hat{x}_{opt})]$, the better the parameter combination is. We randomly regenerated the initial training sets, test sets
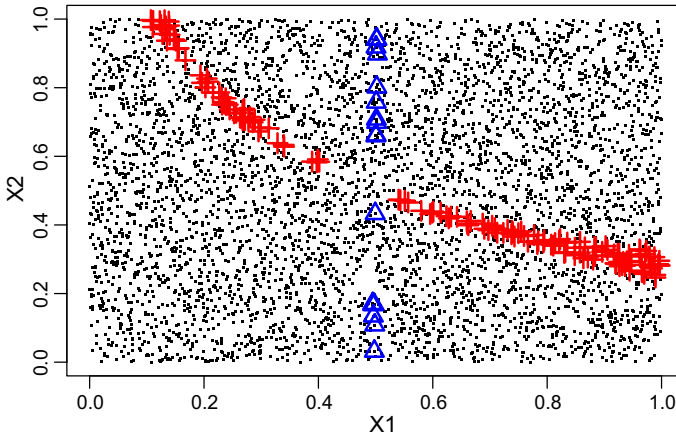
Fig. 3 The illustrative example: selection of the training points according to the implausibility function with cutoff $c = 3$ at the discretization-point-set $DPS = (33, 67)$ in the second iteration of the modified HM algorithm
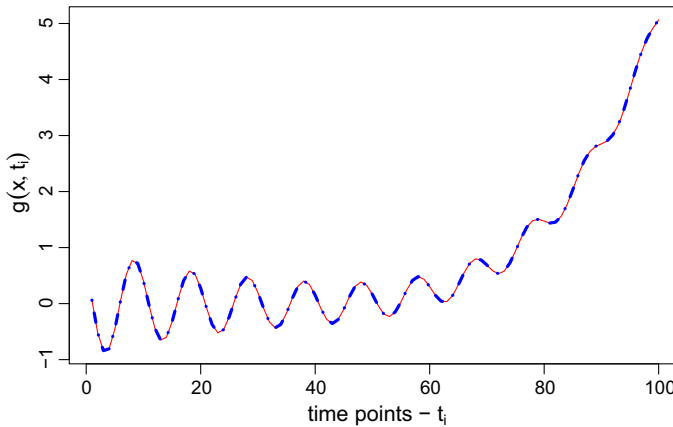

Fig. 4 The illustrative example: the simulator output at the estimated inverse solution $\hat{x}_{opt}$ (dashed blue curve) and the target response (solid red curve)

and the DPS for each combination of $n_1 = (5, 10, 20)$, $c = (1, 2, 3)$, $T_k = (2, 4, 8)$ and $M = (500, 2000, 5000)$, and ran the modified HM algorithm. The results are averaged over 100 random realizations for each combination of n_1, c, T_k and M.

Figure 5 presents the marginal distribution of the median of $\log[\delta(\hat{x}_{opt})]$ over 100 simulations for all possible two-factor combinations of n_1, c, T_k and M. Here, each panel has three sub-panels. For Panel (a), the left most sub-panel corresponds to $n_1 = 5$ and the three dots there correspond to $M = 500$ (solid circle), $M = 2000$ (solid triangle), and $M = 5000$ (plus), respectively. Similarly, the middle sub-panel shows the different values of $\log[\delta(\hat{x}_{opt})]$ for the same three different values of M and a fixed value of $n_1(=10)$. The line segments in other panels and sub-panels can be explained similarly.
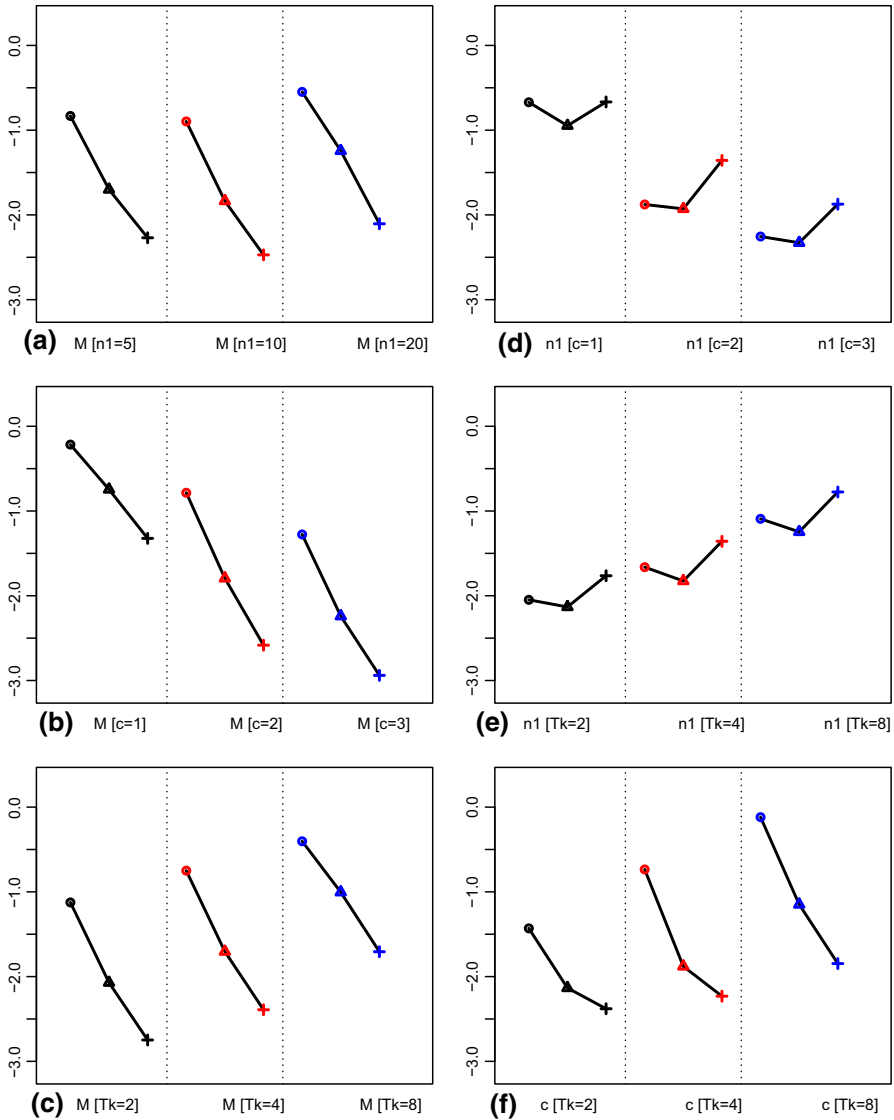
Fig. 5 The illustrative example: marginal distribution of the median of $\log[\delta(\hat{x}_{opt})]$ over 100 simulations for different two-factor combinations of n_1, c, T_k and M

From Fig. 5 we can draw some inference regarding the sensitivity and preference for the algorithmic parameters. For example, Panels (a), (b) and (c) show that as the value of M increases, from 500 to 5000, the value of $\log[\delta(\hat{x}_{opt})]$ decreases monotonically. Naturally, here $M = 5000$ is the best choice. Although it may not be obvious from Panel (a), Panels (d) and (e) clearly demonstrate that $n_1 = 10$ give better results for this example, since in all of these cases, the value of $\log[\delta(\hat{x}_{opt})]$ for $n_1 = 10$ is smaller than that of $n_1 = 5$ or 20. Similarly, Panels (b) and (d) support the choice of
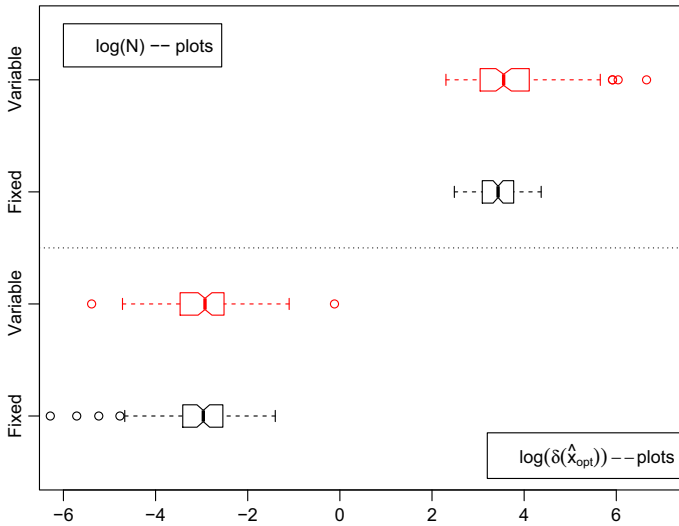
Fig. 6 The illustrative example: sensitivity of selecting DPS measured with respect to the total run-size and optimized $\log[\delta(\hat{\boldsymbol{x}}_{opt})]$

$c = 3$, and the same conclusion can be drawn from Panel (f), since each of the three lines of this panel has the lowest value of $\log[\delta(\hat{\boldsymbol{x}}_{opt})]$ at $c = 3$. Finally, Panels (c), (e) and (f), all clearly indicate that $T_k = 2$ gives the lower value of $\log[\delta(\hat{\boldsymbol{x}}_{opt})]$ than that for 4 and 8.

Together, these six panels of Fig. 5 lead to some intuitive conclusions, such as the higher the value of M or c, the better the performance of the proposed HM algorithm. However, some other conclusions are not that intuitive, and these simulations shed more light on the optimal choice of the algorithmic parameters. For example, it turns out that a higher number of dicretized points (T_k) may not necessarily yield a better performance of the HM algorithm. Finally, if the size of the initial design is either too small or too large, the HM algorithm will not be very efficient. It is important to note that the inferences drawn here are based only on this small simulation study for a simple test function based simulator, and the optimal choices for the algorithmic parameters will have to be carefully chosen for another application.

Since the size of the discretization-point-set (value of T_k) plays a crucial role in the performance of HM algorithm, the actual location of the discretization points (i.e., DPS) may also affect the performance of the proposed algorithm. Figure 6 presents the performance comparison of the proposed algorithm over 100 simulations. Here, we fix $n_1 = 10$, $c = 3$ and $T_k = 2$, and randomly generate training data and implement the algorithm under two scenarios: *Fixed*—$DPS = (33, 67)$, and *Variable*—randomly generate DPS of size T_k using some space-filling criterion. The top panel of Fig. 6 presents $\log(N)$ distribution and the bottom panel displays $\log[\delta(\hat{\boldsymbol{x}}_{opt})]$ distribution over 100 simulations for both fixed and variable scenario.

It is clear from the top panel of Fig. 6 that the choice of DPS fixed at $(33, 67)$ is clearly better than many other alternatives in terms of the total number of computer

model evaluations. The bottom panel shows that both scenarios *Fixed* and *Variable* give comparable accuracy of the final inverse solution, which is expected as the termination of the algorithm depends on the accuracy of the predictor near the target response, as captured by the implausibility function in Eq. (3). In summary, a good choice of the DPS may be helpful in efficiently finding the inverse solution.

Remark 1 For real-life applications, it is certainly infeasible to experiment with different choices of DPS to find the optimal one. One would have to carefully choose DPS to ensure that the important features are captured. The objective of the above simulation study is to demonstrate that the choice of DPS is important for estimating $\hat{x}_{opt}$ with the fewest number of computer model simulator runs.

Remark 2 A reasonable choice of n_1 is also a non-trivial problem. It varies with the end objective, complexity of the underlying simulator response process and the input dimension. In an attempt to answer this question, Loeppky et al. (2009) suggests a rule of thumb of 10 points per input dimension to be enough for getting a good overall idea of the underlying process (i.e., $n_1 = 10d$, where d is the input dimension). However, our objective is to estimate the inverse solution only and not to explore the entire input space with same accuracy. Thus the choice of $n_1 = 10d$ is not necessarily optimal in our case. In a sequential design framework for estimating pre-specified features of interest, e.g., global minimum or the inverse solution, Ranjan et al. (2008) recommends using $n_1 \in [N/3, N/2]$ for building the initial surrgoate.

4 Case studies

This section illustrates the implementation of the proposed history matching approach for the calibration of two hydrological models. The first case study deals with Matlab-Simulink model which simulates runoff from windrow compost pad over a period of time. The second case study refers to estimating the inverse solution of a well-known reservoir model called Soil and Water Assessment Tool (SWAT).

4.1 Case study 1: Matlab-Simulink model

Duncan et al. (2013) investigated the rainfall-runoff relationship for the windrow composting pad, and developed a compartmental model for estimating the amount of runoff from the composting pad (represented as a change in pond volume). It quantifies the surface runoff, infiltration and lateral seepage using differential equations developed for each section of the compost pad. Additionally, the model takes several factors as inputs, for instance, length, width, slope of compost pad, area covered by compost windrows, depth of surface/sub-surface, depression/embankment depths, initial surface/sub-surface water content, and model coefficients of the saturated hydraulic conductivity of the gravel media (K_{sat1}) and the saturated hydraulic conductivity of the supporting soil below the media (K_{sat2}). As per Duncan et al. (2013), the following four inputs/parameters are the most influential: depth of surface, depth of sub-surface and two coefficients of the saturated hydraulic conductivity (K_{sat1} and K_{sat2}). See
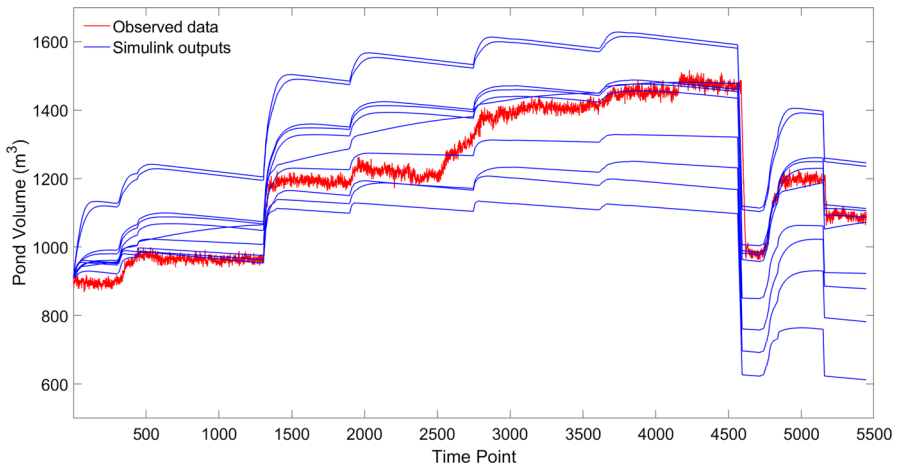
Fig. 7 Field data ($g_0(t_i)$) from Bioconversion center, UGA (represented by the red curve) and the Matlab-Simulink model outputs $g(x, t_i)$ (represented by the blue lines) for $i = 1, 2, \ldots, 5445$ at randomly generated x (depth of surface, depth of sub-surface and two coefficients of the saturated hydraulic conductivity K_{sat1} and K_{sat2}). Time period is December 23, 2010–January 30, 2011

Table 1 Summary statistics of the field data (collected at Bioconversion center, University of Georgia, Athens, USA) required for the Matlab-Simulink Model case study

Variable (units)	Summary					
	Min	Median	Mode	Mean	Std	Max
Rainfall (cm)	0	0	0	0.002	0.011	0.345
Pond volume (m^3)	867.50	1203.00	1192.40	1207.20	191.40	1515.90

Duncan et al. (2013) for more details on data collection, characteristics of composting pad and the Matlab-Simulink model.

For calibration, we used the runoff data (g_0) collected at Bioconversion center, University of Georgia, Athens, USA, as the target response. The raw runoff data (collected on a 10-min interval during 11:50 a.m., December 23, 2010 to 11:50 p.m., January 30, 2011 over $T = 5445$ time points) are represented by the noisy (red) curve in Fig. 7. This figure shows a few random computer model responses superimposed with the field data.

The descriptive statistics of the field data required to compute the runoff are summarized in Table 1.

The objective here is to find the best possible combinations of those four inputs/parameters: depth of surface, depth of sub-surface, K_{sat1} and K_{sat2}, that can generate realistic runoff, i.e., similar to the one obtained from the field data. For convenience in the implementation of the algorithm, the inputs were scaled to $[0, 1]^4$. We start the proposed HM algorithm implementation by choosing $n_1 = 40$ points using a maximin Latin hypercube design (Johnson et al. 1990), and evaluate the simulator on these design points. By carefully examining the nature of the field data, five time
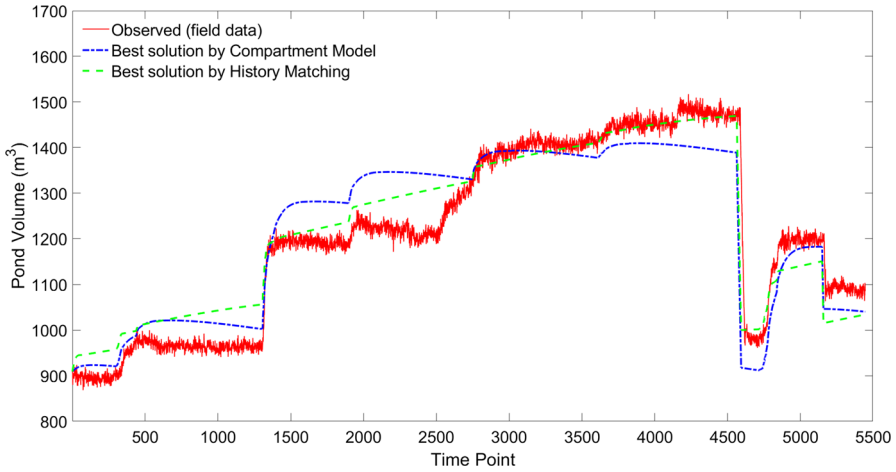
Fig. 8 Calibration Results for the Matlab-Simulink model. The solid red curve represents observed data, blue dash line represents best solution used in the previous study and green dash line corresponds to the best solution using the proposed HM algorithm. Time period is December 23, 2010–January 30, 2011

points ($T_k = 5$) given by {135, 554, 1243, 3232, 4500} were selected from the runoff series (of length $L = 5445$) to discretize the time-series responses. Furthermore, we used the test set of size $M = 5000$ and $c = 3$ for computing the implausibility values and finding the training points for the next iteration. The full implementation required $N = 461$ simulator runs to converge.

The final inverse solution obtained via the proposed HM algorithm is presented in Fig. 8. For a benchmark comparison, we also present the best inverse solution found by Duncan et al. (2013).

For accuracy comparison of different approaches, there are several goodness of fit measures that are more popular in hydrological applications as compared to $\log[\delta(\hat{\boldsymbol{x}}_{opt})]$. We use four such popular measures in this article:

– Root mean squared error

$$
RMSE = \left(\frac{1}{L} \sum_{i=1}^{L} \left| g(\hat{\boldsymbol{x}}_{opt}, t_i) - g_0(t_i) \right|^2 \right)^{1/2}.
$$

– Coefficient of determination R^2 of the simple linear regression (SLR) model, when the dependent variable is the target response and the independent variable is the estimated inverse solution, i.e., R^2 of the SLR model:

$$
g_0(t_i) = g(\hat{\boldsymbol{x}}_{opt}, t_i) + \varepsilon_i, \, i = 1, 2, \ldots, L,
$$

with the assumption of i.i.d. errors ε_i.

Table 2 Goodness of fit comparisons of the proposed HM algorithm and compartmental model (Duncan et al. 2013) for calibrating the Matlab-Simulink model

Matlab-Simulink	RMSE	R^2	NSE	PPTS$_{(5,95)}$	PPTS$_{(1,100)}$
Compartment	71.91	0.86	0.86	4.70	4.75
History matching	55.58	0.93	0.92	3.71	3.77

– Nash–Sutcliffe Efficiency (Nash and Sutcliffe 1970)

$$NSE = 1 - \frac{\sum_{i=1}^{L}[g(\hat{\boldsymbol{x}}_{opt}, t_i) - g_0(t_i)]^2}{\sum_{i=1}^{L}[g_0(t_i) - \bar{g}_0]^2}.$$

– Peak percent threshold statistics (Lohani et al. 2014): $PPTS_{(l,u)}$ is the trimmed mean of

$$|\xi_{t_i}| = \frac{|g_0(t_i) - g(\hat{\boldsymbol{x}}_{opt}, t_i)|}{|g(\hat{\boldsymbol{x}}_{opt}, t_i)|}$$

after eliminating the two tail percentiles, $l\%$ and $u\%$, values of $|\xi_{t_i}|$.

Table 2 summarizes the values of these four goodness of fit measures for the calibration of Matlab-Simulink Model using the proposed HM algorithm and the state-of-the-art Compartmental model (Duncan et al. 2013). For PPTS values we compute measures under two scenarios: no-trimming, and 5% trimming each at the two tails. Note than R^2 and NSE should be maximized, whereas the other two statistics, RMSE and PPTS, should be minimized.

As per Table 2, the proposed HM algorithm outperforms the earlier approach by Duncan et al. (2013) with respect to all three goodness of fit measures, and in particular by a significant $(71.91 - 55.58)/55.58 \times 100 \approx 30\%$ margin according to RMSE, and 26% margin as per $PPTS_{(1,100)}$.

4.2 Case study 2: SWAT model

SWAT model has been widely used for modeling the rainfall-runoff processes across various watersheds and river basins to address climate changes, water quality, land use and water resources management practices (Arnold et al. 1994; Dile et al. 2013; Jayakrishnan et al. 2005; Krysanova and Srinivasan 2015; Srinivasan et al. 2005). This hydrological model takes several inputs, for example, curve number (CN), groundwater delay (GW_{delay}), available water capacity (AWC), baseflow factor (α_{BF}), Manning's coefficient (ν), etc. Based on experts' advise and preliminary variable screening analysis using Sequential Uncertainty Fitting (SUFI2) toolkit, we identified the following five parameters for the calibration exercise: ν, effective hydraulic conductivity in the channel (K), GW_{delay}, groundwater "revap" coefficient (GW_{revap}) and AWC. More details on SUFI2 can be found in Abbaspour et al. (2004, 2007).

Table 3 Summary statistics of the field data (stream flow records observed at $L = 84$ time points for the Middle Oconee River, Georgia) used in the SWAT Model calibration

Variable (units)	Summary					
	Min	Median	Mode	Mean	Std	Max
Streamflow (m^3/s)	0.024	0.277	0.320	0.349	0.280	1.206
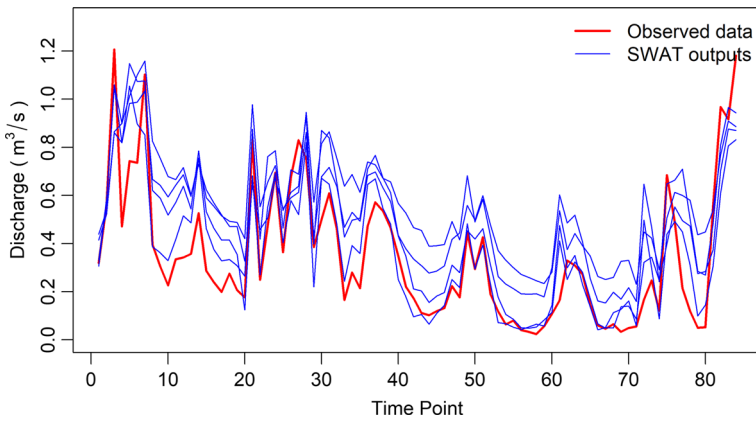
Fig. 9 Middle Oconee river discharge data (USGS Gauge Number 02217500), $g_0(t_i)$ (red curve), and SWAT model discharge outputs $g(\boldsymbol{x}, t_i)$ (blue curves) at random inputs $\boldsymbol{x}$ (Manning's coefficient, effective hydraulic conductivity, groundwater delay, groundwater "revap" coefficient and available water capacity). Time period is January 2003–December 2009

The target response was retrieved from the historical monthly data of streamflow from the US Geological Survey (USGS) water data website for the Middle Oconee River, Georgia, during the period January 2001 to December 2009 (Gauge Number 02217500). We obtained ASTER digital elevation model (DEM) values at 30 m resolution from USGS EarthExplorer platform and Global Climate Data in SWAT format from Texas $A\&M$ University website (https://globalweather.tamu.edu/). We used a warm-up period of 2 years (January 2001 to December 2002) and a calibration period of 7 years (January 2003 to December 2009). For the stream flow records used in SWAT model, the descriptive statistics are listed in Table 3.

Figure 9 shows a few SWAT model runs (in blue—obtained by randomly varying the calibration inputs) and the field data (in red).

Following the steps of the proposed HM algorithm (Sect. 2.2), we rescaled the inputs to $[0, 1]^5$, assigned $n_1 = 50$ for training the initial surrogate, and carefully identified four time instances t_j^* at: 10, 37, 63, 79 for discretizing the output series. The DPS contains two dips and two peaks. Here also we used test sets of size $M = 5000$ and the cutoff for implausibility function to be $c = 3$. Ultimately, the algorithm required $N = 398$ model runs to converge. Figure 10 presents the estimated inverse solution (dashed greed) along with the target response (solid red). For reference comparison, the best solution obtained by SUFI2 (dashed blue) has also been overlaid in Fig. 10.
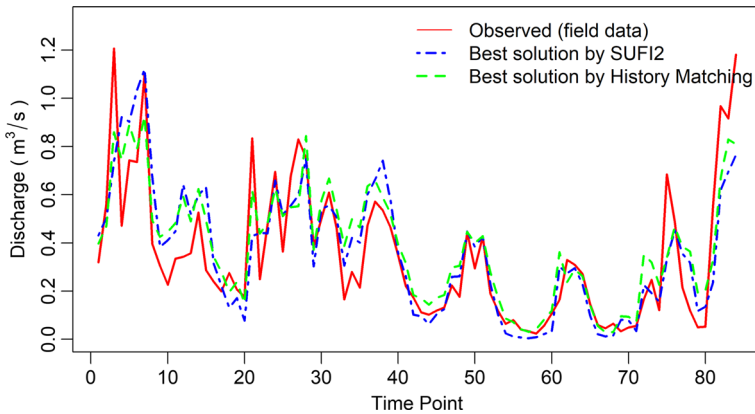
Fig. 10 SWAT model calibration: the solid red curve represents the observed data, blue dashed line represents best solution using SUFI2, and the green dashed line corresponds to the best solution using the HM algorithm. Time period is January 2003–December 2009

Table 4 Accuracy comparisons of the proposed HM algorithm over the state-of-the-art Sequential Uncertainty Fitting (SUFI2) toolkit for the calibration of SWAT model

SWAT model	RMSE	R^2	NSE	$\text{PPTS}_{(5,95)}$	$\text{PPTS}_{(1,100)}$
SUFI2	0.16	0.68	0.67	52.02	65.80
History matching	0.14	0.77	0.75	29.67	30.20

Table 4 presents a more detailed comparison of the two approaches measured with respect to RMSE, R^2, NSE and PPTS. Recall that R^2 and NSE have to be maximized and RMSE and PPTS have to be minimized.

Similar to the previous case study, the proposed HM algorithm exhibits superior performance in terms of all four goodness of fit measures. In particular, the proposed approach demonstrates $(0.16 - 0.14)/0.14 \times 100 \approx 14\%$ improvement as per the RMSE criterion, and an amazing 118% improvement with respect to $PPTS_{(1,100)}$ measure.

5 Discussion

In this study, we applied the proposed modified history matching (HM) algorithm for solving an inverse problem (i.e. calibration problem) for a test function based computer model and two real-life hydrological models. The proposed algorithm demonstrated very good performance in all scenarios. In the first case study (Matlab-Simulink model), the HM algorithm demonstrated approximately 30% better (as per RMSE) performance than the state-of-the-art compartment model calibration results. For the second case study, we observed that the HM algorithm resulted in approximately 14% more accurate (as per RMSE) inverse solution as compared to the one obtained from

SUFI2. Thus, we believe that the proposed HM algorithm can be fruitful for solving calibration problems in hydrological time-series models.

Based on our empirical findings via a simulation study, we infer that the choice of *algorithmic parameters* gives a trade-off between large training-set and accuracy of the inverse solution. Due to the stochastic nature of the HM algorithm, a multi-start approach of the proposed HM algorithm may lead to improved accuracy, and subsampling of D_i in Step 5 may lead to more economical sampling strategy, however one must analyze the tradeoff between the accuracy gain and the additional cost of simulator evaluation for the application at hand. The choice of discretization-point-set is subjective and a key to the success of this algorithm. In practice, one should examine the target response carefully, and choose the points in such a way that they capture the overall variation and important features reasonably well.

Note that the proposed HM approach will find the closest possible approximation in case the simulator turns out to be stochastic and cannot generate the exact same desired output g_0. Although, it is methodologically straightforward to generalize the proposed technique that can adjust for some systematic discrepancies, a bias correction step would require synchronised data on the simulator and actual field trials for multiple input combinations.

References

Abbaspour K, Johnson C, Van Genuchten M (2004) Estimating uncertain flow and transport parameters using a sequential uncertainty fitting procedure. Vadose Zone J 3(4):1340–1352

Abbaspour K, Yang J, Maximov I, Siber R, Bogner K, Mieleitner J, Zobrist J, Srinivasan R (2007) Modelling hydrology and water quality in the pre-alpine/alpine thur watershed using swat. J Hydrol 333(2):413–430

Arnold J, Williams J, Srinivasan R, King K, Griggs R (1994) Swat: soil and water assessment tool. US Department of Agriculture, Agricultural Research Service, Grassland, Soil and Water Research Laboratory, Temple

Boyle DP, Gupta HV, Sorooshian S (2000) Toward improved calibration of hydrologic models: combining the strengths of manual and automatic methods. Water Resour Res 36(12):3663–3674

Chu W, Gao X, Sorooshian S (2010) Improving the shuffled complex evolution scheme for optimization of complex nonlinear hydrological systems: application to the calibration of the sacramento soil-moisture accounting model. Water Resour Res 46:W09530. https://doi.org/10.1029/2010WR009224

Dile Y, Berndtsson R, Setegn S (2013) Hydrological response to climate change for gilgel abay river, in the lake tana basin-upper blue nile basin of ethiopia. PLoS ONE 8(10):e79296

Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour Res 28(4):1015–1031

Duncan O, Tollner E, Ssegane H (2013) An instantaneous unit hydrograph for estimating runoff from windrow composting pads. Appl Eng Agric 29(2):209–223

Franchini M, Galeati G (1997) Comparing several genetic algorithm schemes for the calibration of conceptual rainfall-runoff models. Hydrol Sci J 42(3):357–379

Jayakrishnan R, Srinivasan R, Santhi C, Arnold J (2005) Advances in the application of the swat model for water resources management. Hydrol Process 19(3):749–762

Johnson M, Moore L, Ylvisaker D (1990) Minimax and maximin distance designs. J Stat Plan Inference 26(2):131–148

Kalaba L, Wilson B, Haralampides K (2007) A storm water runoff model for open windrow composting sites. Compost Sci Util 15(3):142–150

Krysanova V, Srinivasan R (2015) Assessment of climate and land use change impacts with swat. Reg Environ Change 15(3):431

Loeppky J, Sacks J, Welch W (2009) Choosing the sample size of a computer experiment: a practical guide. Technometrics 51:366–376

Lohani AK, Goel N, Bhatia K (2014) Improving real time flood forecasting using fuzzy inference system. J Hydrol 509:25–41

MacDonald B, Ranjan P, Chipman H (2015) GPfit: an R package for fitting a Gaussian process model to deterministic simulator outputs. J Stat Softw 64(12):1–23

Montanari A, Toth E (2007) Calibration of hydrological models in the spectral domain: an opportunity for scarcely gauged basins? Water Resour Res 43:W05434. https://doi.org/10.1029/2006WR005184

Nash J, Sutcliffe J (1970) River flow forecasting through conceptual models part i—a discussion of principles. J Hydrol 10:282–290

Ranjan P, Bingham D, Michailidis G (2008) Sequential experiment design for contour estimation from complex computer codes. Technometrics 50(4):527–541

Ranjan P, Thomas M, Teismann H, Mukhoti S (2016) Inverse problem for a time-series valued computer simulator via scalarization. Open J Stat 6(03):528–544

Srinivasan M, Gérard-Marchant P, Veith T, Gburek W, Steenhuis T (2005) Watershed scale modeling of critical source areas of runoff generation and phosphorus transport. JAWRA J Am Water Resour Assoc 41(2):361–377

Tigkas D, Christelis V, Tsakiris G (2015) The global optimisation approach for calibrating hydrological models: the case of medbasin-d model. In: Proceedings of the 9th world congress of EWRA, pp 10–13

Vernon I, Goldstein M, Bower R (2010) Galaxy formation: a Bayesian uncertainty analysis. Bayesian Anal 5(4):619–669

Wilson B, Haralampides K, Levesque S (2004) Stormwater runoff from open windrow composting facilities. J Environ Eng Sci 3(6):537–540

Zhang R, Lin CD, Ranjan P (2018) A sequential design approach for calibrating a dynamic population growth model. arXiv:1811.00153

Natalia V. Bhattacharjee is Research Scientist in the Institute for Health Metrics and Evaluation at the University of Washington, USA. She received her bachelor's and master's degrees from Omsk State Technical University, her secondary master's degree and Ph.D. from University of Georgia (UGA). Before joining UGA, she was a Research Scientist in Water Problems Institute at Russian Academy of Sciences, Moscow, Russia. Her research areas include ecological modeling, time series analysis and geostatistics.

Pritam Ranjan is Associate Professor in the Operations Management and Quantitative Techniques (OM&QT) area at IIM Indore. Prior to joining IIM Indore he was an Associate Professor in the Department of Mathematics and Statistics at Acadia University, Nova Scotia, Canada (during 2007–2015). He obtained B.Stat. and M.Stat. degrees from the Indian Statistical Institute, Kolkata, and a Ph.D. degree in Statistics from Simon Fraser University, BC, Canada. His research areas include design and analysis of computer experiments, statistical modeling for computer simulators, sequential design for feature estimation, and a projective geometric approach for multi-stage fractional factorial designs with randomization restrictions. Currently he is Associate Editor for *the Canadian Journal of Statistics.*

Abhyuday Mandal is Professor in Department of Statistics at the University of Georgia, USA. He received his bachelor's and master's degrees from Indian Statistical Institute, Kolkata, another master's degree from University of Michigan and Ph.D. in 2005 from Georgia Institute of Technology. His research interests include design of experiments, industrial statistics, optimization techniques, small area estimation, drug discovery and fMRI data analysis. Currently he is Associate Editor for *Statistics and Probability Letters*, *Sankhya—Series B* and *Journal of Statistical Theory and Practice*.

Ernest W. Tollner is Professor in the College of Engineering at the University of Georgia, USA. He received his bachelor's and master's degrees from University of Kentucky in 1972 and 1974, and Ph.D. from Auburn University in 1981. His research areas of specialty include hydrology, natural resource engineering, watershed assessment and post-harvest quality. Currently he is Associate Editor for the *Transactions of the American Society of Biological and Agricultural Engineering*.