



Global Fitting of the Response Surface via Estimating Multiple Contours of a Simulator

F. Yang¹ · C. Devon Lin²  · P. Ranjan³

Published online: 3 December 2019
© Grace Scientific Publishing 2019

Abstract

Computer simulators are widely used to understand complex physical systems in many areas such as aerospace, renewable energy, climate modelling, and manufacturing. One fundamental aspect of the study of computer simulators is known as experimental design, that is, how to select the input settings where the computer simulator is run and the corresponding response is collected. Extra care should be taken in the selection process because computer simulators can be computationally expensive to run. The selection should acknowledge and achieve the goal of the analysis. This article focuses on the goal of producing more accurate prediction which is important for risk assessment and decision making. We propose two new methods of design approaches that sequentially select input settings to achieve this goal. The approaches make novel applications of simultaneous and sequential contour estimations. Numerical examples are employed to demonstrate the effectiveness of the proposed approaches.

Keywords Computer experiment · Contour estimation · Gaussian process · Latin hypercube · Maximin design · Sequential design · Space-filling

1 Introduction

Computer models or simulators are increasingly becoming popular for gaining insights of the physical processes and phenomena that are too expensive or infeasible to observe. For example, Greenberg [15] developed a finite volume community

Part of special issue guest edited by Pritam Ranjan and Min Yang—Algorithms, Analysis and Advanced Methodologies in the Design of Experiments.

✉ C. Devon Lin
devon.lin@queensu.ca

¹ College of Mathematics, Sichuan University, Chengdu, China

² Department of Mathematics and Statistics, Queen's University, Kingston, ON, Canada

³ OM&QT Area, Indian Institute of Management Indore, Indore, India

ocean model (FVCOM) for simulating the flow of water in the Bay of Fundy; Bower et al. [3] discussed the formation of galaxies using a simulator called GALFORM; and Bayarri et al. [2] used a simulator called TITAN2D for modelling the maximum volcanic eruption flow height. Realistic computer simulators of complex processes can also be computationally expensive to run, and thus, statistical surrogates trained on a handful of simulator runs are often used for the deeper understanding of the underlying phenomena. Sacks et al. [38] proposed using a realization of the Gaussian process (GP) model as a surrogate for such processes.

The popular objectives of such computer experiments include global fitting, variable screening, and estimation of process features like the maximum, a pre-specified contour or a tail quantile region. The global fitting refers to finding accurate estimation of the underlying true response surface and thus making accurate prediction over the design domain. Assuming that the simulator under consideration is expensive to run, the number of simulator runs would be limited and thus one must be careful in choosing the inputs. Over the last two decades, several innovative methodologies and algorithms have been developed to address some of the concerns. See [10, 34, 35] for details.

We focus on efficient designs for global fitting. In computer experiments literature, a popular technique is to use Latin hypercube designs [27] with some space-filling properties like maximin interpoint distance [18, 28], minimum pairwise coordinate correlation [17, 21], orthogonal array-based structure [29, 39], projection property [19], etc. Such designs aim at filling the input space as evenly as possible, but do not consider the complexity of the response surface. On the other hand, D-optimal designs [18], integrated mean-squared prediction error (IMSPE)-optimal designs [36], and maximum mean-squared prediction error (MMSPE)-optimal designs [37] use the process response information in finding a design for global fitting.

Most of these designs follow one-shot approach, i.e. all design points are obtained at the same time. However, over the past decade, a few sequential designs have also been proposed for global fitting of the response surface that have higher prediction accuracy, for instance, the D-optimal design [12, 14], expected improvement (EI) criterion-based design [22], and minimum potential energy-based design [20]. More specifically, for example, the R package **tgpr** provided the implementation of creating a sequential D-Optimal design for a stationary GP model of fixed parameterization by subsampling from a list of candidates. The algorithm is sequential in nature in that it adds one design point a time from the list of candidates, and the design point is added by maximizing the determinant of the covariance matrix constructed with the existing design points and a point from the list of candidates. In this paper, we propose two new sequential design approaches for global fitting. The proposed approaches obtain the sequential designs for achieving higher prediction accuracy based on the generalized EI criteria for contour estimation which aim to find the input settings that achieve a specific response value [33]. The rationale behind this idea is that to gain higher prediction accuracy is to have more accurate estimate of a response surface, and the estimation of a response surface can be approximated by the estimation of a large number of contours over the range of the responses. Thus, we can generalize the EI criterion

for a single-level contour estimation to that for multiple-level contour estimation. This allows us to use the closed form of the generalized EI criterion for searching sequential designs.

We propose two generalizations of EI-based criteria for contour estimation. First, we recommend splitting the range of simulator outputs into k equi-spaced contours and then develop a new EI criterion for the simultaneous estimation of these pre-specified multiple contours. Second, we propose a new adaptive approach of choosing contour levels for selecting the follow-up trial by maximizing the EI criterion for contour estimation. The performance of the proposed approaches has been compared with several state-of-the-art designs for global fitting.

The remainder of the article is organized as follows. Section 2 presents a quick review of the GP model for building a surrogate of the computer model output, popular sequential design approaches for global fitting [20, 22], and the EI criterion for contour estimation [33]. Section 3 presents the new multiple-contour estimation-based EI method for constructing designs for global fitting of the response surface. In Sect. 4, we propose the new adaptive method of estimating the contour levels for choosing follow-up design points in the sequential framework. The performance comparison of the proposed methods and the existing approaches is discussed in Sect. 5. Finally, Sect. 6 summarizes the key findings and concluding remarks.

2 Background Review

This section reviews the necessary background and the existing relevant work for later development. More specifically, we provide a brief account of reviews on GP models used throughout, the existing sequential design approaches for global fitting as well as the contour estimation in Ranjan et al. [33].

2.1 Gaussian Process Models

Gaussian process models are most widely used in computer experiments to emulate outputs from computer codes (e.g. [38]). Its popularity is due to its simplicity, flexibility, and the ability of providing the predictive uncertainty. Here we cover the key concepts of GP models and refer the reader to Santner et al. [35] and Rasmussen and Williams [34] for details. For a training data of size n , let the i th input and output of a computer code be a d -dimensional vector $\mathbf{x}_i = (x_{i1}, \dots, x_{id})^T$ and a scalar $y_i = y(\mathbf{x}_i)$, for $i = 1, \dots, n$. Typically, without loss of generality, the design domain is assumed to be a unit hypercube, $\mathcal{X} = (0, 1)^d$. A GP model assumes

$$y(\mathbf{x}_i) = \mathbf{f}^T \boldsymbol{\beta} + Z(\mathbf{x}_i), \quad i = 1, 2, \dots, n, \quad (1)$$

where \mathbf{f} is a vector of regression functions, $\boldsymbol{\beta}$ is the vector of regression parameters, $Z(\mathbf{x})$ is a stationary stochastic process with mean zero, constant variance σ^2 , and the correlation between two outputs $y(\mathbf{x}_i)$ and $y(\mathbf{x}_j)$ is denoted by $R(\mathbf{x}_i, \mathbf{x}_j) = \text{corr}(\mathbf{x}_i, \mathbf{x}_j)$. In this article, we focus on the GP models with a constant mean, that is, $\mathbf{f}^T \boldsymbol{\beta} = \mu$. Let $\mathbf{y} = (y_1, \dots, y_n)^T$ be the vector of responses for the training data, and \mathbf{R} be an

$n \times n$ spatial correlation matrix with the (i, j) th element $R(\mathbf{x}_i, \mathbf{x}_j)$. For a GP model in (1), it is equivalent to assume that \mathbf{y} follows a multivariate normal distribution with mean vector $\mu \mathbf{1}_n$ and the covariance matrix $\sigma^2 \mathbf{R}$ with $\mathbf{R} = (R(\mathbf{x}_i, \mathbf{x}_j))$, where $\mathbf{1}_n$ is an n -dimensional column vector of all 1's. Notationally, we denote $\mathbf{y} \sim GP(\mu \mathbf{1}_n, \sigma^2 \mathbf{R})$. There are many choices of valid correlation functions. One popular choice is the Gaussian correlation function,

$$R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^d \exp\{-\theta_k(x_{ik} - x_{jk})^2\}, \tag{2}$$

where θ_k is the correlation parameter for the k th input variable. The unknown parameters in the model include the mean μ , the variance σ^2 , and d correlation parameters $\theta_1, \dots, \theta_d$. They can be estimated via the maximum likelihood approach or Bayesian approach such as Markov chain Monte Carlo (MCMC) [5, 10, 23, 35]. For the maximum likelihood approach, if the correlation parameters are known, the estimates of μ and σ^2 in (1) are

$$\hat{\mu} = (\mathbf{1}_n^T \mathbf{R}^{-1} \mathbf{1}_n)^{-1} \mathbf{1}_n^T \mathbf{R}^{-1} \mathbf{y} \tag{3}$$

and

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}_n \hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}_n \hat{\mu})}{n}. \tag{4}$$

The best linear unbiased predictor (BLUP) at an input \mathbf{x}_0 is given by

$$\hat{y}(\mathbf{x}_0) = E[y(\mathbf{x}_0)|\mathbf{y}] = \mu + \mathbf{r}^T(\mathbf{x}_0) \mathbf{R}^{-1} (\mathbf{y} - \mu \mathbf{1}_n), \tag{5}$$

where $\mathbf{r}(\mathbf{x}_0) = (R(\mathbf{x}_0, \mathbf{x}_1), \dots, R(\mathbf{x}_0, \mathbf{x}_n))^T$. Moreover, the predictive variance of $y(\mathbf{x})$ is

$$s^2(\mathbf{x}_0) = \text{Var}(y(\mathbf{x}_0)|\mathbf{y}) = \sigma^2 (1 - \mathbf{r}^T(\mathbf{x}_0) \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}_0)). \tag{6}$$

In practice, the unknown correlation parameters in (3) and (4) are replaced with the estimates. Thus, μ , σ^2 , \mathbf{R} , and $\mathbf{r}(\mathbf{x}_0)$ in (5) and (6) are replaced by $\hat{\mu}$, $\hat{\sigma}^2$, $\hat{\mathbf{R}}$, and $\hat{\mathbf{r}}(\mathbf{x}_0)$, respectively. There are a number of R packages that can provide the GP model fitting, for example, **mlegp**, **GPfit**, **DiceKriging**, **tgp**, **RobustGaSP** and **SAVE** [6, 14, 16, 26, 30, 32]. These R packages are different in terms of computational efficiency and stability. In general, they should provide similar results. For the reason of stability, we use the R package **GPfit** [26] in this article.

2.2 Existing Sequential Design Approaches for Global Fitting

The general setup of a sequential design approach starts with an initial design and adds one point or a batch of points at a time sequentially. We focus on the sequential approaches of adding one point at a time. The next follow-up point should be chosen based on the information gathered from the existing data and should be most informative among the candidate points. The process of adding points is repeated until a

tolerance based stopping criterion is met or a pre-specified budget is exhausted. The step-by-step process of the sequential design approach is as follows.

- Step 1 Choose an initial design of run size n_0 . Let $n = n_0$.
 Step 2 Build a statistical surrogate model using the available data $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$.
 Step 3 Choose the next design point \mathbf{x}_{n+1} based on a criterion. Run the computer code at the new input \mathbf{x}_{n+1} and obtain the corresponding response y_{n+1} .
 Step 4 Let $n = n + 1$ and repeat Steps 2 and 3 until it reaches the run size budget or satisfies the stopping criterion.

A few remarks are in order. First, the initial design typically comes with some space-filling property like maximin interpoint distance, minimum pairwise coordinate correlation, etc. If the initial run size n_0 is too small, the resulting surrogate model could be wildly inaccurate and mislead the follow-up design choice. On the other hand, if the n_0 is relatively large, it may not fully take the advantage of sequential design criterion in Step 3. Ranjan et al. [33] recommended that the value of n_0 should be between 25 and 35% of the ultimate run size budget. Such a recommendation is based on their sequential design approach for contour estimation. Second, the run size budget certainly depends on the computer code of interest. Loepky et al. [25] provided a rule of thumb for selecting a sample size, that is, 10 times the number of input variables. In our illustrative examples, the total run size is at least $10d$. Third, in principle, any modelling methods such as GP, treed GP (TGP), or Bayesian additive regression trees (BART) [4, 11] can be used as a surrogate in Step 2. We focus on GP modelling in the examples.

2.2.1 Expected Improvement Criterion by Lam and Notz [22]

Lam and Notz [22] introduced a sequential design approach based on an *expected improvement for global fit* (EIGF) criterion which chooses the next input point that maximizes the following expected improvement. The improvement function $I(\mathbf{x})$ is defined as

$$I(\mathbf{x}) = (y(\mathbf{x}) - y(\mathbf{x}_{j^*}))^2$$

with $y(\mathbf{x}_{j^*})$ being the observed output at the sampled point, \mathbf{x}_{j^*} , that is closest in distance to the candidate point \mathbf{x} . The expected improvement is given by

$$E(I(\mathbf{x})) = (\hat{y}(\mathbf{x}) - y(\mathbf{x}_{j^*}))^2 + \text{Var}(\hat{y}(\mathbf{x})) = (\hat{y}(\mathbf{x}) - y(\mathbf{x}_{j^*}))^2 + s^2(\mathbf{x}). \quad (7)$$

Lam and Notz [22] used the Euclidean distance to determine this nearest sampled design point. The expectation in (7) is taken with respect to the predictive distribution of $y(\mathbf{x})$ under the GP model, i.e. $y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x}))$. The EIGF criterion in (7) balances the local search and global search of the next potential design input that guides the search for the ‘informative’ regions with significant variation in the response values.

2.2.2 Sequential Minimum Energy Designs by Joseph et al. [20]

Motivated by the fact in physics that the charged particles in a box repel and try to remain away from each other as much as possible, Joseph et al. [20] viewed a space-filling design in the experimental region as the positions occupied by the charged particles in a box. The charge of each particle represents the experimental response. A minimum energy design is obtained by minimizing the potential energy. Let $q(\mathbf{x})$ be the charge of the particle at the design input \mathbf{x} , and $d(\mathbf{x}_i, \mathbf{x}_j)$ denote the Euclidean distance between the i th and the j th input. Joseph et al. [20] defined the potential generalized energy (GE) of a design $\mathbf{D}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ as

$$GE_p = \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\frac{q(\mathbf{x}_i)q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \right)^p \right\}^{1/p}, \quad (8)$$

where p is in the range of $[1, \infty)$. They further proposed a sequential minimum energy design approach which works as follows. Let $\hat{q}(\mathbf{x}) = \{\hat{y}(\mathbf{x})\}^{-1/(2d)}$, where d is the dimensionality of the input \mathbf{x} . Then the proposed one-point at-a-time greedy algorithm finds the next follow-up design point given by

$$\mathbf{x}_{n+1} = \arg \min_{\mathbf{x}_0 \in \mathcal{X}} \sum_{i=1}^n \left(\frac{\hat{q}(\mathbf{x}_i)\hat{q}(\mathbf{x}_0)}{d(\mathbf{x}_i, \mathbf{x}_0)} \right)^p. \quad (9)$$

The design generated by this algorithm is called *sequential minimum energy design* (SMED).

2.3 Contour Estimation via EI Criterion

The contour at level ‘ a ’ of a simulator response surface consists of all the inputs \mathbf{x} that yield the same response a , that is,

$$S(a) = \{\mathbf{x} \in \mathcal{X} : y(\mathbf{x}) = a\}. \quad (10)$$

Ranjan et al. [33] developed an expected improvement criterion under the sequential design methodology for estimating a contour from an expensive to evaluate computer simulator with scalar responses. The proposed improvement function is,

$$I(\mathbf{x}) = \epsilon^2(\mathbf{x}) - \min \{(y(\mathbf{x}) - a)^2, \epsilon^2(\mathbf{x})\}, \quad (11)$$

where $y(\mathbf{x})$ has a normal predictive distribution, i.e. $y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x}))$, and $\epsilon(\mathbf{x}) = \alpha s(\mathbf{x})$ for a positive constant α . A suggested value for α is 1.96 for the reason that this value defines a region of interest around $S(a)$ to be 95% confidence interval under the normality assumption of the responses. Letting $v_1(\mathbf{x}) = a - \epsilon(\mathbf{x})$ and $v_2(\mathbf{x}) = a + \epsilon(\mathbf{x})$, the closed form of the expectation of the improvement function $I(\mathbf{x})$ with respect to the predictive distribution of $y(\mathbf{x})$ is given by,

$$\begin{aligned}
 E(I(\mathbf{x})) &= \int_{v_1(\mathbf{x})}^{v_2(\mathbf{x})} [\epsilon^2(\mathbf{x}) - (t - a)^2] \phi\left(\frac{t - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) dt \\
 &= [\epsilon(\mathbf{x})^2 - (\hat{y}(\mathbf{x}) - a)^2 - s^2(\mathbf{x})](\Phi(u_2) - \Phi(u_1)) + s^2(\mathbf{x})(u_2\phi(u_2) - u_1\phi(u_1)) \\
 &\quad + 2(\hat{y}(\mathbf{x}) - a)s(\mathbf{x})(\phi(u_2) - \phi(u_1)), \\
 &= T_1 + T_2 + T_3
 \end{aligned}
 \tag{12}$$

where $u_1 = [v_1(\mathbf{x}) - \hat{y}(\mathbf{x})]/s(\mathbf{x})$, $u_2 = [v_2(\mathbf{x}) - \hat{y}(\mathbf{x})]/s(\mathbf{x})$, and $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and the cumulative distribution function of a standard normal random variable, respectively. Note that we define the three terms T_1, T_2 and T_3 for easy explanation below. See Ranjan et al. [33] and the associated Errata for the derivation of (12). The first term T_1 in (12) suggests an input with a large $s(\mathbf{x})$ in the neighbourhood of the predicted contour, while the last term T_3 assigns the weights to points that are far away from the predicted contour with large uncertainties. The second term T_2 is often dominated by the other two terms in (12). Maximizing the EI criterion in (12) results in the inputs with high uncertainty near the predicted contour as well as those far away, achieving both aims of local search and global exploration.

3 Global Fitting by Estimating Multiple Contours

This section proposes a new method for constructing a sequential design for achieving higher prediction accuracy of the overall global fit. The basic sequential framework would remain the same as in Sect. 2.2, that is, start with a good initial design (e.g. maximin Latin hypercube) of size $n_0 \ll n$ and then sequentially add the remaining $n - n_0$ points using some method that feeds on the objective of global fitting. Instead of the conventional approach of trying to evenly fill the input space, the proposed idea is to slice the response surface into multiple contours and then use the sequential design approach to simultaneously estimate those contours. Next, we generalize the EI criterion for contour estimation [33] for simultaneous estimation of multiple contours.

For a given integer $k > 0$ and the set of scalar values $a_1, \dots, a_k \in [y_{\min}, y_{\max}]$, suppose that we are interested in estimating k contours $S(a_1), \dots, S(a_k)$, where $[y_{\min}, y_{\max}]$ represents the range of the true simulator response and $S(\cdot)$ is defined in (10). Without loss of generality, assume $a_1 < a_2 < \dots < a_k$. For choosing the follow-up trial, we propose the improvement function at input \mathbf{x} as

$$I(\mathbf{x}) = \epsilon^2(\mathbf{x}) - \min\{(y(\mathbf{x}) - a_1)^2, \dots, (y(\mathbf{x}) - a_k)^2, \epsilon^2(\mathbf{x})\},
 \tag{13}$$

where $y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x}))$ and $\epsilon(\mathbf{x}) = \alpha s(\mathbf{x})$ for some positive constant α . This improvement function will be nonzero only if $(y(\mathbf{x}) - a_j)^2 < \epsilon^2(\mathbf{x})$ for some j . Therefore, the improvement function can be rewritten as:

$$I(\mathbf{x}) = \max\{0, \epsilon^2(\mathbf{x}) - (y(\mathbf{x}) - a_j)^2, j = 1, 2, \dots, k\}.$$

Since $a_1 < a_2 < \dots < a_k$, the improvement function can be further simplified as

$$I(\mathbf{x}) = \begin{cases} \epsilon^2(\mathbf{x}) - (y(\mathbf{x}) - a_1)^2, & a_1 - \epsilon(\mathbf{x}) \leq y(\mathbf{x}) \leq \min\{a_1 + \epsilon(\mathbf{x}), (a_1 + a_2)/2\} \\ \dots & \dots \\ \epsilon^2(\mathbf{x}) - (y(\mathbf{x}) - a_j)^2, & \max\{a_j - \epsilon(\mathbf{x}), (a_{j-1} + a_j)/2\} \leq y(\mathbf{x}), \\ & y(\mathbf{x}) \leq \min\{a_j + \epsilon(\mathbf{x}), (a_j + a_{j+1})/2\}, \\ & 2 \leq j \leq k - 1; \\ \dots & \dots \\ \epsilon^2(\mathbf{x}) - (y(\mathbf{x}) - a_k)^2, & \max\{a_k - \epsilon(\mathbf{x}), (a_{k-1} + a_k)/2\} \leq y(\mathbf{x}) \leq a_k + \epsilon(\mathbf{x}), \\ 0, & \text{otherwise.} \end{cases}$$

The term $\epsilon(\mathbf{x})$ defines an uncertainty band around each contour that is a function of the predictive standard deviation $s(\mathbf{x})$. For the design points already chosen, the radius of the band is exactly zero. In addition, the criterion will tend to be large for the samples from one of the sets $(\{\mathbf{x} : y(\mathbf{x}) = a_1\}, \{\mathbf{x} : y(\mathbf{x}) = a_2\}, \dots, \{\mathbf{x} : y(\mathbf{x}) = a_k\})$, where $s(\mathbf{x})$ is large.

Similar to other sequential design approaches, we suggest choosing follow-up design points by maximizing the corresponding expected improvement, where the expectation is taken with respect to the predictive distribution, $y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x}))$. For $j = 1, \dots, k$, let $v_{j1}(\mathbf{x})$'s and $v_{j2}(\mathbf{x})$'s be defined as follows,

$$v_{j1}(\mathbf{x}) = \begin{cases} a_1 - \epsilon(\mathbf{x}), & j = 1; \\ \max\{a_j - \epsilon(\mathbf{x}), (a_{j-1} + a_j)/2\}, & 2 \leq j \leq k, \end{cases} \tag{14}$$

and

$$v_{j2}(\mathbf{x}) = \begin{cases} \min\{a_j + \epsilon(\mathbf{x}), (a_j + a_{j+1})/2\}, & 1 \leq j \leq k - 1; \\ a_k + \epsilon(\mathbf{x}), & j = k. \end{cases} \tag{15}$$

Then, the expectation of the improvement function in (13) is simply the sum of the individual contour estimation EI criterion of Ranjan et al. [33] over k cases, i.e.

$$\begin{aligned} E[I(\mathbf{x})] &= \sum_{j=1}^k \int_{v_{j1}(\mathbf{x})}^{v_{j2}(\mathbf{x})} [\epsilon^2(\mathbf{x}) - (t - a_j)^2] \phi\left(\frac{t - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) dt \\ &= \sum_{j=1}^k \left\{ [\epsilon(\mathbf{x})^2 - (\hat{y}(\mathbf{x}) - a_j)^2 - s^2(\mathbf{x})](\Phi(u_{j2}) - \Phi(u_{j1})) \right. \\ &\quad \left. + s^2(\mathbf{x})(u_{j2}\phi(u_{j2}) - u_{j1}\phi(u_{j1})) + 2(\hat{y}(\mathbf{x}) - a_j)s(\mathbf{x})(\phi(u_{j2}) - \phi(u_{j1})) \right\}, \end{aligned} \tag{16}$$

where $u_{j1} = (v_{j1}(\mathbf{x}) - \hat{y}(\mathbf{x}))/s(\mathbf{x})$ and $u_{j2} = (v_{j2}(\mathbf{x}) - \hat{y}(\mathbf{x}))/s(\mathbf{x})$, $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and the cumulative distribution function of a standard normal random variable, respectively. The formulation in (16) reduces to (12) when the number of contour levels is $k = 1$. Compared with the EI criterion in (7) by Lam and Notz [22], the proposed criterion has different weighting on the variance.

Note that the maximization of $E[I(\mathbf{x})]$ over $I(\mathbf{x})$ has two advantages. First, the true value of $y(\mathbf{x})$ (and hence $I(\mathbf{x})$) is unknown for any unsampled design point. Second, some regions of the design space may not have been sufficiently explored yet and the

predictive variance of $\hat{y}(\mathbf{x})$ is relatively high. For such an unsampled design point, the predicted response is not within the $\epsilon(\mathbf{x})$ -band of one of those k estimated contours, but it may be close to the true contours that may lie in the unexplored region. As a result, the EI approach facilitates a balance between the local exploitation versus global exploration.

The use of the EI criterion in (16) involves the choices of k - the number of contours, and the contour levels a_1, a_2, \dots, a_k . Finding their optimal choice with respect to achieving more accurate predictions appears to be a challenging task. A reasonable way to choose their values is to use equi-spaced k contours in the simulator output range $[y_{\min}, y_{\max}]$. The values of y_{\min} and y_{\max} are unknown in general. We thus estimate these two values using the fitted model. That is, we approximate y_{\min} and y_{\max} using the minimal and maximal of $\hat{y}(\mathbf{x}_1), \dots, \hat{y}(\mathbf{x}_m)$ where $\mathbf{x}_1, \dots, \mathbf{x}_m$ are a large number of inputs from the design domain, m is chosen to be $1000d$ in all the examples and $\hat{y}(\mathbf{x}_j)$ is the fitted response at \mathbf{x}_j using the fitted model at the current stage. Note that the existing methods reviewed in Sect. 2.2 do not make use of the information of y_{\min} and y_{\max} . Using such information in the proposed method may contribute to its superior performance.

We now present two illustrations of the proposed multiple-contour estimation EI criterion (referred to as MC criterion) for global fitting with different values of k .

Example 1 Consider the computer model [13] that relates the one-dimensional input x and the output y as,

$$y = \frac{\sin(10\pi x)}{2x} + (x - 1)^4, \quad 0.5 \leq x \leq 2.5. \quad (17)$$

The true relationship between the input x and the output y is displayed in the blue solid curve in Fig. 1. Five initial design points are shown by black empty circles. We then sequentially add 15 design points using the MC criterion in (16). The numerical labels represent the order of the newly added design points. Figure 1a–d illustrates the sequential design scheme with the MC criterion for 1, 5, 10, 20 equally spaced contour levels within the ranges of the fitted surface. When $k = 1$, the majority of the added points are around the contour level $a = 2.0$. For larger values of k , as k equally spaced contour levels are used, the added points by the proposed MC criterion-based sequential design approach are more space-filling. In addition, Fig. 1 reveals that in this example, the proposed approach can choose the inputs that are around the areas where the function changes the direction and locate most of the points in the areas where the computer model is more complex.

Example 2 Consider a computer model with two-dimensional input variables $\mathbf{x} = (x_1, x_2)$, and the output given by

$$y(\mathbf{x}) = [1 + (4x_1 + 4x_2 + 1)](3 + 192x_1x_2), \quad 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1. \quad (18)$$

The contour plot of the response surface is visually displayed in Fig. 2. Suppose a maximin Latin hypercube design of 10 points is generated and the corresponding

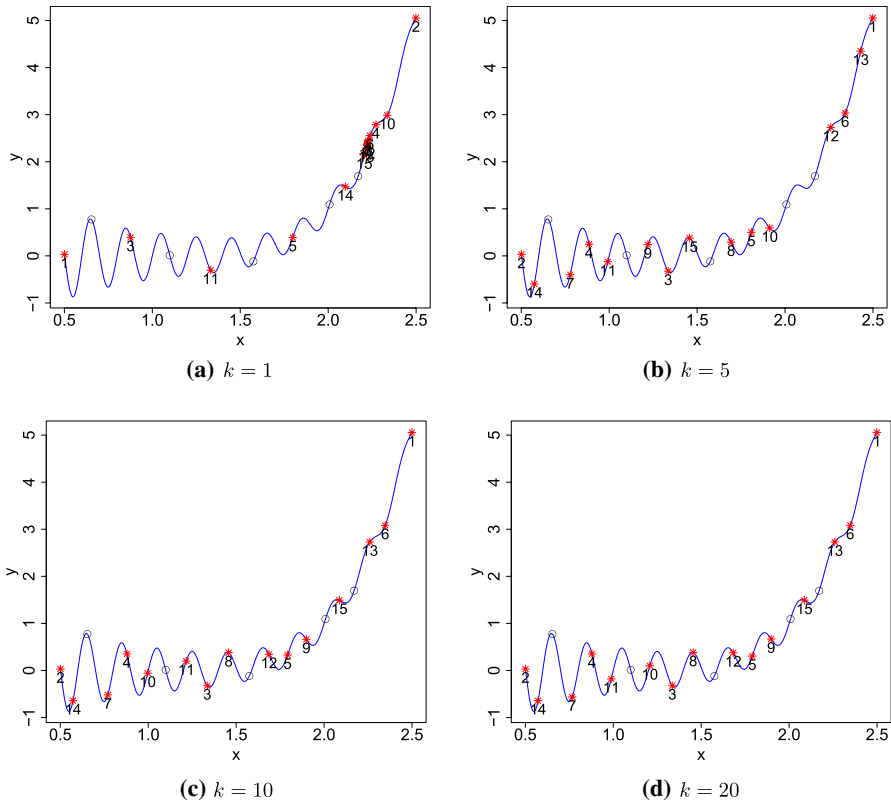


Fig. 1 Illustration of MC criterion with k contour levels. The blue curves represent the true relationship between x and y of the computer model in (17); the black empty circles are the five initial design points; the red stars are locations of follow-up design points, and the numerical values are the order of the added points

responses are collected from the computer model. First, we consider searching for the next follow-up design point for estimating only one contour at the level $a = 300$. Figure 3 shows these 10 design points, the inputs with $I(\mathbf{x}) > 0$, and the maximizer of the EI criterion for contour estimation in (12) from the candidate set on a regular 100×100 rectangular grid.

Next, we consider the simultaneous estimation of three contours at levels $a_1 = 150$, $a_2 = 300$ and $a_3 = 600$ using the MC criterion in (16). Figure 4 shows the inputs from the same 50×50 grid candidate set that achieve nonzero improvement (13), and the point that maximizes the MC criterion. This point, depicted by red empty circle in Fig. 4, is in fact from the set of the points that yield nonzero improvement around the contour level $a_1 = 150$.

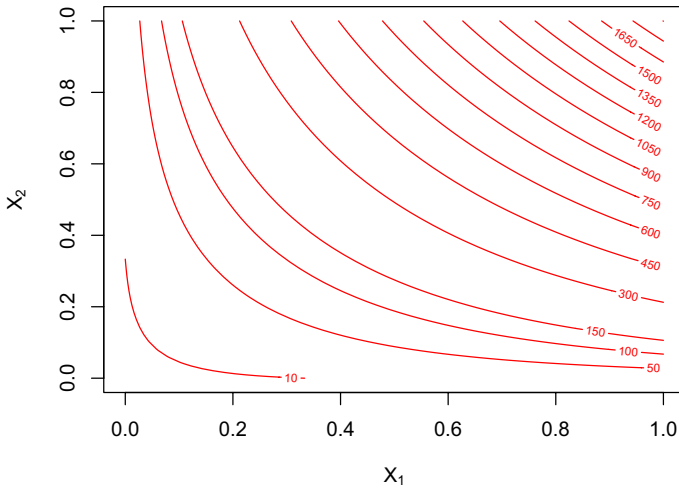


Fig. 2 The contour plot of the response surface of the computer simulator in (18)

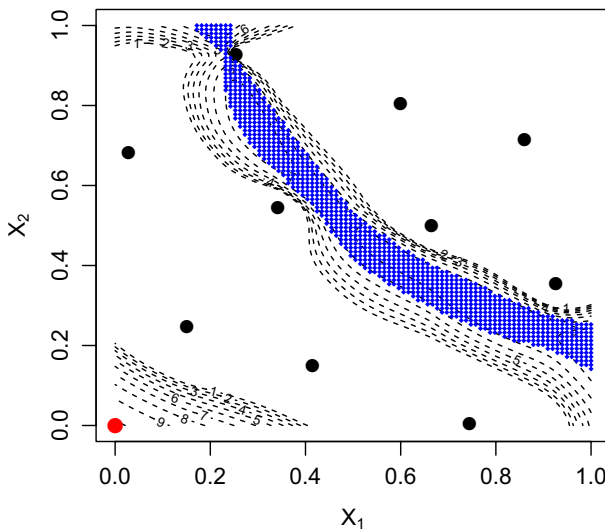


Fig. 3 Illustration of the follow-up point selection method using the EI criterion for contour estimation from the computer model (18). The black solid circles denote the training points, blue dots represent nonzero improvement value, i.e. $\{x : |y(x) - a| \leq \epsilon(x)\}$ for the contour level $a = 300$, the contour lines display $\log(E[I(x)])$ values, and the red solid circle shows the maximizer of the EI criterion

Figure 5 illustrates the complete sequential design scheme with 20 initial design points and 30 follow-up design points for simultaneously estimating three contours at levels $a_1 = 150$, $a_2 = 300$, and $a_3 = 600$. The red squares are the new follow-up points, and the label corresponds to the order the point is added. The last panel displays the squared distance between the estimated contour and the true contour

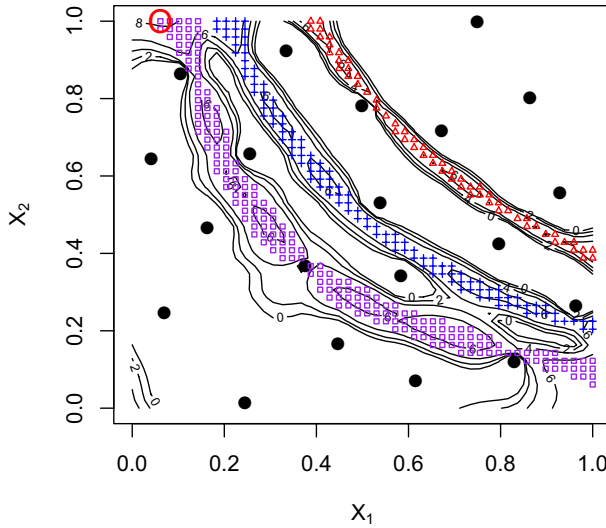


Fig. 4 Illustration of the follow-up point selection method using the MC criterion (at levels $a_1 = 150$, $a_2 = 300$, $a_3 = 600$) for the computer model (18). The black solid circles denote 20 training points. The purple square circles, blue pluses, and red triangles represent improvement around the three contour levels, respectively. The contour lines display $\log(E(I(\mathbf{x})))$, and the red circle represents the maximizer of the MC criterion in (16)

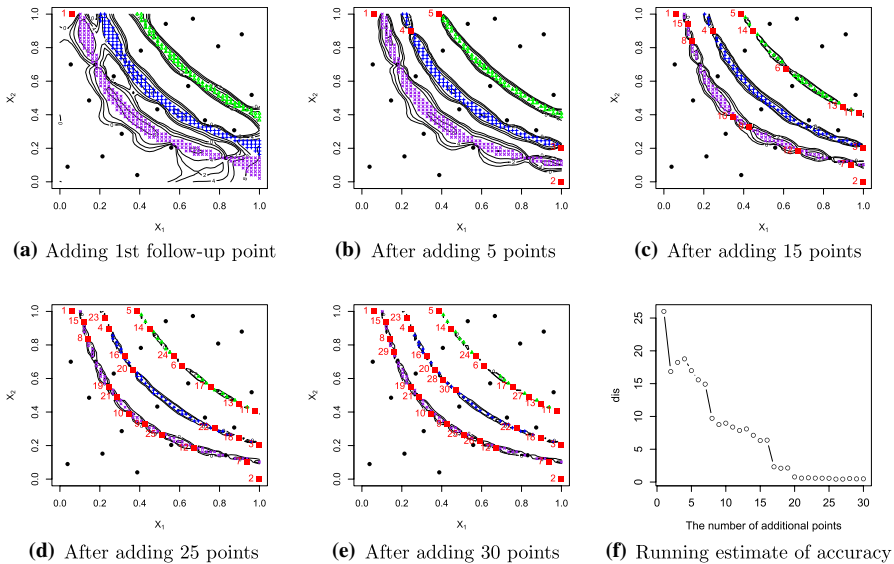


Fig. 5 Illustration of the MC criterion for contour levels $a_1 = 150$, $a_2 = 300$ and $a_3 = 600$ with $n_0 = 20$ initial design points and 30 follow-up points. The accuracy in **f** is measured by the squared distance between the estimated contour after adding i -follow-up points and the true contour

at each stage. It can be observed from Fig. 5 that the estimated uncertainty bands around the three contours become narrower and more accurate. It can also be seen that more points are added to estimate the contour level $a_1 = 150$ than to estimate the other two contours. Some points such as the second point are away from the contour bands.

It is clear from the two examples that the resulting designs do not have the conventional space-filling property. This is desirable as the objective is an overall good fit of the response surface and not to explore the input space. However, as illustrated in Fig. 5, a significant fraction of design points tend to line up on the pre-specified contours, which could lead to biased designs if a_1, \dots, a_k are not chosen appropriately. Next, we propose an efficient method of selecting contour levels.

4 Sequential Estimation of Contours for Global Fitting

In this section, we propose a new approach for choosing the follow-up design points. Different from the previous section where the simultaneous estimation of multiple contours was used for global fitting, we adopt the EI criterion for estimating only one contour level at each stage. That is, at each stage, we choose a contour level and find the design point that maximizes the criterion (12) with the chosen contour level. The important issue then is how to choose the contour level at each stage. We propose the following way to choose such contour level in an automatic way. Suppose at stage j , the training data are $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$ and the corresponding emulator gives the predictive distribution as $y(\mathbf{x}) \sim N(\hat{y}(\mathbf{x}), s^2(\mathbf{x}))$ for any input \mathbf{x} . Let the candidate set for the next follow-up point be $\mathbf{x}_1^*, \dots, \mathbf{x}_m^*$ and

$$\mathbf{x}_{opt}^* = \arg \max_{1 \leq i \leq m} s^2(\mathbf{x}_i^*).$$

Then, we choose the contour level at stage j as $a_j = \hat{y}(\mathbf{x}_{opt}^*)$. In other words, at each stage, we set the contour level to be the fitted response that has maximum predictive variance. This is to encourage exploring the area with maximum uncertainty.

Example 2 (contd.) Consider finding a design for global fitting of the computer simulator in Example 2. The procedure starts with an initial design of size $n_0 = 10$ obtained via maximin Latin hypercube sampling and $n - n_0 = 30$ follow-up points are chosen as per the proposed sequential strategy. Figure 6 displays the follow-up design points found by the proposed method, i.e. the sequential contour estimation-based EI criterion, as well as the trace plot of the contour values.

Note that the resulting design is more space-filling as compared to a systematic layout of points on the contour lines shown in Fig. 5. Again, the design is not completely space-filling and it has some pairs of close-by points.

Before concluding this section, we make some remarks on the computational cost of the approaches. We implemented all four methods [(EIGF, SMED, MC and sequential contour (SC)] in R with the help of the **GPfit** package used for fitting GP models and obtaining the predicted mean and variance, $\hat{y}(\mathbf{x})$ and $s^2(\mathbf{x})$, for every \mathbf{x} in the test set. For all four methods, the overall sequential framework remains the same, but the design criteria for choosing the follow-up trial are different. From a

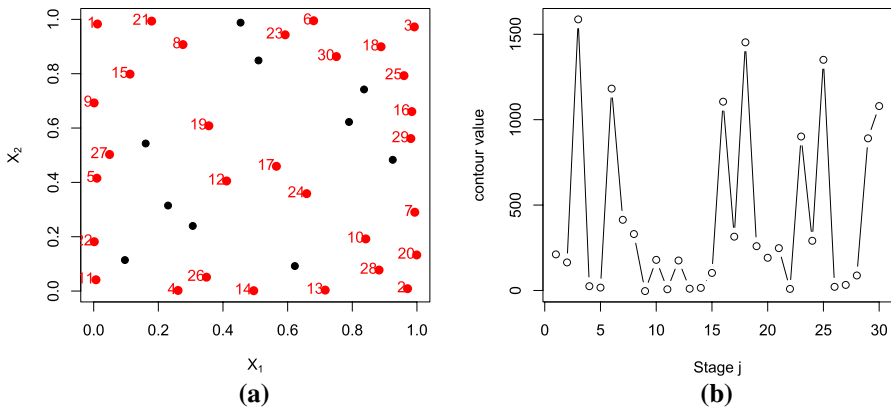


Fig. 6 Illustration of the sequential contour estimation-based EI criterion for global fitting with $n_0 = 10$ and 30 added points in Example 2: **a** the initial points (in black) and the added points (in red); **b** the contour value versus the stage j , for $j = 1, \dots, 30$

computational cost standpoint, the evaluation of the EIGF criterion is dominated by finding $y(\mathbf{x}_{j*})$, which amounts to computing and sorting the distance between \mathbf{x}_0 (the candidate test point) and n training points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The computationally dominating part of the SMED criterion is the evaluation of distance between \mathbf{x}_0 and n training points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, and then a sum (which is inexpensive). Thus, the SMED evaluation should be cheaper than that of the EIGF criterion. For the MC criterion with k contour levels, one has to sort $2k - 2$ numeric strings of size two each, and $2k$ evaluations of normal cumulative distribution function. For SC criterion evaluation, no sorting is required, and the normal cumulative distribution function has to be computed for only two values. However, the SC method requires an additional sorting of $s^2(\mathbf{x})$ values over the test set.

5 Simulated Examples

In this section, we conduct a simulation study to demonstrate the effectiveness of the proposed sequential design approaches. Specifically, we compare the proposed approaches with the following methods:

- a one-shot maximin Latin hypercube design;
- the sequential D -optimal design in the R package **tgp**;
- the sequential approach by Lam and Notz [22];
- the sequential minimum energy design in Joseph et al. [20];
- the proposed multiple contours estimation-based criterion in Sect. 3;
- the proposed sequential contour estimation-based criterion in Sect. 4.

For approach (e), we use the 10 equally spaced contour levels, that is, k is set to be 10. These methods are denoted by ‘maximinLHD’, ‘tgp’, ‘EIGF’, ‘SMED’, ‘MC_10’, and ‘SC_var’.

Several criteria can be used to evaluate the performance of different design approaches in comparison. We adopt the root-mean-squared prediction error (RMSPE) given by

$$\text{RMSPE} = \sqrt{\frac{1}{|\mathcal{X}_{\text{pred}}|} \sum_{\mathbf{x} \in \mathcal{X}_{\text{pred}}} (\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2}, \tag{19}$$

where $\hat{y}(\mathbf{x})$ and $y(\mathbf{x})$ are the predicted response and the true response at the new input \mathbf{x} in the hold-out set $\mathcal{X}_{\text{pred}}$. Another criterion we use is the maximum error provided by

$$\text{Maximum error} = \max_{\mathbf{x} \in \mathcal{X}_{\text{pred}}} |\hat{y}(\mathbf{x}) - y(\mathbf{x})|. \tag{20}$$

For each example below, the initial design for sequential designs is a maximin Latin hypercube design of n_0 runs generated using the *R* package **SLHD** [1]. The model fitting is implemented using the default setting of the function `GP_fit` in the *R* package **GPfit**. The test data are a random Latin hypercube design of $1000d$ points where d is the number of input variables. The parameter α in ‘MC_10’ and ‘SC_var’ is set to be 2.

Example 3 We consider the computer model with two input variables x_1 and x_2 ,

$$y = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \quad -5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15. \tag{21}$$

This model is known as Branin function [9]. We use $n_0 = 10$ initial design points. The total run size budget is 30. Figure 7 displays the boxplots of RMSPEs and maximum errors of the different design approaches over 100 simulations. The results show that in this example the one-shot approach ‘maximinLHD’ is the worst while the approaches ‘MC_10’ and ‘SC_var’ are comparably better than the others.

Example 4 We consider the computer model with three input variables x_1, x_2 , and x_3 [8]

$$y = 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2, \quad 0 \leq x_i \leq 1, \text{ for } i = 1, \dots, 3. \tag{22}$$

We use $n_0 = 20$ initial design points. The total run size budget is 60. Figure 8 displays the boxplots of RMSPEs and maximum errors of the different design approaches over 100 simulations. Again, here the one-shot approach ‘maximinLHD’ is the worst. The approaches ‘tgp’, ‘MC_10’ and ‘SC_var’ are comparably better

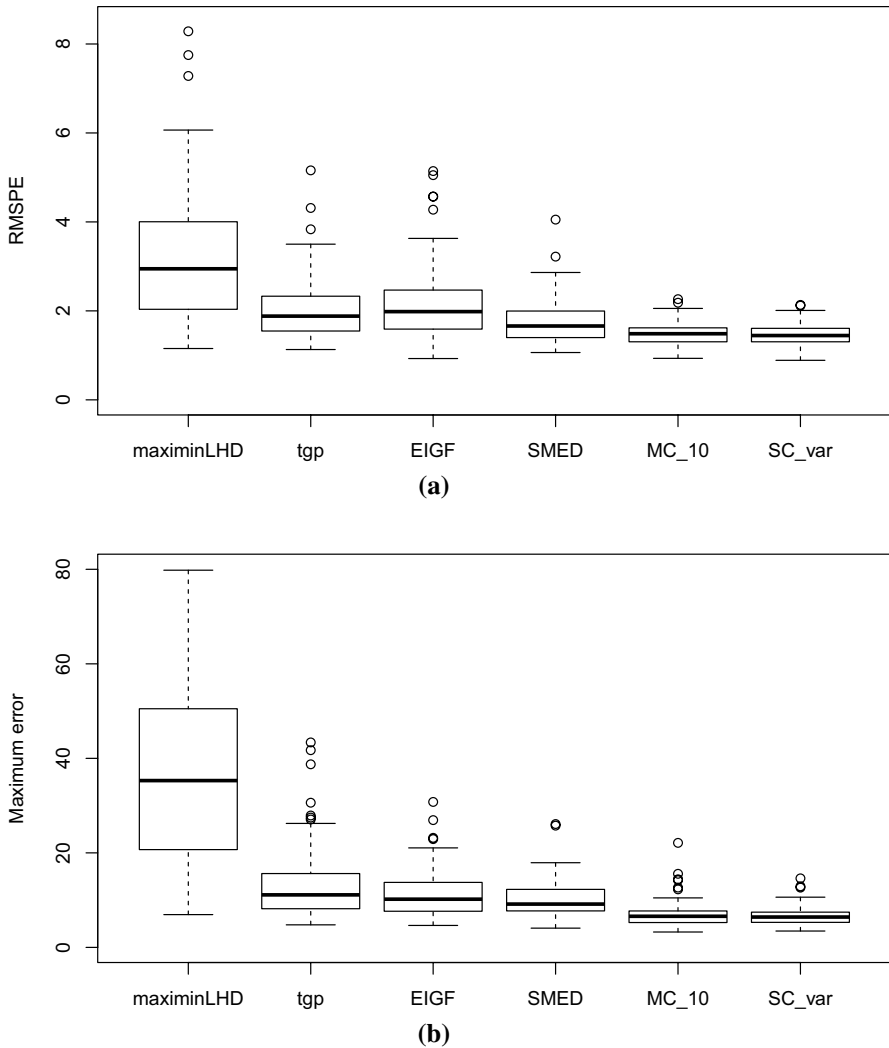


Fig. 7 The boxplots of RMSPEs and maximum errors of the methods ‘maximinLHD’, ‘tgp’, ‘EIGF’, ‘SMED’, ‘MC_10’, and ‘SC_var’ for the computer model in (21) with $n_0 = 10$ and 20 added points over 100 simulations

than the others in terms of RMSPE, and the proposed approaches ‘MC_10’ and ‘SC_var’ are significantly better than the others in terms of maximum errors.

Example 5 We consider the computer model with four input variables $x_1, x_2, x_3,$ and x_4 [31]

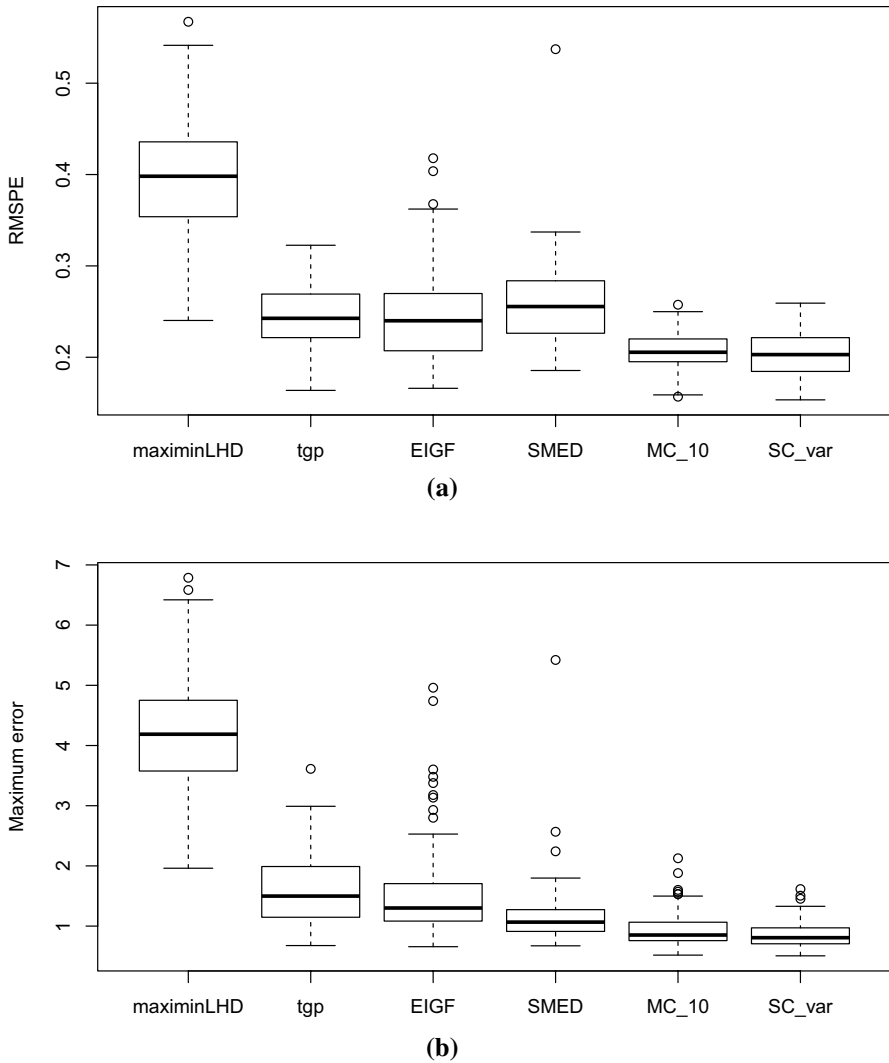


Fig. 8 The boxplots of RMSPEs and maximum errors of the methods ‘maximinLHD’, ‘tgp’, ‘EIGF’, ‘SMED’, ‘MC_10’, and ‘SC_var’ for the computer model in (22) with $n_0 = 20$ and 40 added points over 100 simulations

$$y = \frac{x_1}{2} \left[\sqrt{1 + (x_2 + x_3^2) \frac{x_4}{x_1^2}} - 1 \right] + (x_1 + 3x_4) \exp(1 + \sin(x_3)), 0 \leq x_i \leq 1, \text{ for } i = 1, \dots, 4. \tag{23}$$

We use $n_0 = 27$ initial design points. The total run size budget is 80. Figure 9 displays the boxplots of RMSPEs and maximum errors of the different design

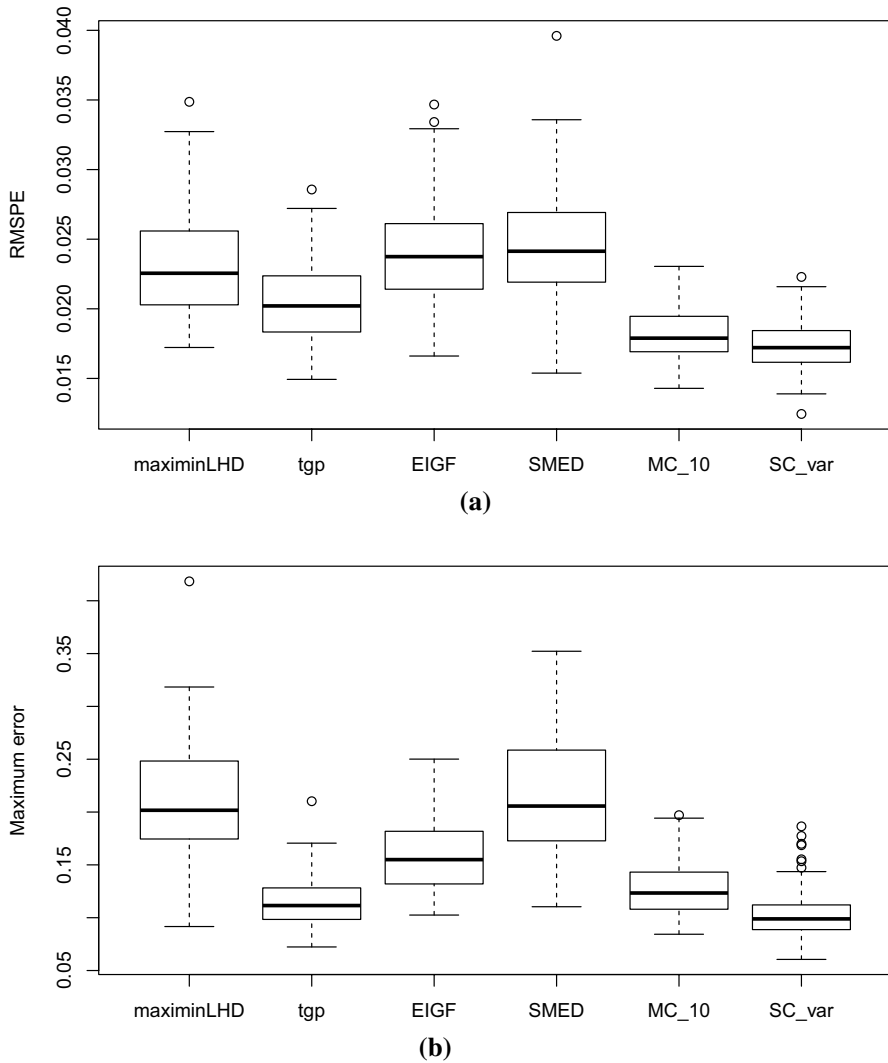


Fig. 9 The boxplots of RMSPEs and maximum errors of the methods ‘maximinLHD’, ‘tgp’, ‘EIGF’, ‘SMED’, ‘MC_10’, and ‘SC_var’ for the computer model in (23) with $n_0 = 27$ and 53 added points over 100 simulations

approaches over 100 simulations. Here, the approach ‘SMED’ is the worst followed by the approach ‘maximinLHD’ in terms of both criteria. The performance of the other four approaches is similar based on RMSPE. However, based on the maximum error, the proposed approaches give more accurate predictions.

6 Concluding Remark

In this article we have developed two sequential design approaches for accurately predicting a complex computer code. The approaches are based on the expected improvement criteria for simultaneously or sequentially estimating contours. We used a GP model as a surrogate for the computer simulator, which is an integral component of the proposed criteria for identifying the follow-up trials. Numerical examples are given to demonstrate that the proposed approaches can significantly outperform the existing approaches.

A few remarks are in order. First, one can easily extend the methodology for a GP surrogate with a small nugget term, which are typically used for either increasing the numerical stability of the model or to account for small noise in the simulator output. Second, if some other surrogate is used instead of GP model, then also the key ideas like formulation of improvement function and sequential estimation of contour levels can be retained. Of course, the resultant expected improvement criteria would change, and in fact, one may not even end up with a closed form expression of the final design criterion for selecting follow-up points. Third, this article focuses on one-point-at-a-time fully sequential designs. However, in some applications, a batch sequential design is more desirable and practical than a fully sequential design [24]. Extending the expected improvement criteria to batch sequential designs is not trivial and would require a thorough investigation. We hope to address such an issue in the near future. Other future work includes the application of the proposed contour estimation-based sequential design approaches for global fitting for computer experiments with both qualitative and quantitative factors [7] and dynamic computer experiments [40].

Acknowledgements The authors would like to thank the chief editor, the handling editor, and the reviewers for their helpful comments. Yang's research was funded by China Scholarship Council. Lin's research was funded in part by the Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. Ranjan's research was partially supported by the Extra Mural Research Fund (EMR/2016/003332/MS) from the Science and Engineering Research Board (SERB), Government of India.

References

1. Ba S (2015) SLHD: maximin-distance (sliced) Latin hypercube designs. R package version 2.1-1
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. Bower RG, Benson AJ et al (2006) The broken hierarchy of galaxy formation. *Mon Not R Astron Soc* 370:645–655
4. Chipman H, Ranjan P, Wang W (2012) Sequential design for computer experiments with a flexible Bayesian additive model. *Can J Stat* 40(4):663–678
5. Currin C, Mitchell T, Morris M, Ylvisaker D (1991) Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J Am Stat Assoc* 86(416):953–963
6. Dancik G (2018) mlegp: maximum likelihood estimates of Gaussian processes. R package version 3.1.7

7. Deng X, Lin CD, Liu KW, Rowe RK (2017) Additive Gaussian process for computer models with qualitative and quantitative factors. *Technometrics* 59:283–292
8. Dette H, Pepelyshev A (2010) Generalized Latin hypercube design for computer experiments. *Technometrics* 52:421–429
9. Dixon LCW, Szego GP (1978) The global optimization problem: an introduction. *Towards Glob Optim* 2:1–15
10. Fang KT, Li R, Sudjianto A (2005) Design and modeling for computer experiments. CRC Press, New York
11. Gramacy RB, Lee HKH (2008) Bayesian treed Gaussian process models with an application to computer modeling. *J Am Stat Assoc* 103(483):1119–1130
12. Gramacy RB, Lee HKH (2009) Adaptive design and analysis of supercomputer experiments. *Technometrics* 51(2):130–145
13. Gramacy RB, Lee HK (2012) Cases for the nugget in modeling computer experiments. *Stat Comput* 22(3):713–722
14. Gramacy RB, Taddy MA (2016) tgp: Bayesian treed Gaussian process models. R package version 2-4-14
15. Greenberg D (1979) A numerical model investigation of tidal phenomena in the Bay of Fundy and Gulf of Maine. *Mar Geod* 2:161–187
16. Gu M, Palomo J, Berger J (2018) RobustGaSP: robust Gaussian stochastic process emulation. R package version 0.5.6
17. Iman RL, Conover WJ (1982) A distribution-free approach to inducing rank correlation among input variables. *Commun Stat Part B Simul Comput* 11:311–334
18. Johnson M, Moore L, Ylvisaker D (1990) Minimax and maximin distance design. *J Stat Plan Inference* 26:131–148
19. Joseph VR, Dasgupta T, Tuo R, Wu CJ (2015) Sequential exploration of complex surfaces using minimum energy designs. *Technometrics* 57:64–74
20. Joseph VR, Gul E, Ba S (2015) Maximum projection designs for computer experiments. *Biometrika* 102:371–380
21. Joseph VR, Hung Y (2008) Orthogonal-maximin Latin hypercube designs. *Stat Sin* 18:171–186
22. Lam CQ, Notz WI (2008) Sequential adaptive designs in computer experiments for response surface model fit. *Stat Appl* 6:207–233
23. Linkletter C, Bingham D, Hengartner N, Higdon D, Ye KQ (2006) Variable selection for Gaussian process models in computer experiments. *Technometrics* 48(4):478–490
24. Loepky JL, Moore LM, Williams BJ (2010) Batch sequential designs for computer experiments. *J Stat Plan Inference* 140(6):1452–1464
25. Loepky JL, Sacks J, Welch WJ (2009) Choosing the sample size of a computer experiment: a practical guide. *Technometrics* 51(4):366–376
26. MacDonald B, Ranjan P, Chipman H (2015) GPfit: an R package for Gaussian process model fitting using a new optimization algorithm. R package version 1.0-1
27. McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21:239–45
28. Morris MD, Mitchell TJ (1995) Exploratory designs for computer experiments. *J Stat Plan Inference* 43:381–402
29. Owen AB (1992) Orthogonal arrays for computer experiments, integration and visualization. *Stat Sin* 2:439–452
30. Palomo J, Paulo R, Garcia-Donato G (2015) SAVE: an r package for the statistical analysis of computer models. *J Stat Softw* 64(13):1–23
31. Park JS (1991) Tuning complex computer codes to data and optimal designs. Unpublished Ph.D. thesis, University of Illinois, Champaign-Urbana
32. Roustant O, Ginsbourger D, Deville Y (2018) DiceKriging: kriging methods for computer experiments. R package version 1.5.6
33. Ranjan P, Bingham D, Michailidis G (2008) Sequential experiment design for contour estimation from complex computer codes. *Technometrics* 50(4):527–541 (Errata (2011), *Technometrics*, 53:1, 109–110)
34. Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge, MA
35. Santner TJ, Williams BJ, Notz WI (2003) The design and analysis of computer experiments. Springer, New York

36. Sacks J, Schiller SB, Welch WJ (1989) Designs for computer experiments. *Technometrics* 31:41–47
37. Sacks J, Schiller S (1988) Spatial designs. In: Gupta, Berger (eds) *Statistical decision theory and related topics IV*, vol 2. Springer, New York, pp 385–399
38. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4:409–423
39. Tang B (1993) Orthogonal array-based Latin hypercubes. *J Am Stat Assoc* 88:1392–1397
40. Zhang R, Lin CD, Ranjan P (2018) Local approximate Gaussian process model for large-scale dynamic computer experiments. *J Comput Graph Stat* 27:798–807

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.