Journal of Statistical Theory and Practice (2023) 17:23
https://doi.org/10.1007/s42519-022-00312-5

ORIGINAL PAPER

Solving an Inverse Problem for Time-Series-Valued Computer Simulators via Multiple Contour Estimation

Pritam Ranjan[1] · **Joseph Resch**[2] · **Abhyuday Mandal**[3]

Accepted: 27 November 2022

Abstract

Computer simulators are often used as a substitute of complex real-life phenomena, which are either expensive or infeasible to experiment with. This paper focuses on how to efficiently solve the inverse problem for an expensive to evaluate time-series-valued computer simulator. The research is motivated by a hydrological simulator which has to be tuned for generating realistic rainfall–runoff measurements in Athens, Georgia, USA. Assuming that the simulator returns $g(x, t)$ over L time points for a given input x, the proposed methodology begins with a careful construction of a discretization (time-) point set of size $k \ll L$, achieved by adopting a regression spline approximation of the target response series at k optimal knot locations $\{t_1^*, t_2^*, \ldots, t_k^*\}$. Subsequently, we solve k scalar-valued inverse problems for simulator $g(x, t_j^*)$ via the contour estimation method. The proposed approach, named MSCE, also facilitates the uncertainty quantification of the inverse solution. Extensive simulation study is used to demonstrate the performance comparison of the proposed method with the popular competitors for several test-function-based computer simulators and a real-life rainfall–runoff measurement model.

Keywords Expected improvement criterion · Gaussian process model · History matching · Hydrological simulation model · Regression splines · Uncertainty quantification

This article is part of the topical collection"Special Issue: AISC-2021 Special Collection" Guest edited by Tahani Coolen-Maturi, Javid Shabbir, and Arman Sabbaghi.

✉ Abhyuday Mandal
amandal@stat.uga.edu

1 Indian Institute of Management, Indore, MP, India

2 University of California, Los Angeles, CA, USA

3 University of Georgia, Athens, GA, USA

Published online: 08 February 2023

1 Introduction

Complex physical experiments are frequently expensive and impractical to perform. The growth in computing power during modern times offers an alternative to carry out such experiments via computer simulation models, such as dynamic traffic patterns of a metropolitan intersection, energy harvesting via wind farms and tidal turbines, quantification of volcanic hazards, hydrological behaviors of an ecosystem, the spread of a wildfire, weather modeling, formation of galaxies, and so on [2, 6, 14, 26, 28, 29, 32, 33, 39, 45, 53]. Realistic computer simulators of complex physical, engineering, and sociological phenomena are often computationally expensive to run, and thus innovative design and analysis techniques have to be developed for deeper understanding of the process.

Over the last three decades, a plethora of innovative methodologies on computer experiments have been developed in the statistics and engineering literature. Some of the seminal papers focus on, the emulation of simulator response via Gaussian Process (GP) models [49], space-filling designs for building good surrogate models to emulate deterministic simulator outputs [23], sequential design approach via a merit-based criterion called the expected improvement for global optimization of an expensive to evaluate simulator [24], a Bayesian approach for the emulation of simulator models in the presence of calibration parameters [27], treed-GP for the emulation of non-stationary simulators [17], and localized GP models [15]. For a detailed discussion on methodological development on this topic, see Santner et al. [50], Fang et al. [12], Rasmussen and Williams [47] and Gramacy [18].

In this paper, we focus on solving an inverse problem for expensive to evaluate computer simulator which produces time-series outputs. Let $g(x) = \{g(x, t_j), j = 1, 2, \ldots, L\}$ be the simulator output for input $x \in \chi$, a hyper-rectangle scaled to $[0, 1]^d$, where d is the input dimension. The inverse solution, S_0, with respect to a pre-specified target $g_0 = \{g_0(t_j), j = 1, 2, \ldots, L\}$, refers to the set of inputs x that generate g_0, i.e.,

$$S_0 = \{x \in \chi : g(x, t_j) = g_0(t_j), j = 1, 2, \ldots, L\}.$$

The application that motivated this study comes from a hydrological simulation model which predicts the rate of rainfall–runoff and sediment yield for a windrow composting pad [11]. Here, the objective is to find the inputs of the hydrological model that generates outputs as close as possible to the real data collected from a watershed from the Bioconversion center at the University of Georgia, Athens, USA.

Inverse problem for expensive to evaluate *scalar-valued* simulators has been extensively investigated in the past few years (e.g., [1, 3, 4, 9, 10, 22, 38, 42, 44, 48]). A closely related research topic is referred to as the estimation of calibration parameters, where the computer simulator takes two types of inputs, controllable design variables and fixed but unknown calibration parameters. Kennedy and O'Hagan [27] proposed a Bayesian framework that accounts for the two types of inputs and models a potential systematic discrepancy between the observed field data and the simulator response. This model received significant attention in both computer experiments and engineer-

ing literature, for instance, Tuo and Wu [52], Pratola et al. [43], Brown and Hund [8], Perdikaris and Karniadakis [40], and Perrin [41].

For simulators with *time-series* response and only controllable inputs, the inverse problem literature includes Bhattacharjee et al. [5], Ranjan et al. [46], Toscano-Palmerin and Frazier [51], Vernon et al. [53], and Zhang et al. [56]. We focus on this setting. Vernon et al. [53] developed an innovative history matching (HM) algorithm for solving the inverse problem of a galaxy formation model called GALFORM. This is multistage sampling strategy, which intelligently eliminates the implausible points from the input space and returns a set of plausible candidates. The main idea was to first select a handful of time-points from the target response series (referred to as the discretization point set (DPS) = $\{t_1^*, t_2^*, \ldots, t_k^*\}$, where $1 \leq t_1^* < \cdots < t_k^* \leq L$) and then optimize a joint discrepancy criterion (called the implausibility function) between the target and predicted response from the GP surrogates of $g(x, t_j^*)$ − the scalar-projection of the process at DPS locations t_j^*, for $j = 1, \ldots, k$. It turns out that the HM algorithm requires too many simulator runs, and for expensive to evaluate computer simulators, this approach would be unaffordable. Recently, Bhattacharjee et al. [5] proposed a small modification in the sampling strategy of the HM algorithm which reduced the required simulator runs without compromising the accuracy of the inverse solution.

In a simplified approach to such an inverse problem, Ranjan et al. [46] introduced a new pseudo-scalar-valued simulator $w(x) = \|g(x) - g_0\|$ and then find the minimizer of $w(x)$ using a global optimization method − build GP surrogate for $w(x)$ coupled with sequential design techniques via the expected improvement (EI) criterion developed by Jones et al. [24]. In the same spirit, Zhang et al. [56] minimized $w(x) = \|g(x) - g_0\|$; however, instead of fitting a scalar-valued GP surrogate of $w(x)$, the authors used a singular value decomposition (SVD)-based GP surrogate [21] for $g(x)$ and developed a saddlepoint approximation of the EI expression of Jones et al. [24], referred to as the saEI approach.

In this paper, we proposed a new MSCE method for solving an inverse problem for time-series-valued computer simulators. The proposed approach has two key components. Inspired by the HM algorithm, we first discretize the target response series at DPS. However, Bhattacharjee et al. [5] and Vernon et al. [53] used an ad hoc method (or a subjective expert opinion) for choosing the DPS. We suggest a more formal approach by fitting a regression spline to the target series g_0 and then identify the desired DPS as the optimal knot locations. We investigated both the sequential search and the simultaneous search methods for finding optimal knots. Then, we solve k scalar-valued inverse problems, i.e., estimate

$$S_j = \{x \in \chi : g(x, t_j^*) = g_0(t_j^*)\}, \ j = 1, 2, \ldots, k.$$

Finding S_j is essentially a contour estimation problem, as in Ranjan et al. [44]. As per our knowledge, the existing literature on inverse problems for time-series-valued simulators (e.g., [46, 53, 56]) uses the global minimization criterion by Jones et al. [24]. In this paper, we propose using the contour estimation EI criterion for iteratively solving these k scalar-valued inverse problems. At the end, the inverse solution of the underlying dynamic simulator is obtained by taking the intersection of all scalar-

valued inverse solutions, which is further used to quantify the uncertainty associated with the estimated inverse solution. Theoretical result that establishes the estimation of the desired inverse solution is also presented. Extensive simulation studies have been used to demonstrate that the proposed approach is more accurate and reliable than its competitors. The results are compared with those of modified HM algorithm [5], scalarization method [46], and saEI method [56].

The remaining sections are outlined as follows: Section 2 reviews the concepts integral to the proposed method and the competing approaches, i.e., the scalarization method by Ranjan et al. [46], the HM approach proposed by Vernon et al. [53] with modification in Bhattacharjee et al. [5], and the saEI method by Zhang et al. [56]. Section 3 provides the elements of the proposed multiple scalar-valued contour estimation (MSCE) method, uncertainty quantification of the inverse solution, and thorough implementation details of the steps. In Sect. 4, we present simulation studies to establish the superiority of the proposed method via three test function-based simulator. Section 5 discusses the real-life motivating hydrological-simulation application. We provide some concluding remarks in Sect. 6.

2 Review of Existing Methodology

In this section, the existing methods that set precedence for this paper are presented. We briefly review GP-based models used as surrogates of the simulator outputs and the EI criterion for choosing the follow-up trials in the sequential design framework. Subsequently, the scalarization method by Ranjan et al. [46], HM algorithm of Vernon et al. [53] and Bhattacharjee et al. [5], and saEI method by Zhang et al. [56] are also reviewed.

2.1 Gaussian Process-based Models

The evaluation of a computer simulator for complex phenomena can often be computationally expensive, and hence, the emulation via a statistical surrogate becomes much more practical. Sacks et al. [49] presents a GP model as a useful surrogate of deterministic simulator output. For a set of input–output combinations, a stationary GP model, also called as the ordinary Kriging, assumes:

$$y(x_i) = \mu + Z(x_i), \qquad i = 1, \ldots, n,$$

where μ is the mean and $Z(x_i)$ is a GP with $E(Z(x_i)) = 0$ and a covariance structure of $Cov(Z(x_i), Z(x_j)) = \sigma^2 R(\theta; x_i, x_j)$. There are several popular choices of $R(\cdot, \cdot)$, e.g., Gaussian correlation, power-exponential family, and Matérn correlation. The power-exponential correlation structure will have the $(i, j)^{\text{th}}$ term $R_{ij}(\theta)$ as:

$$R(Z(x_i), Z(x_j)) = \prod_{k=1}^{d} \exp\left\{ -\theta_k \mid x_{ik} - x_{jk} \mid^{p_k} \right\} \quad \text{for all } i, j, \tag{1}$$

where $0 < p_k \leq 2$ are smoothness parameters and θ_k measure the correlation strength. In this paper, we assume power exponential correlation with $p_k = 1.95$ (for numerical stability and smoothness). The best linear unbiased predictor for the response at any unsampled point x^* is given by:

$$\hat{y}(x^*) = \hat{\mu} + r(x^*)^T R_n^{-1}(y - 1_n\hat{\mu}), \tag{2}$$

where $r(x^*) = \left[\text{corr}(z(x^*), z(x_1)), \ldots, \text{corr}(z(x^*), z(x_n))\right]^{\mathrm{T}}$, R_n is the $n \times n$ correlation matrix with elements R_{ij} (as in Eq. (1)), and the prediction uncertainty is quantified by

$$s^2(x^*) = \hat{\sigma}^2\left(1 - r(x^*)^T R_n^{-1} r(x^*)\right). \tag{3}$$

The flexibility of the correlation structure, and the closed-form expressions for mean prediction and associated uncertainty make the GP model a popular surrogate for complex computer model outputs. Throughout this paper, the *R* package `GPfit` [34] has been used to fit the basic scalar-valued GP models.

Fitting a GP model requires numerous inverse calculations of size $n \times n$ each, which becomes computationally daunting as n increases and particularly for simulation studies when the entire exercise has to be repeated thousands of times. Gration and Wilkinson [14] developed an *R* package called `laGP`—a local approximate GP (laGP) model for large data sets. The main idea is to fit local GP model for prediction at any given point in the input space. The process of finding the local set of size $m(\ll n)$ starts with a k-nearest neighbor set around the point of interest and then selecting the remaining $m - k$ points guided by a model-based criterion. Finally, the prediction at the point of interest is obtained using the GP model built on this local neighborhood set of size m. See Gramacy [15] for methodological details. In this paper, if $n > 50$, `laGP` package has been used for all GP model fittings within the simulation exercises. For $n \leq 50$, we used the `GPfit` package.

2.2 Sequential Design

It has been established on many occasions that sequential designs outperform its competitors for finding a pre-specified feature of interest, e.g., the global minimum or the inverse solution, for a computationally intensive deterministic scalar-valued computer simulator [13, 18, 24, 42, 44, 50, 56]. The basic framework consists of three key steps, finding a good initial design, fitting the statistical surrogate and choosing the follow-up trials by optimizing a merit-based criterion (EI is the most popular one).

In computer experiments, the popular choice for an initial design includes a space-filling design such as a maximin Latin hypercube design (LHD) [36, 54], a maximum projection LHD [25], uniform design, and orthogonal array-based LHD [55]. Once an initial design is chosen, the responses are generated by evaluating the simulator at each input. A surrogate model is then fitted to the training data $(x_i, y_i), i = 1, 2, \ldots, n$. We use the GP / laGP model (detailed in Sect. 2.1) for this purpose. After which,

a sequential design criterion such as EI is evaluated over the entire input space to find the input x_{new} − the maximizer of EI (see [24] and [6] for details). The x_{new} and corresponding true simulator response are augmented to the training set (i.e., $n = n + 1$). The surrogate (GP model) is refitted to this augmented training set. The iterative process of optimizing EI to choose x_{new} and refitting the surrogate to the augmented data is repeated until the total budget of N points is exhausted. The feature of interest (e.g., the global optimum or the inverse solution) would be extracted from the final surrogate fit.

2.3 Inverse Problem via Scalarization

Ranjan et al. [46] assumed $w(x) = \|g(x) - g_0\|$ to be the output of a new scalarized simulator, which is expensive to evaluate, and thus, the popular sequential approach by Jones et al. [24] was applied to find the global minimum. That is, a GP model (Sect. 2.1) is used to emulate the scalar-valued response $w(x)$, and the EI criterion by Jones et al. [24] dictates how to choose the follow-up points. Note that in Sect. 2.1, $y(x)$ denotes a scalar simulator response, whereas in this section, we denote $w(x)$ as the scalar response. The EI criterion, as per the Gaussian predictive distribution with mean and variance given by (2) and (3), has a closed-form expression:

$$E[I(x)] = (w_{min} - \hat{w}(x))\Phi\left(\frac{w_{min} - \hat{w}(x)}{s(x)}\right) + s(x)\phi\left(\frac{w_{min} - \hat{w}(x)}{s(x)}\right),$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the normal probability density function (pdf) and cumulative distribution function (cdf), respectively.

The EI-based approach has gained immense popularity because it facilitates a balance between the exploration and exploitation, which further implies that the entire input space is explored thoroughly and hence eventually all global minima would be found. That is, if there are more than one solution of the inverse problem, then the EI-based approach would be able to detect them. Finally, the inverse solution is obtained by minimizing the responses over the training data or the predicted response over a dense set via the final fitted surrogate.

2.4 EI Criterion for Contour Estimation

For a scalar-valued deterministic computer simulator, Ranjan et al. [44] developed an EI criterion for estimating the inputs that lead to a pre-specified target response $y(x) = a$. The proposed improvement function is given by:

$$I(x^*) = \epsilon^2(x^*) - \min\Big[\{y(x^*) - a\}^2, \epsilon^2(x^*)\Big].$$

where $\epsilon(x^*) = \alpha s(x^*)$ for a positive constant α (e.g., $\alpha = 0.67$ corresponds to 50% confidence, and $\alpha = 1.96$ represents 95% level of confidence, under normality), $s(x^*)$ is defined in (3), and a is the pre-specified target response. Hence, the EI value (which is simply the expected value of the improvement function under the predictive

distribution $y(x^*) \sim N(\hat{y}(x^*), s^2(x^*))$ is:

$$\begin{aligned} E[I(x^*)] = &\left[\epsilon^2(x^*) - \{\hat{y}(x^*) - a\}^2\right]\Big\{\Phi(u_2) - \Phi(u_1)\Big\} \\ &+s^2(x^*)\Big[\{u_2\phi(u_2) - u_1\phi(u_1)\} - \{\Phi(u_2) - \Phi(u_1)\}\Big] \\ &+2\Big\{\hat{y}(x^*) - a\Big\}s(x^*)\Big\{\phi(u_2) - \phi(u_1)\Big\}, \end{aligned} \tag{4}$$

where $u_1 = [a - \hat{y}(x^*) - \epsilon(x^*)]/s(x^*)$, and $u_2 = [a - \hat{y}(x^*) + \epsilon(x^*)]/s(x^*)$.

Similar to the EI in Jones et al. [24], this EI criterion also facilitates the balance between local and global searches. In other words, all pieces of the contours are expected to be detected eventually.

2.5 History Matching for the Inverse Problem

HM approach was developed by Vernon et al. [53] and was subsequently modified by Bhattacharjee et al. [5] to solve the inverse problem for a time-series-valued computer simulator. The HM approach starts by selecting a handful of time-points $\{t_1^*, t_2^*, \ldots, t_k^*\}$, which are referred to as a DPS and has size k, which is significantly smaller than L, the total length of the response series. The said approach uses the simulator outputs at only the DPS time-points and approximates the desired inverse solution by eliminating the set of implausible points from the input space via an innovative criterion called the implausibility function.

The HM algorithm is implemented via a multistage sampling technique. First, a large space-filling initial design $\{x_1, x_2, \ldots, x_n\}$ is used to evaluate the time-series-valued simulator and extract the scalar projections of the input–output training set at the DPS locations. Subsequently, the algorithm iterates between the following four steps:

(1) For $j = 1, 2, \ldots, k$, fit k scalar-valued GP surrogates to $\{(x_i, g(x_i, t_j^*)), i = 1, 2, \ldots, n\}$, where n is the size of the training set.
(2) Evaluate a criterion called the implausibility function over a large test set. For each $j = 1, \ldots, k$, the implausibility criterion is defined as:

$$IM_j(x) = \frac{|\ \hat{g}(x, t_j^*) - g_0(t_j^*)\ |}{s_{t_j^*}(x)},$$

where $\hat{g}(x, t_j^*)$ is the predicted response derived from the GP surrogate corresponding to the simulator response at time point t_j^* and $s_{t_j^*}(x)$ is the associated uncertainty. From the test set, points are deemed implausible if $IM_{max}(x) > c$, where c is the predetermined cutoff chosen in an ad hoc manner and

$$IM_{max}(x) = \max\{IM_1(x), IM_2(x), \ldots, IM_k(x)\}.$$

Points in the complement set are said to be plausible.

(3) Select the plausible design points, augment it to the training set, and go to Step 1.
(4) At the end of the procedure, the approximate inverse solution is extracted from the training set or from the predicted response over a dense set via the final surrogate.

Bhattacharjee et al. [5] recommended a modification in the HM algorithm and used a small initial design as per the popular $n_0 = 10 \cdot d$ rule-of-thumb [19, 31] as compared to a large initial design. This helped in achieving the desired accuracy of the inverse solution with significantly fewer runs. However, the size of the training set in Bhattacharjee et al. [5] can still become very large very fast because the algorithms recommend choosing all plausible points in Step 3. For instance, their (Matlab-simulink) hydrological model example required 461 simulator runs for estimating the inverse solution. In this paper, we implement a subsampling strategy via clustering and then select only the cluster centers instead of all plausible points. This will ensure that the input space is explored thoroughly with much fewer training points. We follow this twofold modified HM algorithm for all simulations.

2.6 Saddlepoint Approximation-based EI

Zhang et al. [56] used SVD-based GP model originally developed by Higdon et al. [21] for fitting a surrogate to the time-series output $g(x)$ of a computer simulator. Although slightly more complicated, but here also, the predicted mean response and the associated uncertainty (i.e., mean square error) have closed-form expressions. Subsequently, the authors applied the EI criterion in Jones et al. [24] to $w(x) = \|g(x) - g_0\|$, i.e.,

$$E[I(x)] = E[(w_{min} - w(x))_+ | Data],$$

however, the expectation had to be computed with respect to the SVD-GP—the surrogate model for $g(x)$. Here, the authors proposed a saddlepoint approximation for computing $E[I(x)]$. They also developed an *R* package called `DynamicGP` which implements this methodology. The usage of the most important function called `saEI` is shown as follows:

```
saEIout = saEI(xi,yi,yobs,nadd,candei,candest,func,...,
nthread=4,clutype="PSOCK")
```

where `xi` and `yi` denote the initial training data, `yobs` is the target response, `nadd` is the number of follow-up points to be added, `candei, candest` are the test sets for optimizing the saEI criterion and extracting the inverse solution, respectively. Since the SVD-GP model fitting and saddlepoint approximation calculations are computationally intensive, parallel computing environment can also be used via specifying the number of threads (`nthread`) and cluster type (`clutype`).

3 Proposed Methodology

Most of the existing methodologies to solve the inverse problem for simulator with time-series response use the global minimization criterion by Jones et al. [24]. We

propose a methodology that is based on the usage of scalar-valued contour estimation criterion by Ranjan et al. [44] for the inverse problem under a limited budget constraint.

Similar to the HM algorithm, we discretize the simulator response at a DPS of size $k(\ll L)$ that aims to capture the important features of the target response series. However, instead of choosing the DPS via a subjective judgement, we propose using a systematic construction approach via regression spline approximation of the target series g_0. Subsequently, we propose to iteratively solve the k scalar-valued inverse problems using the efficient contour estimation method (outlined in Sects. 2.2 and 2.4). Finally, the desired inverse solution is obtained by taking the intersection of these k sets of inverse solutions. There are several parts of the proposed methodology that requires detailed discussion.

Construction of DPS Fit a (regression) cubic spline function to the target response series g_0 and then use the set of knot locations as the DPS. However, finding optimal set of knots is a classical yet challenging problem.

Two obvious approaches to address this issue are "simultaneous search" and "sequential search". The "simultaneous search" finds the best k-knot combination by simultaneously searching the k-dimensional time-point grid with $\binom{L}{k}$ options and optimize a goodness-of-fit criterion like mean-square error (MSE) for the fitted spline approximation. Subsequently, the optimal value of k, and the corresponding set of knots, can be obtained using the elbow method, where the MSE is plotted against the number of knots and the objective is to identify the elbow of the plot.

On the other hand, the alternative "sequential search" follows a greedy approach for constructing the DPS. The idea is similar to the construction of a regression tree, where the split-points are essentially the knot locations. That is, we start with no knots and find the best location for the first knot by minimizing the overall MSE as per the spline regression fit. The optimal location for the second knot is found by fixing the first knot location. Continuing further in this manner, the search for optimal location for the j-th knot assumes that the optimal location of the previous $(j-1)$ knots is known. Finally, the optimal number of knots is found using the elbow method. For implementation, the R package `splines` is called upon for this purpose, while the command `bs()` is used for finding B-spline basis functions in the linear model environment.

We quickly illustrate the sequential search scheme by applying it to a test function. Suppose the simulator outputs are generated via Easom function [35],

$$g(x, t_j) = \cos(x_1)\cos(x_2)\exp\big\{-(x_1 - \pi t_j)^2 - (x_2 - \pi)^2)\big\},$$

where t_j are L equidistant time points scaled in $[0, 1]$ for $j = 1, \ldots, L = 200$, and the input space is scaled to $(x_1, x_2) \in [0, 1]^2$. We select the target response g_0 corresponding to the input set $x_0 = (0.8, 0.2)$. Pretending that x_0 is unknown, the objective of the inverse problem would be to find $x = (x_1, x_2)$ such that $g(x, t_j) \approx g_0(t_j)$ for all $j = 1, \ldots, 200$.

The first element of the DPS is obtained by minimizing the MSE of the cubic spline fitted to the target response over each of the possible 200 time points as the sole knot. We found the optimal first knot at time point $t_1^* = 145$. Keeping the knot at time point $t_1^* = 145$ fixed, we repeated the process and tried the remaining 199 options and found the second optimal knot at time point $t_2^* = 37$. The process continued, and the

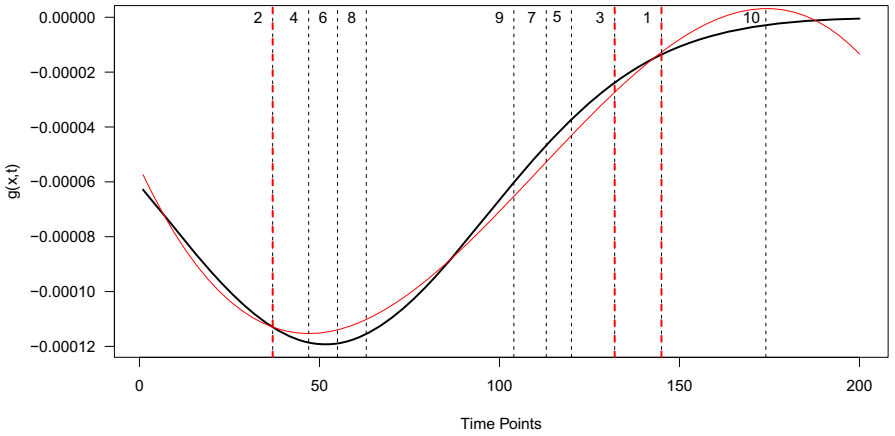


Fig. 1 Easom function: Black solid curve shows $g(x_0)$. The vertical dashed lines depict the ordered positioning of optimal knots for fitting cubic splines to the time series. Red curve shows the reconstructed response using DPS as knots

locations of ten optimal knot are {145, 37, 132, 47, 120, 55, 113, 63, 104, 174} (see Fig. 1).

In Fig. 1, we have illustrated the sequential selection of 10 knots; however, in reality, the required number of knots may be different. The elbow plot method investigates the relationship between MSE and the number of knots and finds the elbow of the plot, i.e., the second derivative reaches a positive value. This would allow for a good fit while maintaining the efficiency of the knots used. Figure 2 shows the corresponding "MSE vs. the number of knots function" plot for the Easom function. In this case, the elbow cutoff is 3. That is, the recommended discretization-point-set (DPS) for this time-series response would be {145, 37, 132}.

Remark 1 *Computational Cost:* Although more accurate than its competitors, the simultaneous search is computationally too expensive (dimension of the search space, $\binom{L}{k}$, grows exponentially for large k). As a result, it may be preferred to settle with a slightly suboptimal (but computationally tractable) set of knots, perhaps via minimizing the goodness-of-fit criterion (e.g., MSE) over a randomly chosen large subset of the L^k grid. Alternatively, one can use the sequential search method discussed above. For all inverse problem estimation examples considered in this paper, we have used the sequential search method for constructing DPS. Of course, in some cases, such a suboptimal method may require a few more discretization points in the DPS to reach the desired accuracy level as compared to the "simultaneous search" method. In Appendix, we presented a more detailed comparison of the computational costs.

For a quick reference, we compare the accuracy of the two search methods for Easom test function (see Fig. 1), where the target series is generated using $x_0 = (0.8, 0.2)$ with additional Gaussian noise. We fitted cubic-spline regression model to the target series with j knots identified using the two search methods. For finding optimal DPS of size j under the simultaneous search method, we followed a computationally cheaper

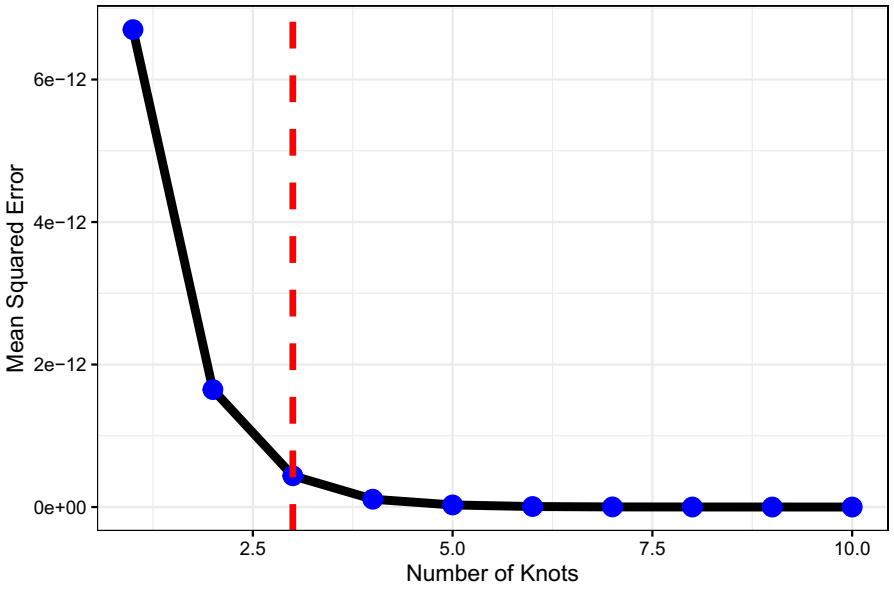


Fig. 2 Easom function: "Mean-squared error versus the number of knots" for 10 knots for spline regression added sequentially one at-a-time

approximation and randomly selected $200 \cdot j$ candidate points instead of fitting $\binom{200}{j}$ MLR models. Figure 3 compares the log(MSE) of the fitted models.

It is clear from Fig. 3 that the simultaneous search method (although computationally more expensive) provides slightly more accurate set of knots (i.e., DPS) for the initial values of j, but, eventually the sequential search scheme exhibits its superior performance. See Appendix for more comprehensive computational cost comparison.

Multiple Scalar-valued Contour Estimation (MSCE) After finding a reasonable DPS, we sequentially estimate k scalar-valued inverse solutions $S_j = \{x \in [0,1]^d : g(x, t_j^*) = g_0(t_j^*)\}$ for $j = 1, 2, ..., k$. Suppose our total simulator run budget is N, then, the process starts by choosing an initial design of size $n_0 (< N)$ from the input space $[0,1]^d$, for which we use a maximum projection Latin hypercube design [25]. The remainder of the budget $(N - n_0)$ is equally distributed in to k parts for estimating S_j, $j = 1, 2, \ldots, k$. That is, the first inverse problem would estimate S_1 using n_0-point initial design and $(N - n_0)/k$ follow-up trials chosen one at-a-time by maximizing the EI criterion (4) and updating the GP surrogate iteratively. The augmented data of size $n_0 + (N - n_0)/k$ are now treated as the initial training set for the second scalar-valued inverse problem. Thus, one would estimate S_j using the initial training set of size $n_0 + (j-1)(N - n_0)/k$, obtained after solving the previous $(j-1)$ scalar-valued inverse problems, and $(N - n_0)/k$ follow-up trials via EI optimization.

For the Easom function, since the DPS is of size three, we need to solve three scalar-valued inverse problems. We set a total training size budget of $N = 50$ points and initial design of size $n_0 = 15$. The budget of follow-up points, $N - n_0 = 35$, is divided approximately evenly for the three inverse problems (i.e., $35 = 12 + 12 + 11$). When

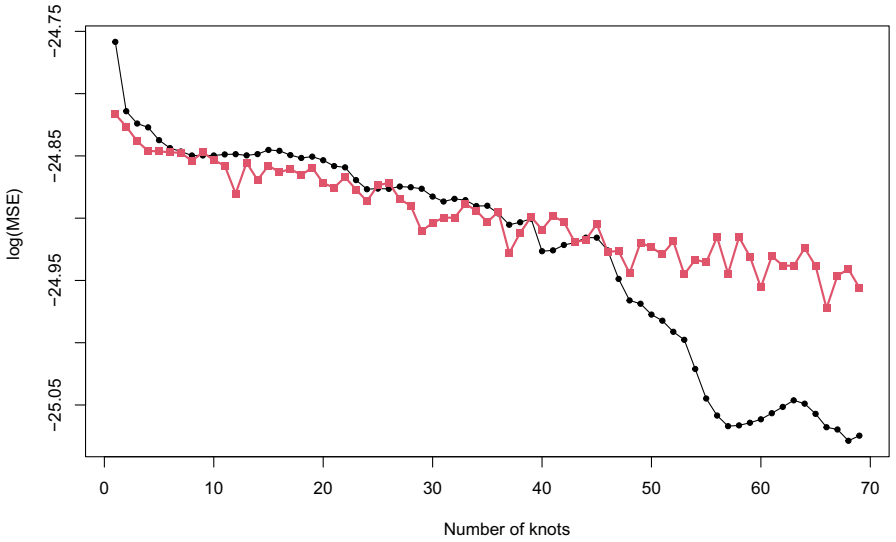


Fig. 3 Easom Function: log(MSE) comparison of splines fitted to the target response, with optimal knots found using two methods: simultaneous search (red squares) and sequential search (black dots)

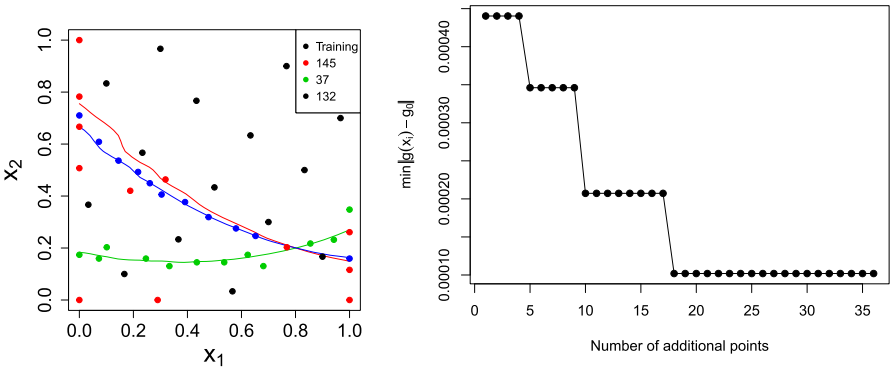


Fig. 4 Easom function: Training data is depicted by dots and the estimated contours are shown by solid curves. The black dots correspond to the initial design, whereas red, green, and blue dots represent the follow-up points obtained via EI optimization for the three scalar-valued contour estimation at $DPS = (145, 37, 132)$, respectively

computing the EI criterion, we set $\alpha = 0.67$ which corresponds to 50% confidence interval under normality. Furthermore, since the input space is only two-dimensional unit square, we use 5000-point random Latin hypercube designs for maximizing the EI criteria for sequentially adding follow-up trials. The left panel of Fig. 4 shows the three estimated contours along with selected follow-up points corresponding to $t_1^* = 145$ (in red), $t_2^* = 37$ (in green) and $t_3^* = 132$ (in blue). The right panel depicts the convergence over iterations as the follow-up points are added to the training data. The progress is measured by the minimum value of $\|g(x_i) - g_0\|$ over the training data.

From Fig. 4, it is clear that for the first contour estimation, more follow-up points focus on global exploration for better overall understanding of the process as compared to the local search for accuracy enhancement of the contour estimate. For the second and third contour estimations, the follow-up points tend to focus more and more on the local search. The second panel of Fig. 4 shows that a good approximation of the inverse solution was obtained after a few additional points were added for the second contour estimation problem.

Remark 2 *Sensitivity of the order of DPS:* In principle (i.e., theoretically), if all scalar-valued inverse problems have been solved accurately, then the overall inverse solution of the simulator with time-series response should also be estimated with high accuracy. However, in practice, it may be tempting to think that the order in which the three (in general, k) scalar-valued inverse problems are solved may affect the accuracy of the overall inverse problem for the underlying time-series-valued simulator. Our investigations based on the simulated examples considered in this paper show that the order does not play a significant role. For instance, Fig. 5 depicts the sensitivity of the order of DPS in the sequential contour estimation approach for the Easom function example. In Fig. 5, the point clouds represent S_1, S_2, S_3 and $\cap_{i=1}^{3} S_j$ in the order of black, red, blue, and yellow for different DPS sequences shown in the figure captions.

Of course, this demonstration based on finitely many examples does not guarantee that the order will not matter for every MSCE implementation of an inverse problem for time-series-valued simulators. If the fitted surrogates at each t_i^* are adequate to find the true inverse solution, then clearly the ultimate inverse solution found at the end should not differ.

Extraction of the Overall Inverse Solution S_0: We approximate S_0 with $\cap_{j=1}^{k} S_j$—the intersection of k scalar-valued inverse solutions obtained at the discretization-point-set (DPS). If there exists a solution of the underlying inverse problem for the dynamic (time-series-valued) simulator, then $\cap_{j=1}^{k} S_j$ will be nonempty. The following result establishes the existence of the inverse solution as per the proposed approach.

Theorem 1 *Let* $S_0 = \{x \in \chi : g(x, t_j) = g_0(t_j),\ j = 1, 2, \ldots, L\}$ *be the true inverse solution for a time-series-valued simulator* $g(x)$ *with respect to* g_0, *and* $S_j = \{x \in \chi : g(x, t_j^*) = g_0(t_j^*)\}$ *be the inverse solution at the* j*-th DPS point* t_j^*, *then,* $S_0 \subseteq \cap_{j=1}^{k} S_j$.

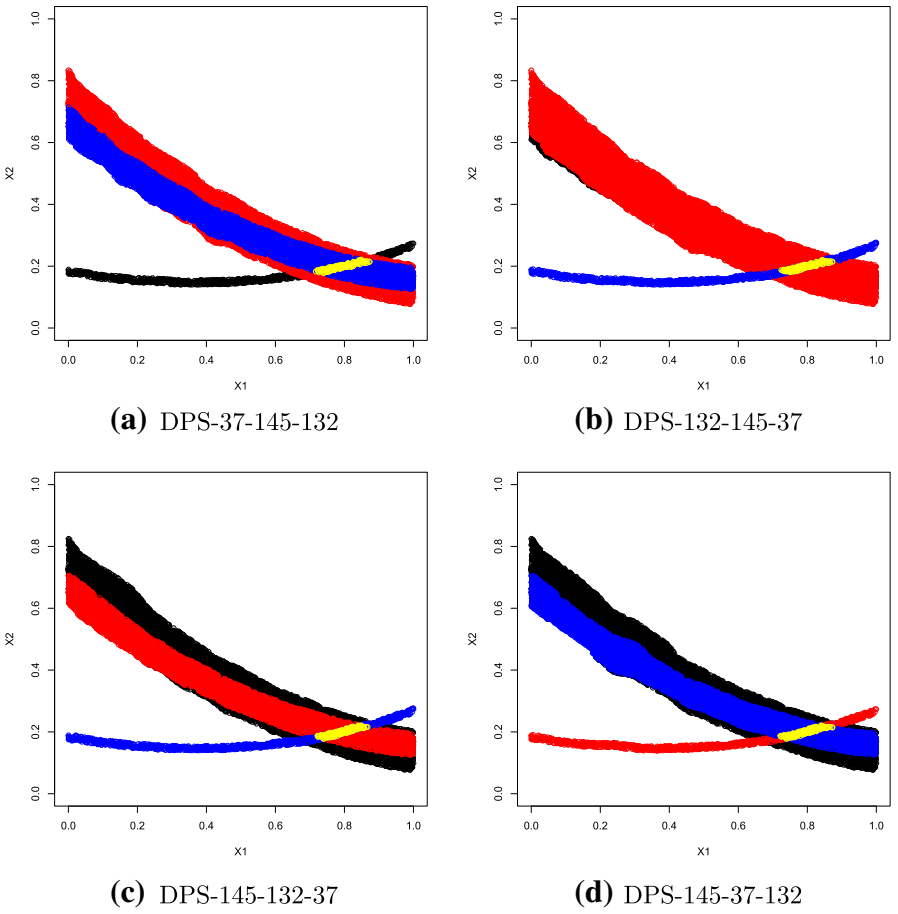
Corollary 1 *If* $\cap_{j=1}^{k} S_j$ *represents a single cluster, then* S_0 *is unique.*

Practical Implementation If $\cap_{j=1}^{k} S_j$ generates multiple distinct clusters, then either the underlying inverse solution has multiple inverse solutions or we have detected some false solution along with the correct solution. This can be further ascertained by increasing the size of DPS and follow the proposed approach.

We have omitted the proof of Theorem 1, as it is straightforward and not giving additional insights to this discussion.

The desired inverse solution would be

$$\hat{x}_{opt} = argmin\{\|g(x) - g_0\|, x \in \cap_{j=1}^{k} S_j\}.$$



(a) DPS-37-145-132 **(b)** DPS-132-145-37

(c) DPS-145-132-37 **(d)** DPS-145-37-132

Fig. 5 Easom Function: Comparison of the inverse solution estimate as per the proposed MSCE method, with $n_0 = 15$ and $N = 50$ points

To implement this, we obtain k GP surrogates $\hat{g}(x, t_j^*)$, after the final iteration, using all N training points found in the due process of estimating k contours. Instead of the exact match, we accept the approximate inverse solutions as $S_j(\delta) = \{x : |\hat{g}(x, t_j^*) - g_0(t_j^*)| < \delta\}$ for some small δ, for each time point t_j^* in DPS. This accounts for the round off errors and other approximations made during the implementation. This tolerance δ has to be judiciously chosen to accurately estimate the inverse solution set.

For Easom function example, it is clear from Figs. 4 and 5 that the three contours intersect on a common point. Here, we set $\delta = 10^{-5}$, and the final inverse solution obtained is $\hat{x}_{opt} = (0.8188, 0.2029)$. Figure 6 shows that the simulator response at $\hat{x}_{opt}$ (blue dashed curve) is virtually indistinguishable as compared to the target response (black solid curve).

Algorithm 1: Multiple scalar-valued contour estimation (MSCE) approach

Input : (1) Input parameters: d, L, n_0, N
(2) time-series-valued computer simulator: $\{g(x, t_j),\ j = 1, ..., L\}$
(3) Target response: $\{g_0(t_j),\ j = 1, \ldots, L\}$
(4) Tolerance: δ

Output: (1) Final training set: $\mathtt{xx}_{N\times d}$ and $\mathtt{yy}_{L\times N}$
(2) Estimated inverse solution: $\hat{\mathtt{x}}_{opt}$

1 Construct a DPS of size $k(\ll L)$ that would capture the important features of the target time-series response, say, $(t_1^*, t_2^*, \ldots, t_k^*)$. See Sect. 3 for the proposed regression spline-based methodology.
2 Choose n_0 points in $\chi = [0, 1]^d$ using a maximum projection Latin hypercube design. Obtain the corresponding simulator response matrix $Y_{L\times n_0}$.
3 **for** $j = 1, \ldots, k$ **do**
4 Use contour estimation method for scalar-valued simulator to estimate $S_j(x) = \{x \in \chi\ :\ g(x, t_j^*) = g_0(t_j^*)\}$. Assume the size of initial design is $n_0 + (j-1)\cdot(N - n_0)/k$, whereas $(N - n_0)/k$ follow-up trials are added sequentially one at-a-time as per the EI criterion in Sect. 2.4.
5 Augment the follow-up points to the initial design for the $(j + 1)$-th scalar-valued inverse problem.
6 Fit final k GP surrogates to $g(x, t_j^*)$ using all N training points. Obtain $S_j = \{x : |\hat{g}(x, t_j^*) - g_0(t_j^*)| < \delta\}$ and $U_j = \{x : |\hat{g}(x, t_j^*) - g_0(t_j^*)| < \alpha s(x, t_j^*)\}$ for $j = 1, 2, \ldots, k$.
7 Extract the final inverse solution as $\hat{x}_{opt} = argmin\{\|g(x) - g_0\|, x \in \cap_{j=1}^k S_j\}$, and report the spread of $\cap_{j=1}^k U_j$ as the associated uncertainty measure.

This notion of extracting the inverse solution via $\cap_{j=1}^k S_j$ can be further extended to quantify the uncertainty in the inverse solution estimate. In spirit of the formulation of the improvement function in Sect. 2.4, define

$$U_j = \{x \in \chi : |\hat{g}(x, t_j^*) - g_0(t_j^*)| < \alpha s(x, t_j^*)\},\ j = 1, 2, \ldots, k,$$

where $\hat{g}$ is obtained from the final fit. Assuming S_0 is unique, and $\cap_{j=1}^k S_j$ is non-empty, the spread of $\cap_{j=1}^k U_j$ can be taken as a measure of uncertainty in estimating the inverse solution. It is intuitive to infer that the spread of $\cap_{j=1}^k U_j$ will converge to

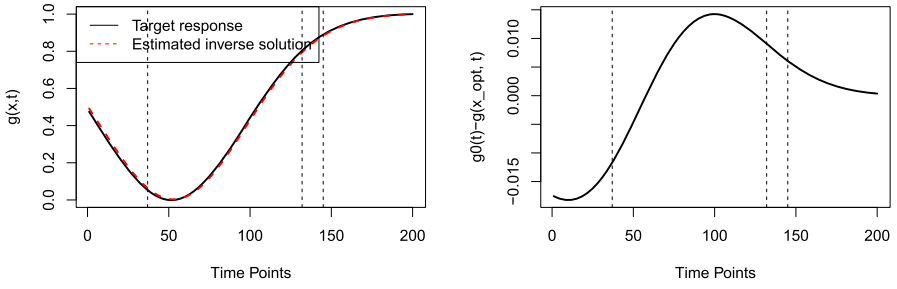


Fig. 6 Easom function: The left panel shows the target response in black solid line and the simulator response at x_{opt} is shown by the dashed red curve. The right panel presents the difference $g_0(t) - g(\hat{x}_{opt}, t)$

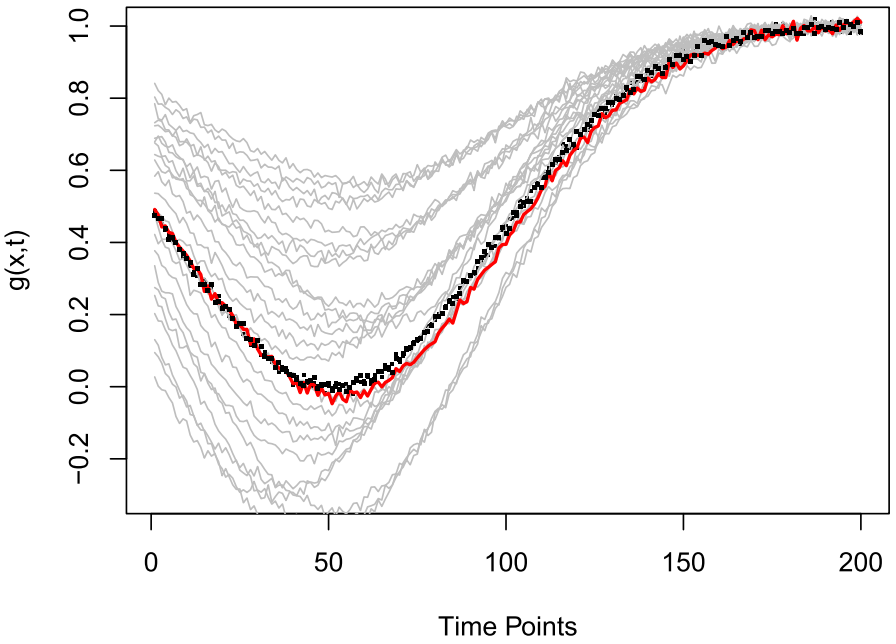


Fig. 7 Easom function: Black dots represent the target response, the red curve shows the best estimate of the inverse solution, and the gray curves show the responses corresponding to $\cap_{j=1}^{k} U_j$

zero as the size of the training data increase to infinity. Here spread($\cap_{j=1}^{k} U_j$), later abbreviated as spread(U_j) in Tables 1 and 2, is equal to $\sum_{r=1}^{d} Var(x_r)$, where x_r is the vector of rth coordinate from the estimated inverse solution $\cap_{j=1}^{k} U_j$. Note that $U_j = S_j(\delta)$ for $\delta = \alpha s(x, t_j^*)$.

We summarize the key steps of the proposed MSCE approach in Algorithm 1.

Approximate Inverse Solution If the simulator output and/or the target response are noisy, then the exact match for the inverse solution would not exist, and hence, the approximation using $\cap_{j=1}^{k} S_j(\delta)$ and $\cap_{j=1}^{k} U_j$ are viable options to obtain the closest possible inverse solution. The target response in our hydrological model is noisy (see Sect. 5). For a quick illustration, we introduce a random Gaussian noise term in the Easom simulator output (i.e., the time-series response is $g(x, t) + \varepsilon(t)$) and the target response corresponds to the same $x_0 = (0.8, 0.2)$, $DPS = (145, 37, 132)$ and $n_0 = 15$ and $N = 30$. Figure 7 presents the simulator responses corresponding to $x \in \cap_{j=1}^{k} U_j$ and the best estimate of the inverse solution.

Given that the simulator returns noisy output, the final estimate appears to be a reasonably good approximation of the desired inverse solution.

4 Simulation Studies

In this section, we use three different test function-based time-series-valued simulators to compare the performance of the proposed method with the modified history matching (HM) algorithm [5, 53], the naive scalarization method [46], and the saddlepoint

approximation-based EI (saEI) approach [56]. For performance comparison between the four methods, we use three popular goodness-of-fit measures called R^2, RMSE, normD, and the uncertainty measure proposed in Sect. 3 (i.e., the spread of $\cap_{j=1}^{k} U_j$). The objective would be to maximize R^2 and minimize RMSE, normD, and the spread of $\cap_{j=1}^{k} U_j$.

- Root-mean-squared error given by

$$RMSE = \left(\frac{1}{L} \sum_{j=1}^{L} \mid g(\hat{x}_{opt}, t_j) - g_0(t_j) \mid^2 \right)^{1/2}$$

  measures the discrepancy between the simulator response at the estimated inverse solution $\hat{x}_{opt}$ and the target response series g_0.
- Coefficient of determination R^2 of the simple linear regression model fitted to the estimated inverse solution and the target response, i.e., R^2 of the following linear regression model

$$g_0(t_j) = g(\hat{x}_{opt}, t_j) + \psi_j, \; j = 1, 2, \ldots, L,$$

  with the assumption of i.i.d. error ψ_j.
- Normalized discrepancy (on log-scale), between the simulator response at the estimated inverse solution and the target response

$$normD = \log \left(\frac{\left\| g_0 - g\left(\hat{x}_{opt}\right) \right\|_2^2}{\| g_0 - \bar{g}_0 1_L \|_2^2} \right)$$

  where $\bar{g}_0 = \sum_{t=1}^{L} g_0(t)/L$ and 1_L is an L-dimension vector of ones. Note that $1 - \exp(normD)$ is a popular goodness-of-fit measure and often referred to as Nash–Sutcliffe efficiency [37].

For each test function, 100 replications were run for different initial training data obtained via maxPro LHD [25], random test sets for optimizing the follow-up criteria, and candidate sets for extracting the inverse solutions. We also prefixed the target series (consequently the DPS) and the n_0 and N combination. Note that the implementation of scalarization method, saEI and MSCE, requires sequential augmentation of one follow-up trial at-a-time by maximizing some criteria, and hence, prefixing the initial design size (n_0) and a total budget (N) is in sync with these three methods. However, the (modified) HM approach accumulates follow-up points in batches, and thus, the total runsize will vary slightly with the initial design and/or the test sets.

4.1 Example 1: Easom Function [35] contd.

We begin by revisiting the illustrative example discussed in Sect. 3 to compare the inverse solutions arrived at by the four methods. Recall that the initial design size

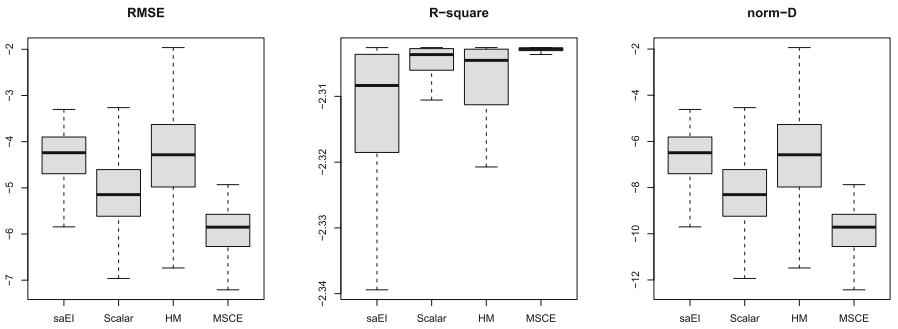


Fig. 8 Easom Function: GOF comparison between the saEI method, scalarization method, modified HM algorithm, and the proposed MSCE approach

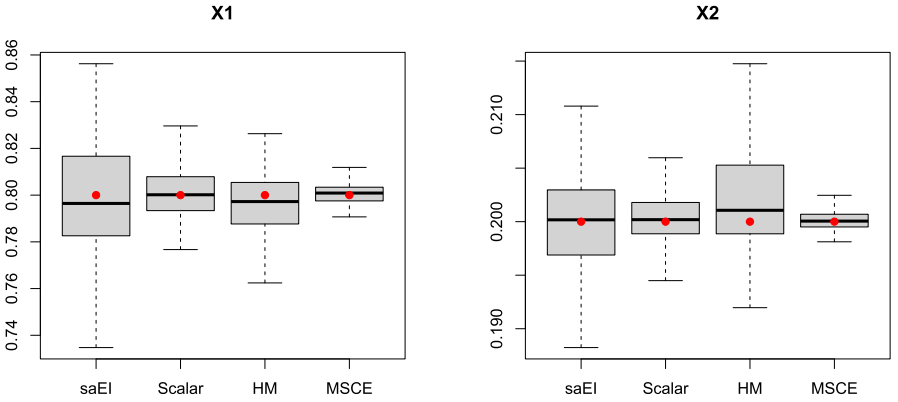


Fig. 9 Easom Function: Distribution of $\hat{x}_{opt}$: comparison between the saEI, scalarization method, HM algorithm, and the MSCE approach. Red dot represents the true x_0 that generated g_0

is set to $n_0 = 15$, the total runsize to $N = 50$, the target series corresponds to $x_0 = (0.8, 0.2)$ which led to $DPS = (145, 37, 132)$. Figure 8 compares the three goodness-of-fit (GOF) measures (R^2, RMSE and norm-D) for all four methods over different replications.

For better visual comparison, we have depicted the distributions of $\log(RMSE)$, $\log(normD)$ and $\log(R^2 - 0.9)$. It is evident from Fig. 8 that MSCE outperforms the other competitors by a big margin with respect to all GOF measures. We also compared the accuracy of estimated inverse solutions over the replications (see Fig. 9).

The left and right panels of Fig. 9 show the boxplots of $\hat{x}_{opt}$ for x_1 and x_2, respectively, for the four competing methods. The larger the boxplots, the bigger the uncertainties associated with the corresponding methods. Figure 9 shows that all methods are able to estimate the inverse solution, but the proposed method MSCE does it more accurately (i.e., the variation is smallest around the true value) as compared to the other competitors. As an alternative means of uncertainty quantification (UQ), we computed the total dispersion of $\cap_{j=1}^{k} U_j$ for all methods in different replications,

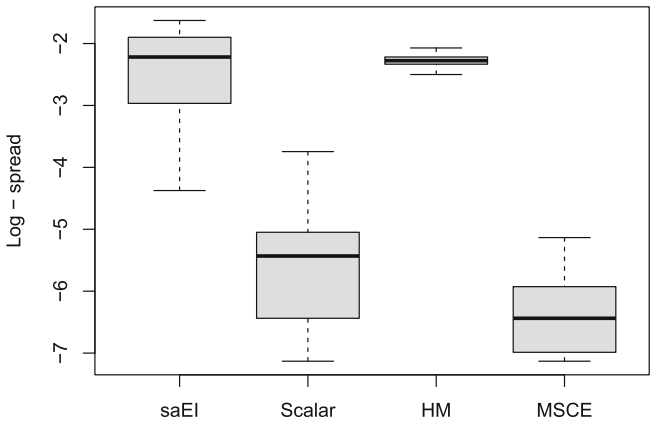


Fig. 10 Easom Function: Distribution of the log-spread of $\cap_{j=1}^{k} U_j$- UQ in the estimate of the inverse solution. Comparison between the saEI, scalarization method, HM algorithm, and the MSCE approach

Table 1 Performance comparison of the four methods (saEI, Scalarization, HM and the proposed MSCE) with respect to four goodness-of-fit measures: R^2, RMSE, norm-D, and spread of $\cap_{j=1}^{k} U_j$ (presented in log-scale to highlight the difference)

Methods	saEI	Scalar	HM	MSCE
(Modified) Levy function ($d = 2$)				
Spread(U_j)	−2.1 (0.15)	−2.5 (0.37)	−2.1 (0.27)	−3.3 (0.19)
RMSE	−8.4 (1.3)	−10.2 (0.88)	−10.7 (0.92)	−12.0 (0.59)
R-squared	−0.07 (0.13)	-0.003 (0.014)	−0.003 (0.016)	-3×10^{-5} (6×10^{-5})
Norm-D	−1.1 (2.7)	−4.7 (1.76)	−5.76 (1.84)	−8.9 (1.2)
Harari and Steinberg [20] function ($d = 3$)				
Spread(U_j)	−2.2 (0.73)	−2.02 (0.76)	−1.73 (0.40)	−3.2 (0.91)
RMSE	−4.4 (0.53)	−5.23 (0.60)	−4.76 (0.72)	−6.1 (0.41)
R-squared	−1.2 (0.37)	−0.77 (0.10)	−0.86 (0.32)	−0.7 (0.005)
Norm-D	−2.0 (1.05)	−3.68 (1.19)	−2.74 (1.45)	−5.5 (0.82)
Bliznyuk et al. [7] function ($d = 5$)				
Spread(U_j)	−0.7 (0.027)	−0.79 (0.05)	−0.61 (0.027)	−0.85 (0.026)
RMSE	−5.8 (0.52)	−5.77 (0.47)	−6.55 (0.51)	−7.47 (0.419)
R-squared	−0.7 (0.019)	−0.72 (0.044)	−0.70 (0.0052)	−0.69 (0.0006)
Norm-D	−3.8 (1.04)	−3.73 (0.94)	−5.29 (1.02)	−7.17 (0.84)

The numbers in parentheses denote the standard error

and the results are summarized in Fig. 10. The lower the boxplots are located on the y-axis, the better the methods perform.

As per this UQ measure as well, we can see that the proposed method gives the most accurate results. Interestingly, the HM method gives consistently inaccurate results.

4.2 More Test Function-Based Examples

In this section, we compare the performance of the four methods via several test function-based time-series-valued computer simulators. All results are averaged over 100 replications. The test functions are listed as follows:

(1) Levy function: The original Levy function by Laguna and Marti [30] produces scalar response for an arbitrary input dimension d. We have modified the test function to generate time-series outputs. For $d = 2$, let

$$y_t(x) = sin(\pi t)^2 + \left[\left(\frac{t}{5} - 1\right)^2 (1 + 10(sin(0.5\pi t + 1))^2)\right] \\ *[(w_1 - 1)^2(1 + 10(sin(\pi w_1 + 1))^2)] \\ + (w_2 - 1)^2(1 + (sin(2\pi w_2))^2),$$

where $w_i = 1 + (x_i - 1)/4$, and $x_i \in (-10, 10)$. For the simulation study in this paper, we have fixed $n_0 = 15$, $N = 45$, $x_0 = (0.5, 0.5)$ and $DPS = (40, 110, 170)$.

(2) Harari and Steinberg [20]: The simulator takes $d(= 3)$-dimensional inputs $x = (x_1, x_2, x_3) \in [0, 1]^3$ and produces time-series response as per

$$y_t(x) = \exp(3x_1 t + t) \times \cos(6x_2 t + 2t - 8x_3 - 6)$$

where $t \in [0, 1]$ on a 200-point equidistant grid. We assumed $x_0 = (0.522, 0.95, 0.427)$ (drawn randomly) for generating the target series and found $DPS = (118, 26, 95)$ as per the algorithm outlined in Sect. 3. Furthermore, the simulation study was conducted with the initial design size of $n_0 = 20$ and a total budget of $N = 50$.

(3) Bliznyuk et al. [7] presents an environmental model which simulates a pollutant spill caused by a chemical accident. Here, the input space is $x = (x_1, x_2, x_3, x_4, x_5)^T \in [7, 13] \times [0.02, 0.12] \times [0.01, 3] \times [30.01, 30.304] \times [0, 3]$, and the simulator outputs are generated as:

$$y_t(x) = \frac{x_1}{\sqrt{x_2 t}} \exp\left(\frac{-x_5^2}{4x_2 t}\right) + \frac{x_1}{\sqrt{x_2(t - x_4)}} \exp\left(\frac{-(x_5 - x_3)^2}{4x_2(t - x_4)}\right) I(x_4 < t)$$

with $t \in [35.3, 95]$ defined over a 200-point equidistant grid. The target time-series response corresponds to $x_0 = (9.640, 0.059, 1.445, 30.277, 2.520)^T$ (randomly chosen). The corresponding DPS turns out to be $(30, 7, 61, 14)$ and the simulation study assumed $n_0 = 30$ and $N = 90$.

It is clear from Figs. 8, 9 and 10 and Table 1 that the proposed MSCE method significantly outperforms the three competitors (saEI, Scalar and HM methods) with

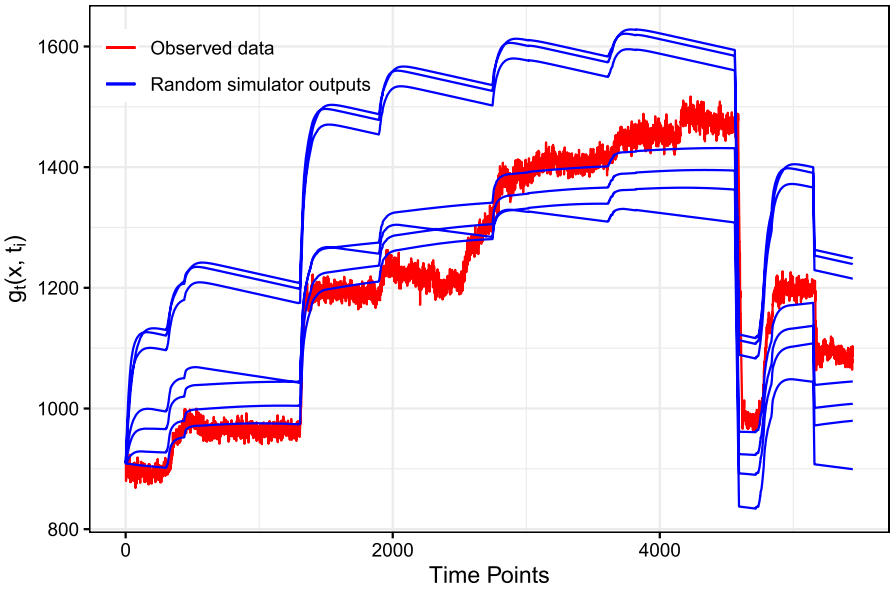


Fig. 11 Hydrological model: Target response g_0 along with a few random simulator outputs observed over 5445 time-points

respect to all four goodness-of-fit (GOF) criteria for four test function-based simulators ranging from $d = 2$ to $d = 5$. Once again recall that the objective is to maximize R^2 whereas minimize the other three GOF measures.

5 Real Application: Rainfall–Runoff Example

The motivating application in Bhattacharjee et al. [5] used a hydrological simulator − Matlab-Simulink model introduced by Duncan et al. [11]—to study the rainfall–runoff relationship for a windrow composting pad. The following four parameters have been identified as the inputs with most significant influence on the output: depth of surface, depth of subsurface and two coefficients of the saturated hydraulic conductivity (K_{sat1} and K_{sat2}). Interested readers can see Duncan et al. [11] for further details on the hydrological model. For the inverse problem, the target response is the rainfall–runoff data (g_0) observed from the Bioconversion center at the University of Georgia, Athens, USA. Figure 11 depicts the observed target response and a few random outputs from the hydrological model.

It is clear from Figure 11 that the target response appears to be noisier than the simulator response, and a little biased as well. The optimal knots in the regression spline approximation of the target response led to $DPS = \{4557, 3359, 4702, 4085\}$. It is important to observe that three of the time-points in DPS are between $t = 4000$ and $t = 5000$—the region with a big sudden dip. Although this clustering behavior is different from the earlier examples, it may be expected as this drastic change in the nature of the target series overpowers small variations in the other region. We

Table 2 Hydrological model: Goodness-of-fit comparisons of the proposed MSCE methods with the modified HM and scalarization methods

Methods	Scalar	HM	MSCE
spread(U_j)	0.3053	0.2892	0.2860
RMSE	67.06	64.69	53.96
R-squared	0.8824	0.8888	0.9314
norm-D	0.1225	0.1140	0.0793

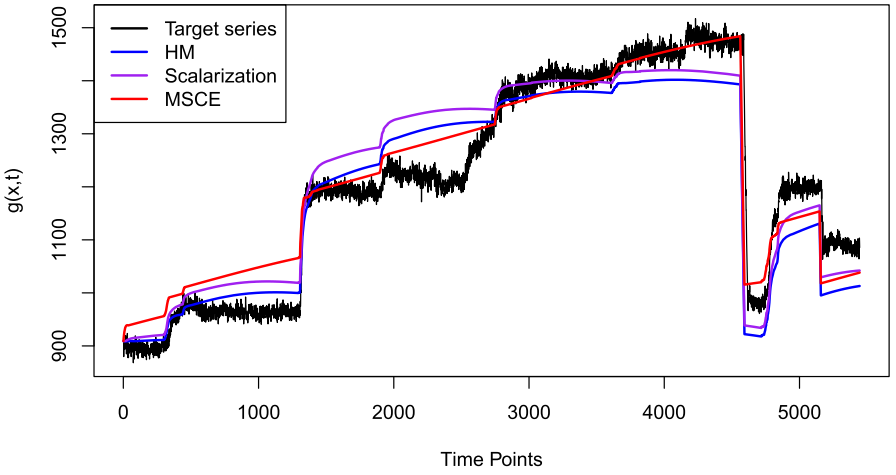


Fig. 12 Hydrological model: Black curve shows the target response, and the estimated inverse solutions corresponding to the modified HM method are shown in blue, MSCE by the red curve, and scalarization method by the purple curve

implement the proposed MSCE approach with $n_0 = 40$-point maxPro Latin hypercube design as an initial design and added additional 10 follow-up points. The results are compared with the modified HM approach and the scalarization method. The saEI approach could not be implemented here, because the R package DynamicGP required passing the computer simulator function, which we did not have access to in the required format. Table 2 summarizes the GOF results.

The results shown in Table 2 are consistent with the trends from the test function-based simulators (in Table 1). That is, the proposed MSCE approach outperforms the other competitors in terms of finding the closest match for the observed runoff data with respect to all four metrics. This is also evident from the visual comparison (Fig. 12) of the simulator responses corresponding to the estimated inverse solutions by different methods.

6 Concluding Remarks and Future Research

In this paper, we have proposed a new MSCE approach of solving the inverse problem for time-series-valued computer simulators by first carefully selecting a handful of time-points for discretizing the target response series (called the DPS) and then

iteratively solve multiple scalar-valued inverse problems at the DPS using the popular sequential algorithm via expected improvement approach developed by Ranjan et al. [44]. The final inverse solution for the underlying dynamic simulator is obtained via the intersection of all scalarized inverse solutions. In this paper, we have suggested using a natural cubic spline-based method for systematically finding the DPS. Based on the our simulation study using several test functions and a real-life hydrological simulator, it is clear that the proposed MSCE method outperforms three competing methods: scalarization technique [46], modified HM algorithm [5], and saEI method [56]. Although we do not have any theoretical justification yet, an intuitive explanation could be that (a) saEI uses saddlepoint approximation, which may not be very accurate; (b) the scalarization method uses GP as a surrogate for the Euclidean distance between the target response and the simulator runs at all time-points, which becomes non-stationary around the inverse solutions, and hence could be a source of inaccuracy; (c) the twofold modification of the original HM method adopted in this paper may have made it less efficient. In contrast, the proposed method carefully selects the DPS and then uses one of the most efficient EI criterion for iteratively solving the inverse problem.

There are a few important remarks worth mentioning. (1) When finding an optimal DPS using spline-based technique, we followed a greedy "forward variable selection"-type approach and identified one best knot at-a-time. The "best selection"-type approach may lead to a better solution; however, it is computationally expensive (seemingly impractical) in finding the best DPS. (2) For solving the scalar-valued inverse problems at the j-th element of the DPS, we took the size of the initial design be $n_0 + (j-1)\cdot(N-n_0)/k$ and budget of follow-up points is $(N-n_0)/k$. Based on our preliminary simulation study, we found no significant improvement in accuracy by changing the order of DPS for solving the scalar-valued inverse problems. We divided the follow-up point resources $N-n_0$ equally among the k scalar inverse problems; however, an efficient distribution of total budget N can be further investigated. (3) A recent paper [51] proposes a new Bayesian methodology for solving the inverse problem for time-series-valued simulators. It would be interesting to compare the performance of our proposed frequentist MSCE approach with their Bayesian optimization technique with computationally expensive integrands. (4) Since the responses are time series in nature, one can investigate including time-correlation structure in the steps of MSCE, for instance, the surrogates at multiple t_j^*, for improved efficiency. (5) This paper assumes the existence of the inverse solution. Although the proposed methodology gives approximate solution in the presence of small noise, further research is required to find the best approximation of the inverse solution if it does not exist in the search space.

Acknowledgements We would like to thank the Editor, the Guest Editor and the three referees for their helpful comments and suggestions which led to significant improvement of the paper.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A: Cost of Constructing Optimal DPS

Suppose we need to construct the DPSs of size $j = 1, 2, \ldots, k$, and the target series has been observed over 200 time points. Then, the costs of constructing these DPSs using the two methods are as follows.

Sequential search: The first optimal knot can be found by fitting 200 different multiple linear regression (MLR) models with 4+1 regression coefficients each (4 for the cubic polynomial and 1 for the knot location term) and then comparing the goodness-of-fit criterion (e.g., MSE or R^2_{adj}). The second optimal knot, given the first one is already known, can be found by fitting 199 different MLR models with 4+2 coefficients each, and so on. That is, in total, for sequentially finding k optimal knots using this method, one needs to fit $\sum_{j=0}^{k-1}(200 - j) = 200k - k(k-1)/2$ different MLR models. In terms of computational complexity, the total cost would be

$$\sum_{j=1}^{k}(200 - (j-1)) \cdot O((4+j)^3),$$

where $O((4+j)^3$ represents the computational cost of fitting a cubic-spline regression model to the target series with j knots.

Simultaneous search: Here, the cost is heavily controlled by the resolution of the search grid, and how exhaustive the search is. For consistency, we find the one-knot optimal set in the exact same manner as in the "sequential search" method, i.e., search the optimal knot over a 200-point grid. If we use the same 200-point grid, then we would have to fit $\binom{200}{j}$ MLR models for finding optimal DPS with j knots. That is, the total cost of constructing optimal DPS sets of size $j = 1, 2, \ldots, k$ would be

$$\sum_{j=1}^{k}\binom{200}{j} \cdot O((4+j)^3).$$

Since $\binom{200}{j}$ grows very rapidly with j, we follow a computationally cheaper approximation and randomly selected $200 \cdot j$ candidate points for estimating the optimal DPS of size j. This is clearly much greater than the cost associated with the sequential search method.

Undoubtedly, the sequential search method does not guarantee the global optimum and may give a suboptimal estimate of the DPS. However, the sequential method will eventually iterate through all time-points; the accuracy of DPS will increase to the maximum achievable level.

References

1. Azzimonti D, Ginsbourger D, Chevalier C, Bect J, Richet Y (2021) Adaptive design of experiments for conservative estimation of excursion sets. Technometrics 63(1):13–26

2. Bayarri MJ, Berger JO, Calder ES, Dalbey K, Lunagomez S, Patra AK, Pitman EB, Spiller ET, Wolpert RL (2009) Using statistical and computer models to quantify volcanic hazards. Technometrics 51:402–413
3. Bect J, Ginsbourger D, Li L, Picheny V, Vazquez E (2012) Sequential design of computer experiments for the estimation of a probability of failure. Stat Comput 22:773–793
4. Bichon BJ, Eldred MS, Swiler LP, Mahadevan S, McFarland JM (2008) Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J 46(10):2459–2468
5. Bhattacharjee NV, Ranjan P, Mandal A, Tollner EW (2019) A history matching approach for calibrating hydrological models. Environ Ecol Stat 26(1):87–105
6. Bingham D, Ranjan P, Welch WJ (2014) Design of computer experiments for optimization, estimation of function contours, and related objectives. Statistics in Action: A Canadian Outlook, 109
7. Bliznyuk N, Ruppert D, Shoemaker C, Regis R, Wild S, Mugunthan P (2008) Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. J Comput Graph Stat 17(2):270–294
8. Brown JL, Hund LB (2018) Estimating material properties under extreme conditions by using Bayesian model calibration with functional outputs. J R Stat Soc: Ser C (Appl Stat) 67(4):1023–1045
9. Cao S et al (2021) Determining the jet transport coefficient $\hat{q}$ from inclusive hadron suppression measurements using Bayesian parameter estimation. Phys Rev C 104(2):024905
10. Cole D, Gramacy R, Leser W (2021) Entropy-based adaptive design for contour finding and estimating reliability. J Qual Technol. https://doi.org/10.1080/00224065.2022.2053795
11. 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
12. Fang K, Li RZ, Sudjianto A (2006) Design and modeling for computer experiments. Computer Science and Data Analysis Series. Taylor & Francis Group, LLC, London
13. Forrester AIJ, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. Proc R Soc Lond A: Math Phys Eng Sci 463:3251–3269
14. Gration A, Wilkinson M (2019) Dynamical modelling of dwarf spheroidal galaxies using Gaussian-process emulation. Mon Notices R Astron Soc 485(4):4878–4892
15. Gramacy RB (2016) laGP: large-scale spatial modeling via local approximate Gaussian processes in R. J Stat Softw 72(1):1–46
16. Gramacy RB, Apley DW (2015) Local Gaussian process approximation for large computer experiments. J Comput Graph Stat 24(2):561–578
17. Gramacy RB, Lee HKH (2008) Bayesian treed Gaussian process models with an application to computer modeling. J Am Stat Assoc 103:1119–1130
18. Gramacy RB (2020) Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. CRC Press
19. Harari O, Bingham D, Dean A, Higdon D (2018) Computer experiments: prediction accuracy, sample size and model complexity revisited. Stat Sin 28:899–919
20. Harari O, Steinberg DM (2014) Convex combination of Gaussian processes for Bayesian analysis of deterministic computer experiments. Technometrics 56(4):443–454
21. Higdon D, Gattiker J, Williams B, Rightley M (2008) Computer model calibration using high-dimensional output. J Am Stat Assoc 103:570–583
22. Jala M, Levy-Leduc C, Moulines E, Conil E, Wiart J (2016) Sequential design of computer experiments for the assessment of fetus exposure to electromagnetic fields. Technometrics 58(1):30–42
23. Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. J Stat Plan Inference 26(2):131–148
24. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13(4):455–492
25. Joseph VR, Gul E, Ba S (2015) Maximum projection designs for computer experiments. Biometrika 102(2):371–380
26. Kaufman C, Schervish MJ, Nychka DW (2008) Covariance tapering for likelihood-based estimation in large spatial data sets. J Am Stat Assoc 103(484):1545–1555
27. Kennedy M, O'Hagan A (2002) Bayesian calibration of computer models. J R Stat Soc Ser B (Statistical Methodology) 63(3):425–464
28. Kennedy JC, Henderson DA, Wilson KJ (2020) Multilevel emulation for stochastic computer models with application to large offshore wind farms, arXiv:2003.08921

29. Krityakierne T, Baowan D (2020) Aggregated GP-based optimization for contaminant source localization. Oper. Res. Perspect. 7:100151
30. Laguna M, Marti R (2002) Experimental testing of advanced scatter search designs for global optimization of multimodal functions. Retrieved June 2013, from http://www.uv.es/rmarti/paper/docs/global1.pdf
31. Loeppky J, Sacks J, Welch W (2009) Choosing the sample size of a computer experiment: a practical guide. Technometrics 51:366–376
32. Lukemire J, Xiao Q, Mandal A, Wong WK (2021) Statistical analysis of complex computer models in astronomy. Eur Phys J Spec Top 230:2253–2263. https://doi.org/10.1140/epjs/s11734-021-00204-y
33. Mandal A, Ranjan P, Wu CJF (2009) G-SELC: optimization by sequential elimination of level combinations using genetic algorithms and Gaussian processes. Ann Appl Stat 3(1):398–421
34. 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(i12)
35. Michalewicz Z (1996) Genetic algorithms+data structures 1/4 evolution programs. Springer, Berlin
36. Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. J Stat Plan Inference 43(3):381–402
37. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I. A discussion of principles. J Hydrol 10(3):282–290
38. Oakley J (2004) Estimating percentiles of uncertain computer code outputs. Appl Stat 53(1):83–93
39. Oberpriller J, Cameron DR, Dietze MC, Hartig F (2021) Towards robust statistical inference for complex computer models. Ecol Lett 24:1251–1261
40. Perdikaris P, Karniadakis GE (2016) Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. J R Soc Interface 13(118):20151107
41. Perrin G (2020) Adaptive calibration of a computer code with time series output. Reliab Eng Syst Saf 196:106728
42. Picheny V, Ginsbourger D, Roustant O, Haftka RT, Kim N-H (2010) Adaptive designs of experiments for accurate approximation of target regions. J Mech Des 132(7)
43. Pratola M, Harari O, Bingham D, Flowers GE (2017) Design and analysis of experiments on nonconvex regions. Technometrics 59(1):36–47
44. Ranjan P, Bingham D, Michailidis G (2008) Sequential experiment design for contour estimation from complex computer codes. Technometrics 50(4):527–541
45. Ranjan P, Haynes R, Karsten R (2011) A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. Technometrics 53(4):366–378
46. Ranjan P, Thomas M, Teismann H, Mukhoti S (2016) Inverse problem for a time series valued computer simulator via scalarization. Open J Stat 6(3):528–544
47. Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge
48. Roy S, Notz WI (2014) Estimating percentiles in computer experiments: a comparison of sequential-adaptive designs and fixed designs. J Stat Theory Pract 8(1):12–29
49. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci, 409–423
50. Santner TJ, Williams BJ, Notz W (2003) The design and analysis of computer experiments. Springer, NewYork
51. Toscano-Palmerin S, Frazier PI (2022) Bayesian optimization with expensive integrands. SIAM J Optim 32(2):417–444
52. Tuo R, Wu CFJ (2015) Efficient calibration for imperfect computer models. Ann Stat 43:2331–2352
53. Vernon I, Goldstein M, Bower RG (2010) Galaxy formation: a Bayesian uncertainty analysis. Bayesian Anal 5(4):619–669
54. Wang H, Xian Q, Mandal A (2021) Musings about constructions of efficient latin hypercube designs with flexible run-sizes. arXiv:2010.09154
55. Wang H, Xian Q, Mandal A (2020) LHD: latin hypercube designs (LHDs). R package version 1.3.1. https://CRAN.R-project.org/package=LHD
56. Zhang R, Lin CD, Ranjan P (2019) A sequential design approach for calibrating dynamic computer simulators. SIAM/ASA J Uncertain Quantif 7(4):1245–1274