Reliability Engineering and System Safety 96 (2011) 1208–1219

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Batch sequential design to achieve predictive maturity with calibrated computer models

Brian J. Williams [a,*], Jason L. Loeppky [b], Leslie M. Moore [a], Mason S. Macklem [b]

[a] Statistical Sciences Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[b] Department of Mathematics and Statistics, University of British Columbia, Okanagan, Kelowna, Canada BC V1V 1V7

ARTICLE INFO

Article history:
Received 12 November 2009
Received in revised form
4 March 2010
Accepted 27 April 2010
Available online 7 April 2011

Keywords:
Computer experiment
Gaussian process
Calibration
Sequential experiment design
Expected improvement
Maximin distance
Entropy

ABSTRACT

Sequential experiment design strategies have been proposed for efficiently augmenting initial designs to solve many problems of interest to computer experimenters, including optimization, contour and threshold estimation, and global prediction. We focus on batch sequential design strategies for achieving maturity in global prediction of discrepancy inferred from computer model calibration. Predictive maturity focuses on adding field experiments to efficiently improve discrepancy inference. Several design criteria are extended to allow batch augmentation, including integrated and maximum mean square error, maximum entropy, and two expected improvement criteria. In addition, batch versions of maximin distance and weighted distance criteria are developed. Two batch optimization algorithms are considered: modified Fedorov exchange and a binning methodology motivated by optimizing augmented fractional factorial skeleton designs.

Published by Elsevier Ltd.

1. Introduction

The National Nuclear Security Administration (NNSA) is tasked with maintaining the reliability of the nation's nuclear weapons stockpile. In 2001, the three national security laboratories began implementation of a Quantification of Margins and Uncertainties (QMU) framework to support quantitative inferences of design margins (M) and performance uncertainties (U) for weapons systems maintained by the stockpile stewardship program that began in 1994. In this article, we develop nine sequential approaches to adding new field data for calibration of computer models to efficiently achieve stability in code predictions of quantities of interest. The concepts developed in this article, perhaps with application specific modifications, support QMU methodology by recommending efficient use of experimental resources to improve code predictions of metrics used in the computation of M and U. In particular, a rigorous sequential experimental design capability would support recommendation I-1b of the recent National Research Council evaluation of QMU methodology as applied to the stockpile stewardship program [1]: "The laboratories and NNSA should strive to improve the connections between advanced simulation and computing simulation programs and experimental programs."

We present the proposed sequential design framework for a general statistical model, followed by specialization to the Kennedy and O'Hagan [2] statistical model commonly used as the basis for calibrating uncertain model parameters in simulations to available field data. In this context, stability is desired in predictions of discrepancy (computer model inadequacy), the calibrated computer model, and physical reality. The focus is on collecting new field data or simulations according to batch sequential algorithms based on maximizing expected improvement and distance-based criteria.

There is now an extensive literature in the statistical design and analysis of computer models on the use of expected improvement criteria in sequential experimental design strategies. The focus has primarily been on optimization of computer models. Schonlau et al. [3] and Jones et al. [4] introduced a one-step sequential algorithm for efficient optimization, with extensions allowing for batch updates and for constrained optimization involving multiple independent model outputs. Williams et al. [5] extended this methodology for sequential optimization of control parameters in integrated computer models. Lehman et al. [6] proposed a sequential algorithm for finding a robust setting of the control variables in an integrated computer model. Keane [7] developed a sequential design to construct Pareto fronts for multi-objective optimization with expensive computer models. Booker et al. [8] provided a rigorous framework for sequential optimization of computer models with convergence theory, extended to constrained optimization problems in Audet et al. [9]. Regis and Shoemaker [10] presented an alternative algorithm for constrained optimization of computer models with

* Corresponding author. Tel.: +1 505 667 2331; fax: +1 505 667 4470.
E-mail addresses: brianw@lanl.gov (B.J. Williams), jason@stat.ubc.ca (J.L. Loeppky), lmoore@lanl.gov (L.M. Moore), mason.macklem@ubc.ca (M.S. Macklem).

0951-8320/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.ress.2010.04.017

convergence results. Ranjan et al. [11] proposed a sequential algorithm for estimating a contour from a computer model.

Johnson et al. [12] and Morris and Mitchell [13] successfully employed distance-based criteria for the purpose of generating initial designs for computer experiments. Loeppky et al. [14] consider the use of distance-based metrics in batch sequential augmentation of computer experiment designs for the purpose of achieving improved global prediction of model output.

Any of the sequential design criteria proposed in this article can be implemented in a one at a time fashion. However, adding runs one at a time can often lead to sub-optimal run placement. Loeppky et al. [14] illustrate that this is true with many of the common criteria used to select new runs for improving surrogate predictions of complex computer models. Furthermore, budget and time constraints often dictate that batches of new runs must be added. In the case of a slow computer model, one at time updates are simply not feasible due to the time required to produce new runs. In addition it is often the case that multiple processors are available for collecting new code runs, so that one at a time updates would be inefficient since not all of the available resources are utilized. In the case of physical experiments, it is generally more cost effective for laboratories to process batches of experimental units at a time. This may be driven by the complexity of arranging experimental setups or by a fixed allocation of time for use of an experimental facility. In general the choice of batch size is typically not under the control of the experimenter but is dictated by the availability of resources.

Section 2 formally introduces the predictive maturity framework. Two expected improvement criteria are introduced for application in batch sequential design. Section 3 specializes the predictive maturity framework to the computer model calibration approach introduced by Kennedy and O'Hagan [2]. Batch sequential implementations of integrated and maximum mean square error criteria, maximum entropy and distance-based criteria, are developed. Section 4 presents a simulation study to evaluate the performance of twelve experiment design strategies in terms of prediction errors for calibration problems with and without discrepancy present. Section 5 concludes with a discussion of sequential design to achieve predictive maturity.

2. Predictive maturity

Suppose there is interest in making predictive inference about a physical system. For example, let $\xi(\mathbf{x})$ denote the true response of the physical system assuming design settings $\mathbf{x}$, where inputs $\mathbf{x}$ are controllable parameters that can be "dialed in" prior to conducting field experiments to learn about $\xi(\cdot)$. In actuality, we never directly observe $\xi(\mathbf{x})$ for any $\mathbf{x}$; we observe corrupted versions $y(\mathbf{x})$, where a variety of possible errors $\varepsilon(\mathbf{x})$ prevent direct observation of $\xi(\mathbf{x})$

$$y(\mathbf{x}) = \xi(\mathbf{x}) + \varepsilon(\mathbf{x}).$$

For the moment, $\varepsilon(\mathbf{x})$ collectively represents all sources of observational error, including measurement errors, systematic errors, and replicate variability.

To make progress, statistical models for $\xi(\mathbf{x})$ and $\varepsilon(\mathbf{x})$ are specified. For example, empirical modeling of data often involves specifying a linear model for $\xi(\mathbf{x})$

$$\xi(\mathbf{x}) = \sum_{i=0}^{p} \beta_i f_i(\mathbf{x}),$$

where $f_i(\mathbf{x})$ are regression functions (e.g. $f_0(\mathbf{x}) = 1$, $f_i(\mathbf{x}) = x_i$, $i = 1,\ldots,p$, specifies a *simple* linear model), and independent and identically distributed mean-zero Gaussian errors $\varepsilon(\mathbf{x})$

$$\varepsilon(\mathbf{x}) \sim N(0, 1/\lambda_\varepsilon),$$

where λ_ε is the precision (inverse variance) of the observational error process. In the following we will consider a statistical model for $\xi(\mathbf{x})$ that connects $\xi(\mathbf{x})$ to a "best" code calculation. In general applications, errors $\varepsilon(\mathbf{x})$ are often heteroscedastic or correlated. To keep this discussion general, we suppose the statistical models for $\xi(\mathbf{x})$ and $\varepsilon(\mathbf{x})$ are indexed by parameters $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$. For example, in the linear model specification above

$$\boldsymbol{\theta}_\xi = (\beta_{\mathbf{0}},\ldots,\beta_p) \text{ and } \theta_\varepsilon = \lambda_\varepsilon.$$

A Bayesian approach to statistical inference is taken in the following discussion. Given field data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ observed at design settings $(\mathbf{x}_1,\ldots,\mathbf{x}_n)$, and *a priori* knowledge about $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$ embodied in the joint prior density $\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon)$, the posterior density of $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$ is derived as follows

$$\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)) \propto L(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon),$$

where $L(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ is the *likelihood* function, i.e. the joint density of the data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ viewed as a function of the parameters $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$. The likelihood function provides the metric by which parameter settings are judged to be consistent with data: higher values indicate greater consistency. The posterior distribution updates prior knowledge about the parameters $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$ by conditioning on the data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$: parameter settings simultaneously consistent with the data and prior are more likely, with the likelihood component gaining influence as more field data are collected.

We turn our attention now to the problem of predicting reality $\xi(\mathbf{x})$ at unsampled input setting $\mathbf{x}$. Inference is based on the *predictive* density $\pi(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$,

$$\pi(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)) = \int_{\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon} \pi(\xi(\mathbf{x})|\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon,y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))\, d\boldsymbol{\theta}_\xi\, d\boldsymbol{\theta}_\varepsilon,$$

which represents the knowledge about $\xi(\mathbf{x})$ obtained by collecting field data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$. The first density in the integral, $\pi(\xi(\mathbf{x})|\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon,y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$, is derived directly from the statistical modeling assumptions. Often, given parameter settings $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$, $\xi(\mathbf{x})$ is independent of $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ so that

$$\pi(\xi(\mathbf{x})|\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon,y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)) = \pi(\xi(\mathbf{x})|\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon);$$

however, this need not be the case and, in fact, this simplification is not available to us in the main analysis to follow. The second density in the integral, $\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$, is the posterior distribution of $\boldsymbol{\theta}_\xi$ and $\boldsymbol{\theta}_\varepsilon$. In the more complicated settings of practical analyses, an analytic expression for $\pi(\boldsymbol{\theta}_\xi,\boldsymbol{\theta}_\varepsilon|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ is typically unavailable; in this event, techniques such as Markov chain Monte Carlo (MCMC) are employed to sample this posterior, and the predictive density is estimated by the mixture density

$$\pi(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)) \approx \frac{1}{M}\sum_{i=1}^{M} \pi(\xi(\mathbf{x})|\boldsymbol{\theta}_{\xi,i},\boldsymbol{\theta}_{\varepsilon,i},y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)).$$

It is natural to define predictive maturity in terms of achieving stability in the predictive density $\pi(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ as a function of the field data collected. Specifically, stability is reached when the collection of additional field data results in minimal changes to the predictive density as measured by an appropriate metric.

2.1. Information

One approach to measuring changes in probability density functions is through information gain. In Bayesian experiment

design (Chaloner and Verdinelli [15]), this is typically measured through the expected Kullback–Leibler distance between the posterior and the prior predictive distributions. In the subsequent development, we will utilize the following result.

Result 1. Suppose $\mathbf{y}=(\mathbf{y}_1^T,\mathbf{y}_2^T,\mathbf{y}_3^T)^T$ has a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix

$$\boldsymbol{\Sigma}=\begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13}\\ \boldsymbol{\Sigma}_{12}^{\mathbf{T}} & \boldsymbol{\Sigma}_{22} & \boldsymbol{\Sigma}_{23}\\ \boldsymbol{\Sigma}_{13}^{\mathbf{T}} & \boldsymbol{\Sigma}_{23}^{\mathbf{T}} & \boldsymbol{\Sigma}_{33}\end{pmatrix}.$$

Let $\pi_1(\mathbf{y}_1)=\pi(\mathbf{y}_1|\mathbf{y}_2)$ and $\pi_2(\mathbf{y}_1)=\pi(\mathbf{y}_1|\mathbf{y}_2,\mathbf{y}_3)$. Define the following matrices:

$$\boldsymbol{\Sigma}_{11.2}=\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{12}^T,$$
$$\boldsymbol{\Sigma}_{33.2}=\boldsymbol{\Sigma}_{33}-\boldsymbol{\Sigma}_{23}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{23},$$
$$\boldsymbol{\Sigma}_{13.2}=\boldsymbol{\Sigma}_{13}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{23}.$$

Then

$$E\left[\log\left(\frac{\pi_2}{\pi_1}\right)|\mathbf{y}_2\right]=\frac{1}{2}\left[\log(|\boldsymbol{\Sigma}_{11.2}|)-\log(|\boldsymbol{\Sigma}_{11.2}-\boldsymbol{\Sigma}_{13.2}\boldsymbol{\Sigma}_{33.2}^{-1}\boldsymbol{\Sigma}_{13.2}^T|)\right].$$

Alternative distance metrics have been employed to generate experiment designs. For example, Bingham and Chipman [16] utilize Hellinger distance in their proposed criterion for selecting an experiment design with two-level factors that maximizes the ability to discriminate between linear models.

Consider the following batch sequential algorithm for achieving predictive maturity:

1. Given field data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$, propose new input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ and define the *improvement*

   $$I_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)=-\log\left(\frac{f_{n+n_p}(\xi(\mathbf{x}))}{f_n(\xi(\mathbf{x}))}\right),$$

   where

   $$f_n(\xi(\mathbf{x}))=\pi\left(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)\right) \text{ and}$$
   $$f_{n+n_p}(\xi(\mathbf{x}))=\pi\left(\xi(\mathbf{x})|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n),y(\mathbf{x}_1^*),\ldots,y(\mathbf{x}_{n_p}^*)\right).$$

   Compute the *posterior expected improvement*

   $$EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)=-E\left[\log\left(\frac{f_{n+n_p}(\xi(\mathbf{x}))}{f_n(\xi(\mathbf{x}))}\right)|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)\right].$$

2. Determine settings $\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+n_p}$ minimizing $\max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)$:

   $$EI_n(\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+n_p})=\underset{\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*}{\operatorname{argmin}}\ \max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*).$$

   Collect field data $y(\mathbf{x}_{n+1}),\ldots,y(\mathbf{x}_{n+n_p})$ and set n to $n+n_p$.
3. Continue steps (1)–(2) until changes in $EI_n(\cdot)$ become negligible with respect to an absolute or relative stopping criterion.

This procedure will be referred to as the Expected Improvement for Predictive Stability (EIPS) algorithm. Given input setting $\mathbf{x}^*$, $\max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}^*)$ is the largest entropy between the current and proposed predictive densities for $\xi(\mathbf{x})$. Ideally, we would like this entropy to be as close to zero as possible, indicating that little information is gained by collecting new field data for predicting $\xi(\mathbf{x})$. On the other hand, larger values of this metric correspond to greater instability, and so choosing $\mathbf{x}_{n+1}$ to minimize $EI_n(\mathbf{x}|\mathbf{x}^*)$ results in the largest possible one-step reduction in instability. The EIPS algorithm possesses the desirable feature that it uses the predictive distribution itself, rather than summary metrics such as moments, to make inference regarding how best to collect new field data.

2.2. Moments

As an alternative, we consider sequential algorithms based on expected improvement criteria for global prediction of response surfaces. For example, define the improvement

$$I_n(\mathbf{x})=|\xi(\mathbf{x})-\xi_n|^g,$$

where g is a positive integer and $\xi_n=\{\xi(\mathbf{x}_j):\ \mathbf{x}_j \text{ is closest to } \mathbf{x},\ j=1,\ldots,n\}$. The "closeness" of $\mathbf{x}_j$ to $\mathbf{x}$ is measured via a distance metric, such as Euclidean or Mahalanobis distance. This is an integer power of the absolute difference between $\xi(\cdot)$ evaluated at input setting $\mathbf{x}$ and at the nearest previously sampled input setting. We propose adding the batch of input sites $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ to the current design. The proposed algorithm requires the *posterior expected improvement*:

$$EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)=E\left[\left|\xi(\mathbf{x})-\xi_{n+n_p}\right|^g|y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n)\right],$$

where $\xi_{n+n_p}=\left\{\xi(\boldsymbol{\chi}_j):\boldsymbol{\chi}_j \text{ is closest to } \mathbf{x},\ j=1,\ldots,n+n_p\right\}$ and $\boldsymbol{\chi}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*\}$.

To understand the behavior of the expected improvement, note that it partitions the input space into $n+n_p$ disjoint regions. Region j consists of all $\mathbf{x}$ "closer to" $\boldsymbol{\chi}_j$ than any other sampled input setting, for $j=1,\ldots,n+n_p$. This criterion is bounded below by zero and takes larger values if either of the following circumstances occurs:

1. Large prediction variance in the difference $\xi(\mathbf{x})-\xi(\boldsymbol{\chi}_j)$ for any $\mathbf{x}$ in region j, suggesting some predictions are imprecise given current field data.
2. Rapid changes in predictions on short length scales measured by the squared difference in predictions at $\mathbf{x}$ and $\boldsymbol{\chi}_j$ in region j, suggesting the possibility of biased prediction resulting from undersampling this region.

This suggests collecting field data at input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ minimizing $\max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)$, in order to mitigate either or both of the circumstances above that can lead to prediction errors. This leads to the following batch sequential algorithm for achieving predictive maturity:

1. Given field data $(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$, propose new input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ and compute $EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)$.
2. Determine settings $\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+n_p}$ minimizing $\max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*)$:

   $$EI_n(\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+n_p})=\underset{\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*}{\operatorname{argmin}}\ \max_{\mathbf{x}} EI_n(\mathbf{x}|\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*).$$

   Collect field data $y(\mathbf{x}_{n+1}),\ldots,y(\mathbf{x}_{n+n_p})$ and set n to $n+n_p$.
3. Continue steps (1)–(2) until changes in $EI_n(\cdot)$ become negligible with respect to an absolute or relative stopping criterion.

This algorithm generalizes the Expected Improvement for Global Fit (EIGF) algorithm of Lam and Notz [17], which was developed for one-step sequential additions of new deterministic computer model runs for the purpose of efficiently achieving good response surface surrogates for prediction at unsampled model input settings. Hence, we will refer to this procedure as the

Generalized EIGF, or GEIGF, algorithm. Note that increasing the value of g produces a more global search.

In the subsequent development, we will utilize the following result.

Result 2. Suppose $\mathbf{y}=(\mathbf{y}_1^T,\mathbf{y}_2^T,\mathbf{y}_3^T)^T$ has a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix

$$\boldsymbol{\Sigma}=\begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13}\\ \boldsymbol{\Sigma}_{12}^{\mathbf{T}} & \boldsymbol{\Sigma}_{22} & \boldsymbol{\Sigma}_{23}\\ \boldsymbol{\Sigma}_{13}^{\mathbf{T}} & \boldsymbol{\Sigma}_{23}^{\mathbf{T}} & \boldsymbol{\Sigma}_{33}\end{pmatrix}.$$

Let $I_{1,g}^{(j)}=\sum_{i=1}^{n_1}|y_{1i}-y_{2i,j}|^g$ and $I_{3,g}^{(j)}=\sum_{i=1}^{n_1}|y_{1i}-y_{3i,j}|^g$, where n_1 is the length of $\mathbf{y}_1$, g is a positive integer, and j indexes a block of $\mathbf{y}_2$ or $\mathbf{y}_3$ having length equal to n_1. Define the following matrices:

$$\boldsymbol{\Sigma}_{11.2}=\boldsymbol{\Sigma}_{11}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{12}^T,$$
$$\boldsymbol{\Sigma}_{33.2}=\boldsymbol{\Sigma}_{33}-\boldsymbol{\Sigma}_{23}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{23},$$
$$\boldsymbol{\Sigma}_{13.2}=\boldsymbol{\Sigma}_{13}-\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{23},$$
$$\boldsymbol{\Sigma}_{(1-3).2}^{j}=\boldsymbol{\Sigma}_{11.2}-\boldsymbol{\Sigma}_{13.2}^{j}-(\boldsymbol{\Sigma}_{13.2}^{j})^T+\boldsymbol{\Sigma}_{33.2}^{j},$$

where $\boldsymbol{\Sigma}_{13.2}^{j}$ and $\boldsymbol{\Sigma}_{33.2}^{j}$ denote submatrices corresponding to the j-th block of $\mathbf{y}_3$. Then

$$E[I_{h,g}^{(j)}|\mathbf{y}_2]=\sum_{i=1}^{n_1}(s_{hi|2}^2)^{g/2}\sum_{k=0}^{g}\binom{g}{k}z_{hi|2}^k\left[(-1)^k m_{g-k}(z_{hi|2})+m_{g-k}(-z_{hi|2})\right],$$

$h=1,3,$

where $s_{1i|2}^2=((\boldsymbol{\Sigma}_{11.2}))_{ii}$ (the (i,i) element of $\boldsymbol{\Sigma}_{11.2}$) and $s_{3i|2}^2=\left(\left(\boldsymbol{\Sigma}_{(1-3).2}^{j}\right)\right)_{ii}$; $z_{1i|2}=(y_{2i,j}-\hat{y}_{1i})/s_{1i|2}$ and $z_{3i|2}=(\hat{y}_{3i,j}-\hat{y}_{1i})/\ s_{3i|2}$ for $\hat{\mathbf{y}}_1=\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\mathbf{y}_2$ and $\hat{\mathbf{y}}_3=\boldsymbol{\Sigma}_{23}^T\boldsymbol{\Sigma}_{22}^{-1}\mathbf{y}_2$; and

$$m_\ell(z)=z^{\ell-1}\phi(z)+(\ell-1)m_{\ell-2}(z),$$

for $\ell\geq 2$ a positive integer, $m_0(z)=1-\Phi(z)$, $m_1(z)=\phi(z)$, and $\phi(z)$, $\Phi(z)$ denote the standard Gaussian probability density and cumulative distribution functions, respectively.

When proposing a batch of new input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$, the maximization of expected improvement in the EIPS or GEIGF algorithms can be replaced by an integration of expected improvement over the input space with respect to a weight function $w(\mathbf{x})$, which we take to be 1 in the subsequent development. However, this function could be chosen to put more weight on parts of the input space where output is more sensitive to input changes, for example.

3. Calibration and prediction of computer models

Kennedy and O'Hagan [2] proposed what has become a standard statistical framework for calibrating uncertain parameters in computer models to available field data, and analyzing the resulting calibrated predictions for the purposes of validating a computer model or certifying an engineering system, for example. Let $\eta(\mathbf{z},\mathbf{t})$ denote the computer model evaluated at design settings $\mathbf{z}$ and physics model settings $\mathbf{t}$. Note the design parameters $\mathbf{z}$ in the computer model match up one to one with the design parameters $\mathbf{x}$ specifying experiments —different notation is used to make it clear there is no requirement that experiments and computer model runs be collected at the exact same design settings. The parameters $\mathbf{t}$ are specific to physics models embedded in the code. They may or may not have corresponding "true" (but unknown) values in nature. One feature distinguishing them from design settings is that they cannot be controlled for the purposes of collecting field data. Let $\mathbf{t}=\boldsymbol{\theta}$ denote the unknown setting of the physics parameters that provides the best match of code calculation to physical reality. Let the discrepancy $\delta(\mathbf{x})$ be defined as the difference between reality and the best code calculation

$$\delta(\mathbf{x})=\xi(\mathbf{x})-\eta(\mathbf{x},\boldsymbol{\theta}).$$

We assume $\delta(\mathbf{x})$ is a mean-zero Gaussian process (GP) with correlation length parameters $\boldsymbol{\rho}_\delta$ and precision parameter λ_δ. Let $r_\delta(\mathbf{x}_1-\mathbf{x}_2)$ denote the correlation function of this GP. We assume a product power exponential form for the covariance function

$$c_\delta(\mathbf{x}_1-\mathbf{x}_2)=\frac{1}{\lambda_\delta}r_\delta(\mathbf{x}_1-\mathbf{x}_2),$$
$$r_\delta(\mathbf{x}_1-\mathbf{x}_2)=\prod_{j=1}^{p}\rho_{\delta j}^{4(x_{1j}-x_{2j})^2},\quad \rho_{\delta j}=\exp(-\beta_{\delta j}/4),$$

where $\rho_{\delta j}$ represents the correlation between discrepancies with inputs at the same levels except in the j-th dimension, where the inputs are separated by the midrange value of 0.5, assuming all inputs have been scaled to the unit cube. Higdon et al. [18] and Williams et al. [19] provide details on this parameterization of Gaussian process models and additional references.

If the code is "fast", i.e. computationally inexpensive, no statistical model for $\eta(\mathbf{z},\mathbf{t})$ is required as code runs can be obtained as needed. If the code is "slow", a stochastic model must be specified for $\eta(\mathbf{z},\mathbf{t})$ to allow prediction of unsampled input settings $(\mathbf{z},\mathbf{t})$. In this setting, we assume a mean-zero Gaussian process for $\eta(\mathbf{z},\mathbf{t})$, a priori independent of $\delta(\mathbf{x})$, indexed by correlation length parameters $\boldsymbol{\rho}_\eta$ and precision parameter λ_η. Let $r_\eta((\mathbf{z}_1,\mathbf{t}_1)-(\mathbf{z}_2,\mathbf{t}_2))$ denote the correlation function of this GP. As before, we assume a product power exponential form for the covariance function

$$c_\eta((\mathbf{z}_1,\mathbf{t}_1)-(\mathbf{z}_2,\mathbf{t}_2))=\frac{1}{\lambda_\eta}r_\eta((\mathbf{z}_1,\mathbf{t}_1)-(\mathbf{z}_2,\mathbf{t}_2)),$$
$$r_\eta((\mathbf{z}_1,\mathbf{t}_1)-(\mathbf{z}_2,\mathbf{t}_2))=\prod_{j=1}^{p}\rho_{\eta j}^{4(z_{1j}-z_{2j})^2}\prod_{j=1}^{q}\rho_{\eta j+p}^{4(t_{1j}-t_{2j})^2},$$
$$\rho_{\eta,j}=\exp(-\beta_{\eta j}/4).$$

The covariance and correlation functions restricted to the $\mathbf{z}$ inputs are to be interpreted as follows: $c_\eta(\mathbf{z}_1-\mathbf{z}_2)=c_\eta((\mathbf{z}_1,\mathbf{t})-(\mathbf{z}_2,\mathbf{t}))$ and $r_\eta(\mathbf{z}_1-\mathbf{z}_2)=r_\eta((\mathbf{z}_1,\mathbf{t})-(\mathbf{z}_2,\mathbf{t}))$.

We will proceed under the assumption of a "slow" code. To complete the model specification, we will assume the observation errors $\varepsilon(\mathbf{x})$ are mean-zero, Gaussian noise with precision λ_ε, independent of $\boldsymbol{\eta}(\mathbf{z},\mathbf{t})$ and $\delta(\mathbf{x})$.

We assume a prior distribution for the parameters

$$\pi(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon)=\pi(\boldsymbol{\theta})\pi(\boldsymbol{\rho}_\eta)\pi(\lambda_\eta)\pi(\boldsymbol{\rho}_\delta)\pi(\lambda_\delta)\pi(\lambda_\varepsilon).$$

The prior distribution on the best unknown physics parameter setting $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$, is generally derived from a combination of expert judgment and analysis of relevant separate (or integral) effects data. See Higdon et al. [18] for standard prior assignments to the Gaussian process model parameters. Given field data $\mathbf{y}^n=(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n))$ and computer model runs $\boldsymbol{\eta}^m=(\eta(\mathbf{z}_1,\mathbf{t}_1),\ldots,\eta(\mathbf{z}_m,\mathbf{t}_m))$, the posterior distribution of all the parameters is derived from

$$\pi(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon|\boldsymbol{\eta}^m,\mathbf{y}^n)\propto L(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon|\boldsymbol{\eta}^m,\mathbf{y}^n)\pi(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon)$$
$$L(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon|\boldsymbol{\eta}^m,\mathbf{y}^n)\propto L(\boldsymbol{\theta},\boldsymbol{\rho}_\eta,\lambda_\eta,\boldsymbol{\rho}_\delta,\lambda_\delta,\lambda_\varepsilon,\boldsymbol{\eta}^m|\mathbf{y}^n)L(\boldsymbol{\rho}_\eta,\lambda_\eta|\boldsymbol{\eta}^m).$$

The likelihood of the observed field data and calculations is represented as the product of two terms: the first is the likelihood of the observed field data given the calculations, and the second is the likelihood of the calculations. This decomposition is

important, as the second term can be computationally intensive when the number of calculations m is large. Thus computational considerations may dictate fixing $(\boldsymbol{\rho}_\eta,\lambda_\eta)$ at reasonable values (e.g. a maximum likelihood estimate) and reducing the complete likelihood to the more computationally tractable first term (assuming the amount of field data, n, is relatively small). As this posterior distribution is analytically intractable, samples from it are taken using MCMC methods.

We will track predictions for the discrepancy $\pi(\delta(\mathbf{x})|\boldsymbol{\eta}^m,\mathbf{y}^n)$ in the proposed predictive maturity algorithms. As mentioned previously, predictions of physical reality $\pi(\xi(\mathbf{x})|\boldsymbol{\eta}^m,\mathbf{y}^n)$ and calibrated code $\pi(\eta(\mathbf{x},\boldsymbol{\theta})|\boldsymbol{\eta}^m,\mathbf{y}^n)$ could be tracked using simple modifications of the same algorithms. We assume that a high-quality code surrogate is available prior to collecting new field data, although the algorithm could be modified to simultaneously accomplish that goal. We propose adapting the approach of Loeppky et al. [14] for obtaining stability in code surrogate performance (emulator maturity) to achieving the goal of predictive maturity with calibrated computer models:

1. Assume initial field data and computer model runs have been collected. If possible, algorithms for achieving emulator maturity have been run with the computer model. A calibration is performed using these initial field data and model runs.
2. Apply a sequential design criterion to the discrepancy $\delta(\mathbf{x})$ for the purpose of sequentially choosing a batch of design settings at which new field experiments will be conducted, using estimates of the relevant statistical model parameters from the current calibration.
3. If desired, pair up the new design sites obtained by the process in step (2) with physics parameter settings $\mathbf{t}$ to obtain new computer model runs. For example, physics parameter settings could be chosen according to some space-filling criterion or from the results of the current calibration, perhaps based on sampling the $\boldsymbol{\theta}$ posterior distribution.
4. Perform a new calibration that incorporates the new field data resulting from step (2), and the new computer model runs from step (3). Repeat steps (2)–(3) until terminating the algorithm based on a stopping rule.

3.1. Expected improvement for calibrated computer models

We now specialize Results 1 and 2 to the Kennedy and O'Hagan [2] calibration framework. The adoption of GP models in this framework allows the assumptions of these results to be satisfied. Let $D_1=\{\mathbf{d}_{11},\ldots,\mathbf{d}_{1n_1}\}$ and $D_2=\{\mathbf{d}_{21},\ldots,\mathbf{d}_{2n_2}\}$ denote two designs. The notation $\mathbf{R}(D_1,D_2)$ denotes a $n_1\times n_2$ matrix of correlations with (i,j) element $r(\mathbf{d}_{1i}-\mathbf{d}_{2j})$. Let $D_y=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, $D_\eta=\{(\mathbf{z}_1,\mathbf{t}_1),\ldots,(\mathbf{z}_m,\mathbf{t}_m)\}$, $D_\eta^{\boldsymbol{\theta}}=\{(\mathbf{x}_1,\boldsymbol{\theta}),\ldots,(\mathbf{x}_n,\boldsymbol{\theta})\}$, $D_\xi=\{\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*\}$, and $D_\xi^{\boldsymbol{\theta}}=\{(\mathbf{x}_{n_p}^*,\boldsymbol{\theta}),\ldots,(\mathbf{x}_{n_p}^*,\boldsymbol{\theta})\}$. The EIPS criterion is given by $-E\left[\log(\pi_2/\pi_1)|\mathbf{y}_2\right]$ in Result 1 with the following substitutions

1. $\mathbf{y}_1=\delta(\mathbf{x})$, $\mathbf{y}_2=(y(\mathbf{x}_1),\ldots,y(\mathbf{x}_n),\eta(\mathbf{z}_1,\mathbf{t}_1),\ldots,\eta(\mathbf{z}_m,\mathbf{t}_m))$, and $\mathbf{y}_3=(\xi(\mathbf{x}_1^*),\ldots,\xi(\mathbf{x}_{n_p}^*))$;

2. $\boldsymbol{\Sigma}_{11}=1/\lambda_\delta$, $\boldsymbol{\Sigma}_{12}=(\mathbf{R}_\delta(\{\mathbf{x}\},D_y)/\lambda_\delta\quad\mathbf{0}^T)$, $\boldsymbol{\Sigma}_{13}=\mathbf{R}_\delta(\{\mathbf{x}\},D_\xi)/\lambda_\delta$,
$$\boldsymbol{\Sigma}_{22}=\begin{pmatrix}\mathbf{R}_\eta(D_y,D_y)/\lambda_\eta+\mathbf{R}_\delta(D_y,D_y)/\lambda_\delta+\boldsymbol{\Lambda}/\lambda_\varepsilon & \mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)/\lambda_\eta\\ \mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)^T/\lambda_\eta & \mathbf{R}_\eta(D_\eta,D_\eta)/\lambda_\eta\end{pmatrix},$$
where
$\boldsymbol{\Lambda}=\mathrm{Diag}(\sigma_i^2,i=1,\ldots,n)$,
$$\boldsymbol{\Sigma}_{23}=\begin{pmatrix}\mathbf{R}_\eta(D_y,D_\xi)/\lambda_\eta+\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta\\ \mathbf{R}_\eta(D_\eta,D_\xi^{\boldsymbol{\theta}})/\lambda_\eta\end{pmatrix},\ \text{and}$$
$$\boldsymbol{\Sigma}_{33}=\mathbf{R}_\eta(D_\xi,D_\xi)/\lambda_\eta+\mathbf{R}_\delta(D_\xi,D_\xi)/\lambda_\delta,$$
where σ_i^2 is the assumed variance of $y(\mathbf{x}_i)$ (if available);
3. $\boldsymbol{\Sigma}_{11.2}=(1/\lambda_\delta)\left(1-tr[\mathbf{R}_\delta(\{\mathbf{x}\},D_y)^T\mathbf{R}_\delta(\{\mathbf{x}\},D_y)\mathbf{S}_{11}^{-1}]/\lambda_\delta\right)$

where
$$\mathbf{S}_{11}=\mathbf{R}_\eta(D_y,D_y)-\mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)\mathbf{R}_\eta^{-1}(D_\eta,D_\eta)\mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)^T/\lambda_\eta+\mathbf{R}_\delta(D_y,D_y)/\lambda_\delta+\boldsymbol{\Lambda}/\lambda_\varepsilon,$$
$$\boldsymbol{\Sigma}_{33.2}=\mathbf{T}_{\eta\xi\xi}/\lambda_\eta+\mathbf{R}_\delta(D_\xi,D_\xi)/\lambda_\delta-(\mathbf{T}_{\eta y\xi}/\lambda_\eta+\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta)^T\mathbf{S}_{11}^{-1}(\mathbf{T}_{\eta y\xi}/\lambda_\eta+\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta)$$
where
$$\mathbf{T}_{\eta\xi\xi}=\mathbf{R}_\eta(D_\xi,D_\xi)-\mathbf{R}_\eta(D_\eta,D_\xi^{\boldsymbol{\theta}})^T\mathbf{R}_\eta^{-1}(D_\eta,D_\eta)\mathbf{R}_\eta(D_\eta,D_\xi^{\boldsymbol{\theta}})\ \text{and}$$
$$\mathbf{T}_{\eta y\xi}=\mathbf{R}_\eta(D_y,D_\xi)-\mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)\mathbf{R}_\eta^{-1}(D_\eta,D_\eta)\mathbf{R}_\eta(D_\eta,D_\xi^{\boldsymbol{\theta}}),\ \text{and}$$
$$\boldsymbol{\Sigma}_{13.2}=(\mathbf{R}_\delta(\{\mathbf{x}\},D_\xi)-\mathbf{R}_\delta(\{\mathbf{x}\},D_y)\mathbf{S}_{11}^{-1}(\mathbf{T}_{\eta y\xi}/\lambda_\eta+\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta))/\lambda_\delta.$$

The EIPS criterion is easily extended to functional computer and experimental outputs (e.g. time series or collections of features) in the Kennedy and O'Hagan framework. Functional computer output is modeled as follows:

$$\boldsymbol{\eta}(\mathbf{z},\mathbf{t})=\mathbf{K}\mathbf{w}(\mathbf{z},\mathbf{t}),$$

where $\mathbf{K}$ is a matrix of fixed basis vectors (e.g. subset of eigenvectors selected from principal component analysis of a suite of computer model runs) and $\mathbf{w}$ is the vector of corresponding coefficients. The functional discrepancy is modeled similarly,

$$\boldsymbol{\delta}(\mathbf{x})=\mathbf{D}\mathbf{v}(\mathbf{x}),$$

where $\mathbf{D}$ is a matrix of fixed basis vectors (e.g. Gaussian kernels) and $\mathbf{v}$ is the vector of corresponding coefficients. Following Higdon et al. [20], we assume the coefficients of basis decompositions in both the computer model and discrepancy are modeled as *a priori* independent GPs. Let $\mathbf{v}_i(D_1)=(v_i(\mathbf{d}_{11}),\ldots,v_i(\mathbf{d}_{1n_1}))$ denote the i-th discrepancy coefficient evaluated on the design D_1, and $\mathbf{w}_i(D_2)=(w_i(\mathbf{d}_{21}),\ldots,w_i(\mathbf{d}_{2n_2}))$ denote the i-th model coefficient evaluated on D_2. This extension is accomplished by setting $\mathbf{y}_1=(v_1(\mathbf{x}),\ldots,v_{p_\delta}(\mathbf{x}))$, $\mathbf{y}_2=(\hat{\mathbf{v}},\hat{\mathbf{u}},\hat{\mathbf{w}})$ where $(\mathbf{v},\mathbf{u},\mathbf{w})=(\mathbf{v}_1(D_y),\ldots,\mathbf{v}_{p_\delta}(D_y),\mathbf{w}_1(D_\eta^{\boldsymbol{\theta}}),\ldots,\mathbf{w}_{p_\eta}(D_\eta^{\boldsymbol{\theta}}),\mathbf{w}_1(D_\eta),\ldots,\mathbf{w}_{p_\eta}(D_\eta))$, and $\mathbf{y}_3=(\mathbf{v}_1(D_\xi),\ldots,\mathbf{v}_{p_\delta}(D_\xi),\mathbf{w}_1(D_\xi^{\boldsymbol{\theta}}),\ldots,\mathbf{w}_{p_\eta}(D_\xi^{\boldsymbol{\theta}}))$. Here, p_δ and p_η are the total number of discrepancy and model coefficients, and $\mathbf{y}_2$ is the least squares estimate of the basis coefficients found by projecting the complete vector of experimental data and model outputs onto the basis as described by Higdon et al. [20].

The GEIGF criterion is given by $E[I_{3,g}|\mathbf{y}_2]$ in Result 2 with the above substitutions, modified as follows:

1. $\mathbf{y}_3=(\delta(\mathbf{x}),\ldots,\delta(\mathbf{x}_n),\delta(\mathbf{x}_1^*),\ldots,\delta(\mathbf{x}_{n_p}^*))$;
2. $\boldsymbol{\Sigma}_{13}=(\mathbf{R}_\delta(\{\mathbf{x}\},D_y)/\lambda_\delta\quad\mathbf{R}_\delta(\{\mathbf{x}\},D_\xi)/\lambda_\delta)$,
$$\boldsymbol{\Sigma}_{23}=\begin{pmatrix}\mathbf{R}_\delta(D_y,D_y)/\lambda_\delta & \mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\ \text{and}$$
$$\boldsymbol{\Sigma}_{33}=\begin{pmatrix}\mathbf{R}_\delta(D_y,D_y)/\lambda_\delta & \mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta\\ \mathbf{R}_\delta(D_y,D_\xi)^T/\lambda_\delta & \mathbf{R}_\delta(D_\xi,D_\xi)/\lambda_\delta\end{pmatrix};$$
3. $\boldsymbol{\Sigma}_{33.2}=\begin{pmatrix}\mathbf{T}_{\delta yy} & \mathbf{T}_{\delta y\xi}\\ \mathbf{T}_{\delta y\xi}^T & \mathbf{T}_{\delta\xi\xi}\end{pmatrix}$ where
$\mathbf{T}_{\delta yy}=(1/\lambda_\delta)(\mathbf{I}_n-\mathbf{R}_\delta(D_y,D_y)\mathbf{S}_{11}^{-1}/\lambda_\delta)\mathbf{R}_\delta(D_y,D_y)$,
$\mathbf{T}_{\delta y\xi}=(1/\lambda_\delta)(\mathbf{I}_n-\mathbf{R}_\delta(D_y,D_y)\mathbf{S}_{11}^{-1}/\lambda_\delta)\mathbf{R}_\delta(D_y,D_\xi)$ and
$\mathbf{T}_{\delta\xi\xi}=(1/\lambda_\delta)(\mathbf{R}_\delta(D_\xi,D_\xi)-\mathbf{R}_\delta(D_y,D_\xi)^T\mathbf{S}_{11}^{-1}\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta)$, and

$$\boldsymbol{\Sigma}_{13.2} = \frac{1}{\lambda_\delta}\big(\mathbf{R}_\delta(\{\mathbf{x}\},D_y)(\mathbf{I}_n - \mathbf{S}_{11}^{-1}\mathbf{R}_\delta(D_y,D_y)/\lambda_\delta) \\ \mathbf{R}_\delta(\{\mathbf{x}\},D_\xi) - \mathbf{R}_\delta(\{\mathbf{x}\},D_y)\mathbf{S}_{11}^{-1}\mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta\big);$$

4. $\hat{y}_1 = \mathbf{R}_\delta(\{\mathbf{x}\},D_y)\mathbf{S}_{11}^{-1}(\mathbf{y}^n - \mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)\mathbf{R}_\eta^{-1}(D_\eta,D_\eta)\boldsymbol{\eta}^m)/\lambda_\delta$ and
$\hat{\mathbf{y}}_3 = \begin{pmatrix}\mathbf{R}_\delta(D_y,D_y)\\ \mathbf{R}_\delta(D_y,D_\xi)^T\end{pmatrix}\mathbf{S}_{11}^{-1}(\mathbf{y}^n - \mathbf{R}_\eta(D_\eta^{\boldsymbol{\theta}},D_\eta)\mathbf{R}_\eta^{-1}(D_\eta,D_\eta)\boldsymbol{\eta}^m)/\lambda_\delta.$

We take $\mathbf{y}_3 = (\mathbf{v}_1(D_y),\ldots,\mathbf{v}_{p_\delta}(D_y),\mathbf{v}_1(D_\xi),\ldots,\mathbf{v}_{p_\delta}(D_\xi))$ in the extension of this criterion to the functional data setting.

We examine the GEIGF criterion with two choices of g, 2 and 4. Lam [21] chose $g=2$ as a standard setting, but noted that $g=4$ produced better results for some examples where a more global search is required.

Comments on the proposed algorithm:

1. The discrepancy was chosen as the focus for generating new field data, motivated by the fact that achieving stability in the discrepancy is critical in order that use of the discrepancy as an empirical correction to the computer model be deemed acceptable for interpolative, and possibly extrapolative, predictions required in validation or certification contexts. However, expected improvements for both physical reality and calibrated code can be tracked by making simple modifications to the expected improvement for the discrepancy and, in fact, the algorithm could be modified to generate updates based on one or a mixture of these alternatives.
2. Code updates may make use of knowledge gained from the calibration about the region of physics parameter space most relevant to matching physical reality. The focus of emulator maturity is on ensuring a quality global fit to the computer model output, which will be beneficial to calibration by reducing both bias and variability in the code surrogate and thus more rapidly leading to identification of the relevant region of parameter space.
3. The algorithms as presented are designed to accommodate batch updates. However, the multivariate optimization required to add all proposed inputs in a batch at once quickly becomes computationally challenging. Hence the inputs are added using a modified Fedorov exchange algorithm (Fedorov [22]), which performs a greedy optimization over each proposed run in the batch while fixing the others, cycling through the proposed runs until negligible improvement in the criterion value is observed. GP parameters are not updated until a batch is completed.
4. The requirement that candidate design settings from step (2) of the proposed sequential design algorithm be fixed for generating the candidate computer model run in step (3) can be relaxed so that different candidate design settings are allowed in steps (2) and (3). However, the algorithm reflects the typical situation that in practice, code runs are often desired at the design settings for which field data are available or proposed.

3.2. Mean square error-based criteria

Neither the EIPS nor the GEIGF criteria submit to closed-form integration over the input domain with respect to standard weight functions such as the uniform or Gaussian density functions. Integrals must be estimated using standard numerical integration techniques such as Monte Carlo sampling or quadrature (Genz and Malik [23], O'Hagan [24]). If ease of integration over the input domain is of concern, one could consider batch updates using the integrated mean square error criterion for discrepancy, IMSE-δ. Lu et al. [25] develop this criterion in terms of predicting physical reality $\xi(\mathbf{x})$ (rather than discrepancy $\delta(\mathbf{x})$) for the purpose of recommending follow-up computer model runs and field data experiments simultaneously. The IMSE-δ criterion chooses candidate input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ to minimize the integrated posterior variance of the discrepancy function, $\int_R Var(\delta(\mathbf{x})|\mathbf{y}^n,\boldsymbol{\eta}^m,\xi(\mathbf{x}_1^*),\ldots,\xi(\mathbf{x}_{n_p}^*))w(\mathbf{x})\,d\mathbf{x}$, for rectangular region R. The posterior variance is given by

$$Var(\delta(\mathbf{x})|\mathbf{y}^n,\boldsymbol{\eta}^m,\xi(\mathbf{x}_1^*),\ldots,\xi(\mathbf{x}_{n_p}^*)) = \boldsymbol{\Sigma}_{11.2} - \boldsymbol{\Sigma}_{13.2}\boldsymbol{\Sigma}_{33.2}^{-1}\boldsymbol{\Sigma}_{13.2}^T,$$

where the matrices on the right-hand side are as specified for the EIPS criterion in Section 3.1. This can be easily integrated with respect to some common weight functions. We will demonstrate this for $w(\mathbf{x})=1$ on the unit cube. Let $\psi(\chi,\omega;l,u) = \sqrt{(\pi/\omega)}[\Phi(\sqrt{2\omega}(u-\chi)) + \Phi(\sqrt{2\omega}(\chi-l)) - 1]$ where $\Phi(\cdot)$ denotes the standard Gaussian cumulative distribution function. The following result is required.

Result 3.

$$\int_{l_k}^{u_k} \exp\left[-\beta_k(x_k - x_{ik})^2 - \beta_k(x_k - x_{jk}^*)^2\right] dx_k \\ = \exp\left[-\frac{1}{2}\beta_k(x_{ik} - x_{jk}^*)^2\right]\psi\left(\frac{1}{2}(x_{ik}+x_{jk}^*), 2\beta_k; l_k, u_k\right).$$

Table 1 defines three matrices needed in the calculation of IMSE-δ.

Applying Result 3 on the unit cube, we obtain

$$\int_{[0,1]^p} var\left(\delta(\mathbf{x})|\mathbf{y}^n,\boldsymbol{\eta}^m,\xi(\mathbf{x}_1^*),\ldots,\xi(\mathbf{x}_{n_p}^*)\right) d\mathbf{x} = \\ \left(1 - (tr[\boldsymbol{\Psi}_{\delta y}(\mathbf{S}_{11}^{-1} + \mathbf{S}_{y\xi}\boldsymbol{\Sigma}_{33.2}^{-1}\mathbf{S}_{y\xi}^T)] + tr[(\boldsymbol{\Psi}_{\delta\xi} - 2\boldsymbol{\Psi}_{\delta\xi y}\mathbf{S}_{y\xi})\boldsymbol{\Sigma}_{33.2}^{-1}])/\lambda_\delta\right)/\lambda_\delta,$$

where $\mathbf{S}_{y\xi} = \mathbf{S}_{11}^{-1}\left(\mathbf{T}_{\eta y\xi}/\lambda_\eta + \mathbf{R}_\delta(D_y,D_\xi)/\lambda_\delta\right)$. The posterior variance can also be used in the MMSE-δ criterion, which chooses candidate input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ to minimize the maximum posterior variance of the discrepancy function. Sacks and Schiller [26] compared IMSE and MMSE criteria for spatial design applications, while Sacks et al. ([27,28]) considered these criteria in the context of computer experiment design.

The functional extensions of these criteria take $\mathbf{y}_3$ to be equivalent to physical reality, $\mathbf{y}_3 = (\mathbf{v}_1(D_\xi),\ldots,\mathbf{v}_{p_\delta}(D_\xi),\ \mathbf{w}_1(D_\xi^{\boldsymbol{\theta}}),\ldots,\mathbf{w}_{p_\eta}(D_\xi^{\boldsymbol{\theta}}))$.

3.3. Entropy

Shewry and Wynn [29] developed a maximum entropy criterion for sampling and experiment design. Given a finite system, the motivation of this criterion is to select a sample from the system to maximize the information for predicting unsampled

Table 1
Matrices needed for IMSE-δ calculation.

Matrix	Size	(i,j) element
$\boldsymbol{\Psi}_{\delta y}$	$n \times n$	$\prod_{k=1}^p \exp\left[-\frac{1}{2}\beta_{\delta,k}(x_{ik}-x_{jk})^2\right]\psi\left(\frac{1}{2}(x_{ik}+x_{jk}), 2\beta_{\delta,k}; 0, 1\right)$
$\boldsymbol{\Psi}_{\delta\xi}$	$n_p \times n_p$	$\prod_{k=1}^p \exp\left[-\frac{1}{2}\beta_{\delta,k}(x_{ik}^*-x_{jk}^*)^2\right]\psi\left(\frac{1}{2}(x_{ik}^*+x_{jk}^*), 2\beta_{\delta,k}; 0, 1\right)$
$\boldsymbol{\Psi}_{\delta\xi y}$	$n_p \times n$	$\prod_{k=1}^p \exp\left[-\frac{1}{2}\beta_{\delta,k}(x_{ik}^*-x_{jk})^2\right]\psi\left(\frac{1}{2}(x_{ik}^*+x_{jk}), 2\beta_{\delta,k}; 0, 1\right)$

system values given the sample. The ME-δ criterion chooses candidate input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ to maximize $|\boldsymbol{\Sigma}_{33.2}|$, where this matrix is specified as for the EIPS criterion in Section 3.1 with $\mathbf{y}_3=(\delta(\mathbf{x}_1^*),\ldots,\delta(\mathbf{x}_{n_p}^*))$ in the scalar output case and $\mathbf{y}_3=(\mathbf{v}_1(D_\xi),\ldots,\mathbf{v}_{p_\delta}(D_\xi))$ in the functional output case.

3.4. Distance-based criteria

Two additional batch sequential design criteria applied to discrepancy will be considered: maximin Euclidean distance and maximin sensitivity-weighted distance. Let $\ell_{\boldsymbol{\tau}}(\mathbf{x}_1,\mathbf{x}_2)=\sqrt{\sum_{i=1}^{p}\tau_i(x_{1i}-x_{2i})^2}$ denote Mahalanobis distance with mean $\mathbf{0}$ and standard deviations $1/\sqrt{\tau_i}$. Consider the following distance metric involving two designs, a fixed design $D_1=\{\mathbf{d}_{11},\ldots,\mathbf{d}_{1n_1}\}$ and a proposed design $D_2=\{\mathbf{d}_{21},\ldots,\mathbf{d}_{2n_2}\}$:

$$\ell_{\boldsymbol{\tau}}(D_1,D_2)=\min_{\substack{i=1,\ldots,n_1\\ j=1,\ldots,n_2}}\ell_{\boldsymbol{\tau}}(\mathbf{d}_{1i},\mathbf{d}_{2j})\wedge\min_{\substack{i=1,\ldots,n_2-1\\ j=i+1,\ldots,n_2}}\ell_{\boldsymbol{\tau}}(\mathbf{d}_{2i},\mathbf{d}_{2j}).$$

The maximin Euclidean distance (MmED) and maximin sensitivity-weighted distance (MmSWD) criteria choose input settings $\mathbf{x}_1^*,\ldots,\mathbf{x}_{n_p}^*$ at each stage to maximize $\ell_{\mathbf{1}}(D_y,D_\xi)$ ($\boldsymbol{\tau}=\mathbf{1}$) and $\ell_{\boldsymbol{\beta}_\delta}(D_y,D_\xi)$ ($\boldsymbol{\tau}=\boldsymbol{\beta}_\delta$), respectively, where $\boldsymbol{\beta}_\delta$ is the vector of roughness parameters ($\beta_{\delta,i}=-4\log(\rho_{\delta,i})$) in the GP model of discrepancy. The roughness parameters are related to the local (and global) sensitivities of the input parameters, hence the terminology "sensitivity-weighted distance." These criteria maximally separate proposed input settings for new experiments from existing experiments according to two choices of distance measure. Johnson et al. [12] introduced maximin Euclidean distance as a criterion to select designs for GP modeling of deterministic computer model output. Sensitivity-weighted distance is motivated by the fact that $\ell_{\boldsymbol{\beta}_\delta}(D_y,D_\xi)$ is a bijection of the correlation between the discrepancies corresponding to the nearest input settings in D_y and D_ξ or in D_ξ. Therefore, the MmSWD criterion minimizes the maximum discrepancy correlation, thus proposing input settings in locations of design space at which neighborhood information for making GP predictive inference is weak.

A functional extension of MmSWD maximizes $\sum_{i=1}^{p_\delta}\ell_{\boldsymbol{\beta}_{v_i}}(D_y,D_\xi)/\lambda_{v_i}$, where $\boldsymbol{\beta}_{v_i}$ and λ_{v_i} are the roughness and precision parameters corresponding to the discrepancy basis coefficient v_i, respectively. This criterion gives greater weight to minimum distances involving discrepancy coefficients with larger variance and thus larger overall contributions to discrepancy.

4. Example

We conducted a simulation study to assess the effects of several factors on global prediction for the three scalar outputs of interest in calibration problems, described in Section 3: discrepancy, best code calculation (calibrated code), and reality. The design space ($\mathbf{x}$) is six-dimensional, and the code has four calibration parameters ($\boldsymbol{\theta}$). The design used for code emulation is fixed (a 100-run space-filling Latin hypercube (LH) design in the ten code inputs), and only the impact of experimental information on global prediction is considered.

The "code" used in this study is the ten dimensional version of a function used by Sobol' and Levitan [30],

$$\eta(\mathbf{z})=\exp\left(\sum_{i=1}^{10}b_iz_i\right)-I_{10}$$

where

$$I_{10}=\prod_{i=1}^{10}\frac{e^{b_i}-1}{b_i}\ \text{ and }\ z_i\in[0,1],\quad i=1,\ldots,10.$$

The first six inputs, $z_1,\ldots,z_6$, are taken to be the design settings while the last four inputs, $z_7,\ldots,z_{10}$, are the calibration parameters. The coefficients b_i chosen for this study result in the input main and total effect sensitivity indices (Chan et al. [31]) of Table 2.

The total standard deviation of code output with respect to uniform input variation is approximately 42, with 80% of this variance explained collectively by the main effects of the ten inputs. This specification represents a code that varies smoothly with each input and has most of its output variation explained by relatively simple effects, a situation typical of many applications in the physical sciences.

Physical reality was constructed for this study by taking the true setting for each of the four calibration parameters to be 0.5, and adding a discrepancy to the resulting best code calculation. The discrepancy is constructed as follows: First, a mean zero and variance one Gaussian process is constructed that uses the canonical configuration of Loeppky et al. [32] to define the correlation lengths $\rho_{\delta,i}$ in the six design dimensions

$$\beta_{\delta,i}=\tau\left[\left(1-\frac{i-1}{6}\right)^b-\left(1-\frac{i}{6}\right)^b\right],\quad \rho_{\delta,i}=\exp(-\beta_{\delta,i}/4),\quad i=1,\ldots,6.$$

The parameters $\tau>0$ and $b\geq 1$ are specified. Next, a sample is generated from this GP on a 625-run space-filling LH design in six dimensions, centered to have mean zero, and scaled to have a specified variance (5% or 15% of total variance in code output, see below). Finally, discrepancy at any design setting is computed by evaluating the kriging predictor with these correlation lengths and the centered, scaled sample.

The parameters τ and b are overall measures of output complexity and effect sparsity, respectively. Larger values of τ correspond to shorter correlation lengths in each input dimension, resulting in a discrepancy with increased activity on short length scales. Smaller values of b result in a more even distribution of input effects; if $b=1$ each input has an equivalent effect on output variation. Therefore, as discussed in Loeppky et al. [32], discrepancy constructed as described in the previous paragraph will be more difficult to predict for larger values of τ and smaller values of b.

Experimental data are generated by adding zero-mean Gaussian noise with specified variance to this constructed physical reality.

This study compares twelve design strategies for conducting a total budget of 64 experiments. A symmetric LH design (Ye et al. [33]) that specifies all 64 runs in one stage is compared with nine sequential design criteria and two space-filling sequences: Sobol' LP_τ (Sobol [34]) and scrambled Sobol' LP_τ (Matousek [35]). The sequential criteria are EIPS, GEIGF (with $g=2$ and 4), IMSE-δ, MMSE-δ, ME-δ, MmED, MmSWD, and bin-based LH design (Loeppky et al. [14]). These criteria are all implemented in batch sequential fashion. With the exception of bin-based LH designs,

Table 2
Main and total effect sensitivity indices for the ten code inputs.

Input	b_i	Main effect (%)	Total effect (%)	Input	b_i	Main effect (%)	Total effect (%)
z_1	1.1895	17.8	26.3	z_6	0.6895	6.1	9.6
z_2	1.0895	15.0	22.5	z_7	0.5895	4.5	7.1
z_3	0.9895	12.4	18.9	z_8	0.4895	3.1	5.0
z_4	0.8895	10.1	15.6	z_9	0.3895	2.0	3.2
z_5	0.7895	8.0	12.5	z_{10}	0.2895	1.1	1.8

the batch sequential design algorithm applied to each design criterion proceeds as follows:

1. Start with an initial 32-run space-filling LH design in the six design variables, and the experimental data collected from this design.
2. Estimate any parameters required to calculate the design criterion (e.g. GP parameters of the code emulator and discrepancy function, calibration parameter setting) using information from the current design. We use posterior mean estimates of these parameters based on 1000 MCMC realizations.
3. Augment the current design with a proposed batch of eight runs, and perform a continuous optimization of this batch with respect to the design criterion (using parameter estimates from the previous step if necessary) employing a modified Fedorov exchange (Fedorov [22]).
4. Collect new experimental data on the optimal batch from the previous step, augment the current design with this optimal batch, and continue with step (2) until termination.

In this context, termination occurs once 64 runs are obtained, i.e. after four batches are added to the initial design. Termination could also be defined by sufficient stabilization of the optimal design criterion value.

Step (3) in the above algorithm is modified to construct bin-based LH designs, first introduced in Loeppky et al. [14] and described there in additional detail with examples of particular constructions. Bin-based LH design first identifies an optimal sequence of bin designs, where the bin structure associates values of each input with levels. For example, a common approach divides each input into two bins (e.g. level 0 corresponds to input values [0, 0.5] and level 1 to input values (0.5, 1]). The initial design projects into the first bin structure, as does an orthogonal array based LH design (McKay et al. [36], Tang [37], Owen [38]). The bin augmentations are added by combinatorial design considerations or sequentially optimizing a criterion such as maximin distance over the aggregate bin design. The bin augmentations associate a set of values of the input that is sampled so that the aggregate designs are approximately Latin hypercube, using a procedure such as the following:

1. For each input dimension, the input range associated with each bin level is divided into the number of strata equal to the number of runs that currently project into that bin plus the number of new runs assigned to that bin in the proposed augmentation.
2. For each input dimension assign new input values associated with each bin level in the augmentation as a sample from strata not containing any runs in the current design and optimized with respect to the maximin distance criterion applied to the aggregate design.

Bin-based LH design extends the concept of OA-based LH design with a method to sequentially augment such designs maintaining the LH concept of having dense coverage of marginals. Further extension and development of these methods appear in Loeppky et al. [39].

We now summarize the factors varied in this simulation study:

- Design: The 12 strategies as described above.
- Discrepancy variance: Used in discrepancy construction, this has two levels—5% and 15% of total variance in code output.
- Complexity: The τ parameter in discrepancy construction, with two levels—1 and 10.
- Sparsity: The b parameter in discrepancy construction, with two levels—1 and 9.
- Experimental error: Used in experimental data generation, this has three levels—1%, 5% and 10% of total variance in code output.

Two cases are evaluated: (1) no discrepancy, using a full factorial design in the Design and Experimental Error factors; and (2) discrepancy present, using a full factorial design in all five factors. Each level combination in these designs is replicated five times.

For each final 64-run design, the posterior means of the discrepancy, calibrated code and reality predictive distributions based on 500 MCMC realizations are compared with the corresponding true values using both root mean square and maximum absolute prediction errors (RMSPE and MAPE, respectively) calculated on a 625-run space-filling LH design. Analyses of variance on the RMSPE and MAPE values were run for both cases, allowing for main effects and two-factor interactions, and the AIC criterion was used to select the best subset of effects for each analysis. Tukey's honest significant difference method (Tukey [40]) was used to identify the significant pairwise differences among main effect levels for each factor presented in the following two subsections.

4.1. Case 1: no discrepancy

RMSPE and MAPE values for predicting discrepancy, which in this case is zero throughout the design parameter space, were inferior for the sequential design strategy based on ME-δ relative to every other strategy. Similarly, RMSPE values for calibrated code predictions were inferior for ME-δ relative to every other strategy. MAPE values were superior for MmED relative to every other strategy except MmSWD, and for MmSWD relative to all but three of the other strategies. MAPE values for ME-δ were inferior except for two of the one-stage design strategies. Finally, RMSPE

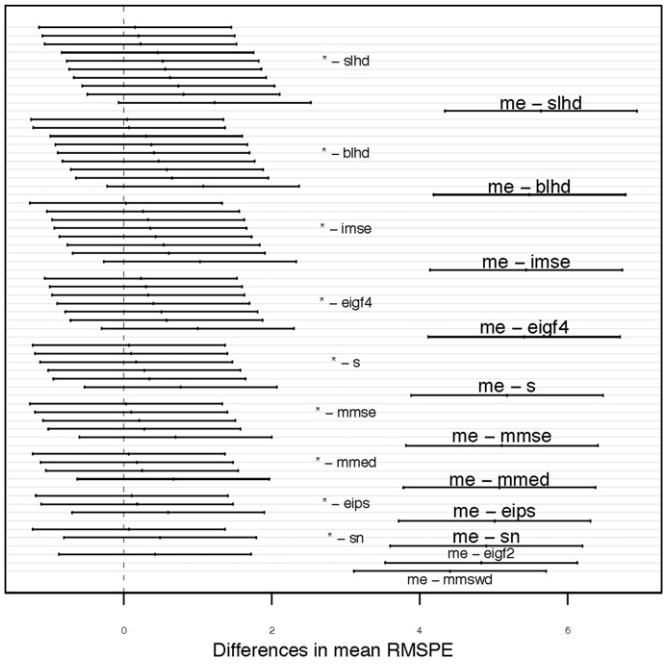

Fig. 1. Tukey's 95% family-wise confidence intervals for difference in mean RMSPE between each pair of design strategies. Confidence intervals are grouped by strategy as indicated: symmetric LH design (slhd), bin-based LH design (blhd), IMSE-δ (imse), GEIGF with $g=4$ (eigf4), Sobol' LP_τ (s), MMSE-δ (mmse), MmED (mmed), EIPS (eips), scrambled Sobol' LP_τ (sn), GEIGF with $g=2$ (eigf2), MmSWD (mmswd), and ME-δ (me).

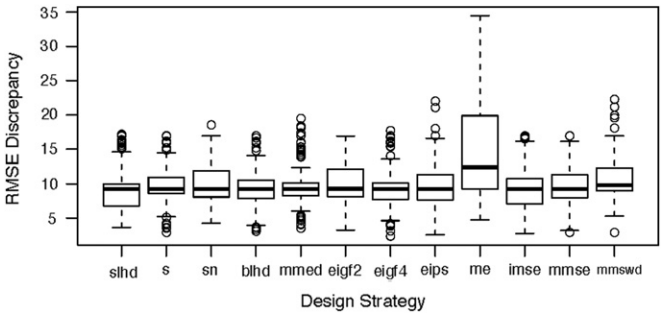

Fig. 2. Boxplots of RMSPE values by design strategy: symmetric LH design (slhd), Sobol' LP_τ (s), scrambled Sobol' LP_τ (sn), bin-based LH design (blhd), MmED (mmed), GEIGF with $g=2$ (eigf2), GEIGF with $g=4$ (eigf4), EIPS (eips), ME-δ (me), IMSE-δ (imse), MMSE-δ (mmse), and MmSWD (mmswd).

Table 3
Tukey's 95% family-wise confidence intervals for difference in mean RMSPE between the two levels of discrepancy variance (δ *var*) and the three levels of experimental error variance (ε var).

Pairwise difference	Lower bound	Upper bound
$RMSPE\ (\delta var = 15\%) - RMSPE\ (\delta var = 5\%)$	1.97	2.60
$RMSPE\ (\varepsilon var = 5\%) - RMSPE\ (\varepsilon var = 1\%)$	1.11	2.04
$RMSPE\ (\varepsilon var = 10\%) - RMSPE\ (\varepsilon var = 1\%)$	2.45	3.39
$RMSPE\ (\varepsilon var = 10\%) - RMSPE\ (\varepsilon var = 5\%)$	0.88	1.81

and MAPE values for predicting reality were inferior for ME-δ relative to every other strategy.

4.2. Case 2: discrepancy present

RMSPE and MAPE values for predicting discrepancy were inferior for ME-δ relative to every other strategy. Fig. 1 provides Tukey's 95% confidence intervals for the difference in mean RMSPE between each pair of design strategies. The poor performance of ME-δ relative to every other design strategy is evident from these results.

Fig. 2 provides boxplots of RMSPE values by design strategy. The poor performance of ME-δ is confirmed, in terms of both greater mean RMSPE and a significantly higher likelihood of observing substantially larger RMSPE values than typically seen with the other design strategies.

Both RMSPE and MAPE values from discrepancy prediction were larger for more complex discrepancy functions (large τ) and for discrepancy functions with less effect sparsity (small b), consistent with the results of Loeppky et al. [32]. Tukey's 95% confidence intervals for difference in mean RMSPE are given by

$RMSPE\ (\tau = 10) - RMSPE\ (\tau = 1) \in (0.58, 1.22)$ and
$RMSPE\ (b = 1) - RMSPE\ (b = 9) \in (0.54, 1.18)$.

As expected, we also found that prediction of discrepancy is more difficult when either discrepancy or experimental error variances are larger. Table 3 presents Tukey's 95% confidence intervals for the difference in mean RMSPE for these factors.

RMSPE values for calibrated code predictions were inferior for ME-δ relative to every other strategy, while MmSWD was inferior to all but three of the other strategies. MAPE values for MmSWD and MmED were superior relative to every other strategy, while Sobol LP_τ, bin-based LH design, and IMSE-δ were inferior to one-stage symmetric LH design, GEIGF with $g=4$ and ME-δ.

Interestingly, RMSPE and MAPE values from calibrated code predictions were *smaller* for more complex discrepancy functions. This suggests that in the presence of discrepancy, some complexity *may* provide useful information about the best calibration parameter setting $\boldsymbol{\theta}$. Of course, excessive complexity is undesirable as it results in inadequate prediction of the discrepancy function.

RMSPE values for predicting reality were superior for the one-stage symmetric LH design relative to all other strategies except for bin-based LH design. The ME-δ strategy was inferior to every other strategy, while MmSWD was inferior to every other strategy except MmED and ME-δ. MmED was inferior to approximately half of the other strategies. MAPE values for MmED were superior to all but three of the other strategies, while ME-δ was inferior to every other strategy.

As with discrepancy prediction, both RMSPE and MAPE values from reality predictions were larger for more complex discrepancy functions.

4.3. Conclusions

Table 4 summarizes the main results of this simulation study:

- The one-stage symmetric LH design is never outperformed in controlling average prediction errors, and only rarely surpassed in controlling maximum prediction errors.
- The maximum entropy criterion ME-δ is outperformed by the other strategies for predicting discrepancy, calibrated code and reality.
- The space-filling sequences and bin-based LH design control average error better than maximum error, while this relationship is reversed for the two distance-based criteria.
- The remaining criteria are rarely outperformed in controlling average or maximum prediction errors, although IMSE-δ is slightly less effective at mitigating maximum errors when discrepancy is present.

Fig. 3 shows projections of the final 64-run designs into (x_1, x_2) space from several of the strategies for the no discrepancy case. Comparing Table 4 and Fig. 3 suggests several observations regarding the relative prediction performance of these design strategies:

- Design strategies that spread points throughout the input space, particularly in active input dimensions, result in smaller average prediction errors. This reflects the fact that GP-based predictors generally perform better when paired with space-filling designs (e.g. Jones and Johnson [41]) due to the local nature of the fitting procedure.
- Design strategies that concentrate runs near boundaries tend to control maximum prediction error better, because the largest prediction errors with GP models are often located on or near boundaries due to a relative lack of information for fitting.

Table 4
Count of pairwise comparisons in favor of each design strategy, aggregated over RMSPE and MAPE values for prediction of discrepancy, calibrated code and reality.

Criteria	No discrepancy		Discrepancy present	
	RMSPE	MAPE	RMSPE	MAPE
Symmetric LH design	3	2	13	5
Sobol LP_τ	3	0	4	−6
Scrambled Sobol LP_τ	3	1	5	−1
Bin LH design	3	0	6	−4
MmED	3	12	−2	19
MmSWD	3	10	−14	13
EIPS	3	1	4	−1
GEIGF ($g=2$)	3	2	4	−1
GEIGF ($g=4$)	3	1	4	3
IMSE-δ	3	1	5	−5
MMSE-δ	3	1	4	−1
ME-δ	−33	−31	−33	−21

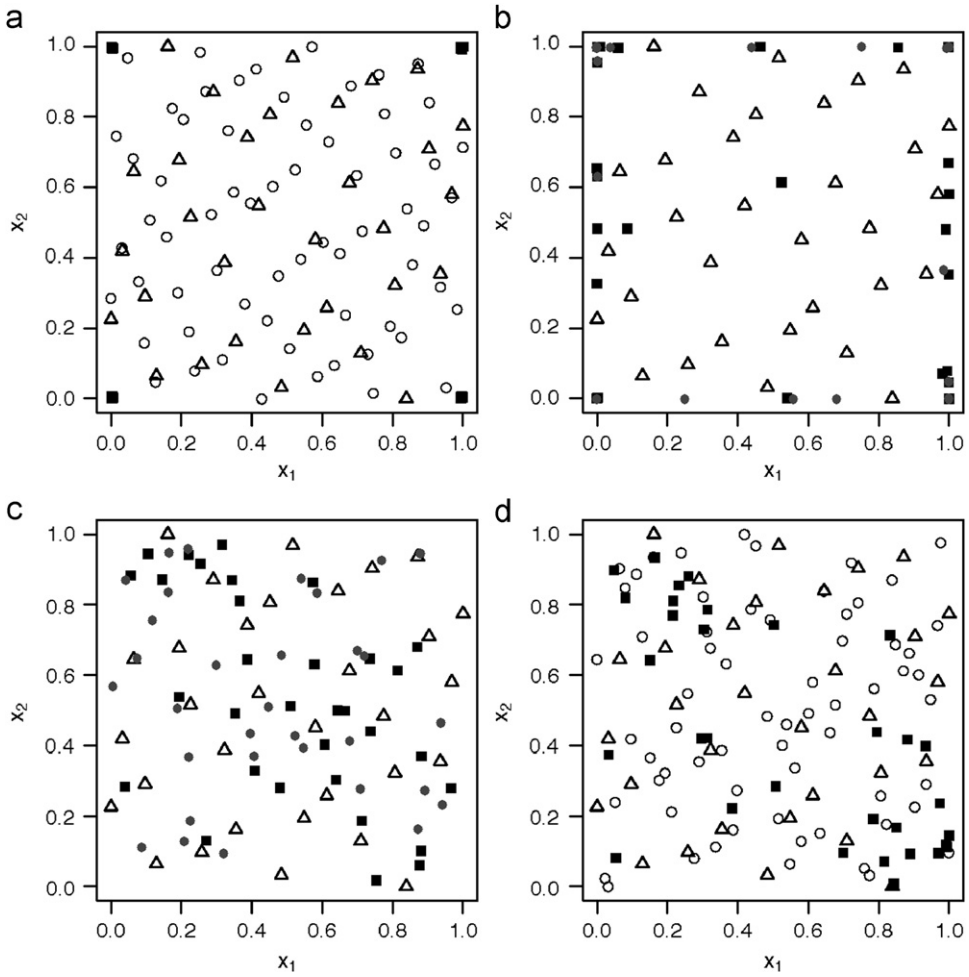

Fig. 3. Final 64-run designs projected into (x_1, x_2) space. Open triangles denote the 32-run initial design used for each sequential design strategy (except bin-based LH design). (a) Open circles denote the 64-run symmetric LH design and black squares denote the 32 runs added by ME-δ. (b) Black (gray) squares (circles) denote the 32 runs added by MmSWD (MmED). (c) Black (gray) squares (circles) denote the 32 runs added by EIPS (GEIGF with $g=4$). (d) Open circles denote the 64-run bin-based LH design and black squares denote the 32 runs added by IMSE-δ.

- The maximum entropy ME-δ criterion locates points near boundaries of the design space, resulting in poor average prediction error, and these points tend to clump, thus also failing to mitigate maximum prediction error.

Clumping is not necessarily discouraged in designs for predictive maturity, as some clumping could be beneficial for estimation of replicate variability in the experimental data. However, it is clear that space-filling design strategies that expend some effort exploring relevant boundaries of design space would tend to perform well with respect to both RMSPE and MAPE.

The four calibration parameters of this simulation study collectively explain roughly 13% of main effect variation (see Table 2). The simulation study was repeated for three additional cases in which the four calibration parameters explain 20%, 40% and 60% of main effect variation. The conclusions presented in this subsection carry over to these cases, although designs based on the ME-δ criterion progressively improve relative to the other strategies with respect to RMSPE and MAPE as the main effect contribution of the design parameters decreases.

5. Discussion

The results of Section 4 suggest that if given a total budget for conducting experiments to be used for calibrating a code, a single-stage space-filling design strategy (such as symmetric LH design) is likely to result in prediction performance comparable to that obtained from sequential strategies. However, if a current experimental database were to be augmented, several design criteria studied in this article would likely be suitable in a (batch) sequential algorithm for efficiently achieving predictive maturity. In particular, bin-based LH design, EIPS, GEIGF, IMSE-δ and MMSE-δ should be considered.

The comparable performance of batch designs relative to single-stage designs does suggest that one might be better off using all available resources in one stage. However, how one should proceed when the initial run sizes are inadequate to achieve the desired level of maturity dictates that sequential strategies must be employed. In such cases it is useful to know that batch sequential updates are nearly as efficient as running the full design, had that been possible. We envision that sequential updates are only employed in situations where the initial run sizes were inadequate.

Most of the sequential design criteria proposed in this article are motivated by the existing literature on computer model evaluation or Bayesian experiment design. The main contributions of this article are extensions of these criteria to allow design augmentation in batches, and application of these criteria to the problem of achieving predictive maturity in discrepancy inferences derived from probabilistic calibration of computer models to experimental data.

This work can be expanded in many directions. For example, the code was fixed in the simulation study of Section 4. However, it is likely that certain code properties could impact the performance of sequential design strategies, such as the proportion of total output variance explained by simple versus complex effects, the partitioning of sensitivity to the design and calibration parameters (including interactions between these), and the behavior of the code near boundaries of the design space relative to the interior region. Variations in the numbers of design and calibration parameters, and the size of the initial design relative to the experimental budget could be explored, as well as the batch sizes of sequential augmentations.

If no good estimate of experimental replicate variability is available, it would be possible to modify the sequential optimization algorithm to require a specified fraction of runs in each new batch to be replicates of runs already made (see Lu et al. [25]). Sensitivity of prediction performance to this fraction could then be explored. Practical applications will often restrict the design space to a discrete, finite set of input level combinations for which it is feasible to conduct new experiments. This would require modification of the continuous optimization used in this article; for example, a Fedorov exchange on the set of allowable design specifications could be employed. Bin-based LH design should be particularly effective in this setting.

Ongoing simulation studies suggest that any benefits of employing a batch sequential design strategy relative to a fixed design strategy dissipate as the dimension of the design space increases beyond even five inputs, consistent with the conclusions reached in Section 4 and Loeppky et al. [14]. This suggests a hybrid sequential design strategy in which batches are constructed, for example, to form a space-filling sample in conjunction with the current design and are selected based on evaluating a design criterion such as one of those proposed here. Evaluation of such hybrid strategies is the subject of ongoing work.

In many applications, including annual assessments of the nuclear weapon systems maintained by NNSA, full system performance calculations require the integration of many individual components and sub-systems. Many sources of experimental data are often available to inform on different levels of the physical hierarchy comprising the full system, perhaps including data on the full system itself. Component level data is generally easier to obtain and less costly compared with sub-system or full system data, while the latter are generally more informative for constraining physical models in the regimes of interest. For a given experimental budget, the question arises as to how these resources should be allocated to various campaigns so that the experimental information obtained for improving the required performance predictions is optimized. The sequential design strategies of this article could be used to recommend new experiments for achieving predictive maturity at each relevant level of the hierarchy, as a component of a comprehensive resource allocation framework.

Model calibration was presented in a fully Bayesian context; however, calculation of the design criterion functions assumed plug-in point estimates of all parameters derived from calibration. This restriction is not essential for the implementation of these criteria, although it certainly reduces computational effort. Two alternatives are readily available:

- Compute the posterior mean of the design criteria, or
- Compute the design criteria using the relevant posterior predictive distributions.

Although the second alternative is the most desirable Bayesian solution, for many criteria it presents serious practical challenges. For example, computation of the EIPS criterion is nontrivial, given that it involves the expected value of a Kullback–Leibler distance between two predictive distributions with respect to the marginal distribution of an output vector.

A consequence of the statistical models assumed for calibration and prediction (Kennedy and O'Hagan [2], Higdon et al. [20]) is that the predictive distributions of relevance to this discussion are mixtures of Gaussian distributions. A simple approach is to approximate these mixture distributions by a single Gaussian distribution, with mean and covariance matched to the corresponding quantities calculated from the mixtures constituting the predictive distributions. A more complicated approach is to construct design criteria that allow more direct calculation with the actual predictive distributions.

We present an example of the latter approach by considering an alternative distance metric that provides modest simplification of the required calculations. Sfikas et al. [42] proposed the following distance metric between two probability density functions p_0 and p_1:

$$D(p_0,p_1) = -\log\left[\frac{2\int p_0(\mathbf{z})p_1(\mathbf{z})\,d\mathbf{z}}{\int\left[p_0^2(\mathbf{z})+p_1^2(\mathbf{z})\right]d\mathbf{z}}\right].$$

This metric is symmetric and positive, and is equal to zero when $p_0=p_1$. When the p_i are mixtures of Gaussian distributions ϕ,

$$p_i(\mathbf{z}) = \int \phi\left(\mathbf{z};\boldsymbol{\mu}_i(\boldsymbol{\omega}),\boldsymbol{\Sigma}_i(\boldsymbol{\omega})\right)\pi_i(\boldsymbol{\omega})\,d\boldsymbol{\omega},$$

we have

$$D(p_0,p_1) = -\log\left[\frac{2\iint \frac{\pi_0(\boldsymbol{\omega})\pi_1(\boldsymbol{\omega}')}{\sqrt{\exp\left[k_{01}(\boldsymbol{\omega},\boldsymbol{\omega}')\right]\ |\Sigma_0(\boldsymbol{\omega})+\Sigma_1(\boldsymbol{\omega}')|}}\ d\boldsymbol{\omega}\ d\boldsymbol{\omega}'}{\iint \frac{\pi_0(\boldsymbol{\omega})\pi_0(\boldsymbol{\omega}')}{\sqrt{\exp\left[k_{00}(\boldsymbol{\omega},\boldsymbol{\omega}')\right]\ |\Sigma_0(\boldsymbol{\omega})+\Sigma_0(\boldsymbol{\omega}')|}}\ d\boldsymbol{\omega}\ d\boldsymbol{\omega}' + \iint \frac{\pi_1(\boldsymbol{\omega})\pi_1(\boldsymbol{\omega}')}{\sqrt{\exp\left[k_{11}(\boldsymbol{\omega},\boldsymbol{\omega}')\right]\ |\Sigma_1(\boldsymbol{\omega})+\Sigma_1(\boldsymbol{\omega}')|}}\ d\boldsymbol{\omega}\ d\boldsymbol{\omega}'}\right]$$

where

$$k_{ij}(\boldsymbol{\omega},\boldsymbol{\omega}') = (\boldsymbol{\mu}_i(\boldsymbol{\omega})-\boldsymbol{\mu}_j(\boldsymbol{\omega}'))^T(\boldsymbol{\Sigma}_i(\boldsymbol{\omega})+\boldsymbol{\Sigma}_j(\boldsymbol{\omega}'))^{-1}(\boldsymbol{\mu}_i(\boldsymbol{\omega})-\boldsymbol{\mu}_j(\boldsymbol{\omega}')).$$

The distribution $\pi_1=\pi_0$ is the posterior distribution of the parameters from the current calibration. It is generally known only up to a normalizing constant, and so the distance criterion $D(p_0,p_1)$ must be estimated based on independent samples $\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_M$ and $\boldsymbol{\omega}'_1,\ldots,\boldsymbol{\omega}'_M$ from π_0:

$$D(p_0,p_1) = -\log\left[\frac{2\sum_{i,j}\frac{1}{\sqrt{\exp[k_{01}(\boldsymbol{\omega}_i,\boldsymbol{\omega}'_j)]|\Sigma_0(\boldsymbol{\omega}_i)+\Sigma_1(\boldsymbol{\omega}'_j)|}}}{\sum_{i,j}\frac{1}{\sqrt{\exp[k_{00}(\boldsymbol{\omega}_i,\boldsymbol{\omega}'_j)]|\Sigma_0(\boldsymbol{\omega}_i)+\Sigma_0(\boldsymbol{\omega}'_j)|}}+\sum_{i,j}\frac{1}{\sqrt{\exp[k_{11}(\boldsymbol{\omega}_i,\boldsymbol{\omega}'_j)]|\Sigma_1(\boldsymbol{\omega}_i)+\Sigma_1(\boldsymbol{\omega}'_j)|}}}\right].$$

A modified EIPS criterion is obtained by setting $p_0=f_n(\xi(\mathbf{x}))$ and $p_1=f_{n+n_p}(\xi(\mathbf{x}))$ in the notation of Section 2.1, and selecting the batch to maximize the minimum distance $D(p_0,p_1)$. Note that $\boldsymbol{\mu}_2(\boldsymbol{\omega})$ is a function of the proposed batch $y(\mathbf{x}_1^*),\ldots,y(\mathbf{x}_{n_p}^*)$, requiring an integration to compute $D(p_0,p_1)$ that cannot be reduced to closed form.

Acknowledgements

The research of Williams and Moore was supported by Cetin Unal of Los Alamos National Laboratory, through the Nuclear Energy Advanced Modeling and Simulation Campaign of the U.S. Department of Energy's Advanced Fuel Cycle Initiative. The research of Loeppky and Macklem was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. The authors thank two anonymous reviewers for providing comments that improved the presentation of this article.

References

[1] National Research Council. Evaluation of quantification of margins and uncertainties methodology for assessing and certifying the reliability of the nuclear stockpile. Washington, DC: National Academy Press; 2008.
[2] Kennedy M, O'Hagan A. Bayesian calibration of computer models (with discussion). J R Stat Soc Series B Stat Methodol 2001;68:425–64.
[3] Schonlau M, Welch WJ, Jones DR. Global versus local search in constrained optimization of computer models. In: Flournoy N, Rosenberger WF, Wong WK, editors. New developments and applications in experimental design, vol. 34. Hayward: Institute of Mathematical Statistics; 1998. p. 11–25.
[4] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim 1998;13:455–92.
[5] Williams BJ, Santner TJ, Notz WI. Sequential design of computer experiments to minimize integrated response functions. Stat Sin 2000;10:1133–52.
[6] Lehman JS, Santner TJ, Notz WI. Designing computer experiments to determine robust control variables. Stat Sin 2004;14:571–90.
[7] Keane AJ. Statistical improvement criteria for use in multiobjective design optimization. AIAAJ 2006;44:879–91.
[8] Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset MW. A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 1999;17:1–13.
[9] Audet C, Dennis JE, Moore DW, Booker A, Frank PD. A surrogate-model-based method for constrained optimization. AIAA-2000-4891:1-10.
[10] Regis RG, Shoemaker CA. Constrained global optimization of expensive black box functions using radial basis functions. J Glob Optim 2005;31:153–71.
[11] Ranjan P, Bingham D, Michailidis G. Sequential experiment design for contour estimation from complex computer codes. Technometrics 2008;50:527–41.
[12] Johnson ME, Moore LM, Ylvisaker D. Minimax and maximin distance designs. J Stat Plan Inference 1990;26:131–48.
[13] Morris MD, Mitchell TJ. Exploratory designs for computational experiments. J Stat Plan Inference 1995;43:381–402.
[14] Loeppky JL, Moore LM, Williams BJ. Batch sequential designs for computer experiments. J Stat Plan Inference 2010;140:1452–64.
[15] Chaloner K, Verdinelli I. Bayesian experimental design: a review. Stat Sci 1995;10:273–304.
[16] Bingham DR, Chipman HA. Incorporating prior information in optimal design for model selection. Technometrics 2007;49:155–63.
[17] Lam CQ, Notz WI. Sequential adaptive designs in computer experiments for response surface model fit. Stat Appl 2008;6:207–33.
[18] Higdon D, Kennedy M, Cavendish J, Cafeo J, Ryne RD. Combining field observations and simulations for calibration and prediction. SIAM J Sci Comput 2004;26:448–66.
[19] Williams B, Higdon D, Gattiker J, Moore L, McKay M, Keller-McNulty S. Combining experimental data and computer simulations, with an application to flyer plate experiments. Bayesian Anal 2006;1:765–92.
[20] Higdon D, Gattiker J, Williams B, Rightley M. Computer model calibration using high-dimensional output. J Am Stat Assoc 2008;103:570–83.
[21] Lam CQ. Sequential adaptive designs in computer experiments for response surface model fit. The Ohio State University, Unpublished PhD thesis; 2008.
[22] Fedorov VV. Theory of optimal design. New York: Academic; 1972.
[23] Genz AC, Malik AA. Remarks on algorithm 006: an adaptive algorithm for numerical integration over an N-dimensional rectangular region. J Comput Appl Math 1980;6:295–302.
[24] O'Hagan A. Bayes–Hermite quadrature. J Stat Plan Inference 1991;29:245–60.
[25] Lu W, Ranjan P, Bingham D, Reese CS, Williams BJ. Follow-up experiment designs for computer models and physical processes. Acadia University technical report, unpublished results.
[26] Sacks J, Schiller S. Spatial designs. In: Gupta SS, Berger JO, editors. Statistical decision theory and related topics IV. New York: Springer-Verlag; 1988. p. 385–99.
[27] Sacks J, Schiller SB, Welch WJ. Design for computer experiments. Technometrics 1989;31:41–7.
[28] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Stat Sci 1989;4:409–23.
[29] Shewry MC, Wynn HP. Maximum entropy sampling. J Appl Stat 1987;14:165–70.
[30] Sobol' IM, Levitan YL. On the use of variance reducing multipliers in Monte Carlo computations of a global sensitivity index. Comput Phys Commun 1999;117:52–61.
[31] Chan K, Tarantola S, Saltelli A, Sobol' IM. Variance-based methods. In: Saltelli A, Chan K, Scott EM, editors. Sensitivity analysis. Chichester: J. Wiley & Sons; 2000. p. 167–97.
[32] Loeppky JL, Sacks J, Welch WJ. Choosing the sample size of a computer experiment: a practical guide. Technometrics 2009;51:366–76.
[33] Ye K, Li W, Sudjianto A. Algorithmic construction of optimal symmetric Latin hypercube designs. J Stat Plan Inference 2000;90:145–59.
[34] Sobol' IM. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computat Math Math Phys 1967;7:86–112.
[35] Matousek J. On the L2-discrepancy for anchored boxes. J Complex 1998;14:527–56.
[36] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 1979;21:239–45.
[37] Tang B. Orthogonal array-based Latin hypercubes. J Am Stat Assoc 1993;88:1392–7.
[38] Owen AB. Controlling correlations in Latin hypercube samples. J Am Stat Assoc 1994;89:1517–22.
[39] Loeppky JL, Moore LM, Williams BJ. Projection array based designs for computer experiments. Los Alamos National Laboratory technical report, unpublished results.
[40] Tukey JW. The problem of multiple comparisons. In: Braun HI, editor. The collected works of John W. Tukey, vol. VIII, multiple comparisons: 1948–1983. New York: Chapman and Hall; 1994. p. 1–300.
[41] Jones B, Johnson RT. Design and analysis for the Gaussian process model (with discussion). Qual Reliab Eng Int 2009;25:515–50.
[42] Sfikas G, Constantinopoulos C, Likas A, Galatsanos NP. An analytic distance metric for Gaussian mixture models with application in image retrieval. In: Duch W, Kacprzyk J, Oja E, Zadrozny S, editors. Artificial neural networks: formal models and their applications—ICANN 2005. Berlin: Springer; 2005. p. 835–40.