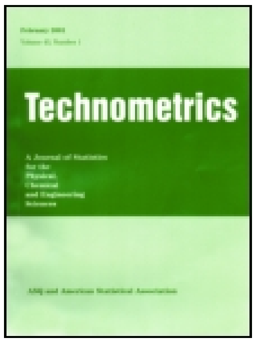

Technometrics

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: http://amstat.tandfonline.com/loi/utch20

Surrogate Modeling of Computer Experiments With Different Mesh Densities

Rui Tuo, C. F. Jeff Wu & Dan Yu

Surrogate Modeling of Computer Experiments With Different Mesh Densities

Rui Tuo

Academy of Mathematics and Systems Science
Chinese Academy of Sciences
100190 Beijing, P.R. China
(tuorui@amss.ac.cn)

C. F. Jeff Wu

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta 30332, GA
(jeffwu@isye.gatech.edu)

Dan Yu

Academy of Mathematics and Systems Science
Chinese Academy of Sciences
100190 Beijing, P.R. China
(dyu@amss.ac.cn)

This article considers deterministic computer experiments with real-valued tuning parameters which determine the accuracy of the numerical algorithm. A prominent example is finite-element analysis with its mesh density as the tuning parameter. The aim of this work is to integrate computer outputs with different tuning parameters. Novel nonstationary Gaussian process models are proposed to establish a framework consistent with the results in numerical analysis. Numerical studies show the advantages of the proposed method over existing methods. The methodology is illustrated with a problem in casting simulation. Supplementary material for this article is available online.

KEY WORDS: Brownian motion; Finite-element analysis; Kriging; Multifidelity data; Nonstationary Gaussian process models; Tuning parameters.

1. INTRODUCTION

Numerical computations like finite-element analysis (FEA) are commonly used in simulating real-world phenomena like soil erosion, climate change, etc. These computations often have a tuning parameter like the mesh density in FEA, which controls the numerical accuracy as well as the computational cost/time. FEA with a coarser mesh is much cheaper but less accurate, while FEA with a finer mesh is more accurate but more costly. Therefore, it can be beneficial to run FEA with two choices of mesh density to exploit the advantages of accuracy and cost. This is particularly useful if many combinations of the input variables should be considered, which is common in mechanical or material design. More combinations can be explored using cheaper but less accurate simulations while a smaller number of expensive but accurate simulations can be used to improve the overall prediction accuracy. The main goal of this article is to develop a framework for studying the stated problem and to propose a class of nonstationary Gaussian process models to link the outputs of simulation runs with different mesh densities to better use the data for modeling and prediction.

Specifically, we consider computer experiments in which a set of partial differential equations (PDEs) is solved numerically to simulate the result of a corresponding physical experiment. There are two types of inputs for such experiments. One type is the *input variables*. Computer runs with different input variables solve different PDEs or the same PDEs but with different initial or boundary conditions. Input variables can be control variables, environmental variables (Santner, Williams, and Notz 2003, pp. 15–16), or calibration variables (Kennedy and O'Hagan 2001). The other type is the *tuning variables*, which determine the performance of the numerical computations. The main focus of this article is on the tuning variables. If the numerical solution is sufficiently accurate for each simulation run, it would not be necessary to incorporate the tuning variables in the statistical model. Then, stationary Gaussian process models would be suitable for modeling the computer outputs (Santner, Williams, and Notz 2003). This is often not the case for two reasons. First, implementing high-accuracy computer runs for the whole experiment can be costly. Second, FEA with finer mesh gives more accurate results than those with coarser mesh. In such scenarios, nonstationary Gaussian process models that incorporate the *varying* accuracies with the mesh density will be more appropriate.

To motivate and justify the proposed model for the tuning parameters, we import some basic concepts and results from numerical analysis. We first describe the mathematical theory which gives an error bound for the finite-element methods. This can be used to guide the construction of the corresponding statistical model. Theoretically, there exists a solution with the highest accuracy, called the exact solution. Since this exact solution is usually not obtainable with a reasonable cost, a statistical approach can be used to find a good approximation to it. To this end, the proposed nonstationary Gaussian process model is used as an *emulator* which integrates the outputs from the simulator

© 2014 American Statistical Association and the American Society for Quality
TECHNOMETRICS, AUGUST 2014, VOL. 56, NO. 3
DOI: 10.1080/00401706.2013.842935

with different mesh densities. When the simulator is expensive to run, a fast and relatively accurate emulator can be a good computational and modeling tool, especially when many combinations of the input variables need to be considered. In view of the widespread use of FEA, the proposed approach can have a wide range of applications.

This work is related to the modeling of computer experiments with multiple levels of fidelity. The existing works have focused on using the *qualitative* information of the resolution level, for example, those of Kennedy and O'Hagan (2000), Reese et al. (2004), Qian and Wu (2008). These methods are applicable if the tuning parameter takes on a few discrete values. We will show that the predictive results can be improved by using the proposed method which uses a real-valued tuning parameter. Another related method is Han, Santner, and Rawlinson (2009), which chose optimal tuning parameters to minimize the discrepancy between the computer outputs and the physical observations. Because we do not assume the existence of physical data, the method of Han et al. was not applicable here. This work is also related to that of Picheny et al. (2013). While our work focuses on modeling of deterministic computer experiments with different mesh densities, their work focuses on modeling for stochastic simulation experiments with tunable precision. Inspired by the current work, Tuo, Qian, and Wu (2013) proposed, in a discussion of Pecheny et al. (2013), a Brownian motion model for stochastic simulations and gave some mathematical justification for the model.

This article is organized as follows. In Section 2, we discuss the nature of tuning parameters and import some concepts and results from numerical analysis and finite-element analysis. In Section 2, we introduce a new class of nonstationary Gaussian process models and show how they can be employed for the problems with different mesh densities. Numerical studies given in Section 4 show the advantages of the proposed models over existing ones. The methodology is illustrated in Section 5 using a casting process simulation problem. Concluding remarks and future work are given in Section 6. In Appendix, a maximum entropy design strategy is considered and arguments are made to justify its use for multifidelity problems.

2. PHYSICAL MODEL AND MESH DENSITY

The prevailing statistical approaches for computer experiments treat computer simulation codes as black-box functions. Since the tuning parameter is part of the algorithm, a reasonable model for the tuning parameter should take the mechanism in the "black-box functions" into account. To this end, we borrow some basic concepts and results from numerical analysis in developing our models. The three basic concepts are: (1) the exact solution, (2) the approximate solution, and (3) the error.

The implementation of a computer experiment is based on a physical model. Suppose a physical model is given by a set of PDEs. The solution to this model can be used to predict the results of the corresponding physical experiment. In a computer experiment, the main interest lies in the *exact solution* to the physical model. If the physical model can be solved in an analytic form, this analytic solution is what we want. However, in our context, this analytic form does not exist so that the exact solution cannot be obtained in finite time. By using a numerical algorithm, the computer can only return an *approximate solution*. The discrepancy between the approximate solution and the exact solution is called the *error*. The size of the error can be controlled by the tuning parameter. Mesh density is the most common tuning parameter in computer experiments. As the mesh density increases, the numerical accuracy is improved, while the computational cost goes up. For a uniform mesh, the mesh size can be represented in one dimension. For a nonuniform mesh, we can also parameterize the mesh size by a multidimensional variable. In this work, we only focus on the uniform case.

The mathematical theory of finite-element methods governs the quantitative relationship between the error and the mesh density. Here, we introduce some concepts and results from Sections 1 and 2 of Brenner and Scott (2007). Suppose $\Omega \in \mathbf{R}^n$. Let $L^1_{loc}(\Omega)$ denote the set of locally integrable functions on Ω, that is, its elements are integrable on any compact subset of the interior of Ω. Let k be a nonnegative integer and $f \in L^1_{\text{loc}}(\Omega)$. Suppose that the weak derivatives $D^\alpha_w f$ exist for all $|\alpha| \leq k$, where α is a vector of integers. Define the Sobolev norm $\|f\|_{W^k_p(\Omega)} = (\sum_{|\alpha|\leq k} \|D^\alpha_w f\|^p_{L^p(\Omega)})^{1/p}$ for $1 \leq p < \infty$, where $\|\cdot\|_{L^p(\Omega)}$ is the norm of the L_p space over Ω. Define the Sobolev spaces via $W^k_p(\Omega) = \{f \in L^1_{loc}(\Omega) : \|f\|_{W^k_p(\Omega)} < \infty\}$. For a nonnegative integer k and $f \in W^k_p(\Omega)$, define the Sobolev seminorm $|f|_{W^k_p(\Omega)} = (\sum_{|\alpha|=k} \|D^\alpha_w f\|^p_{L^p(\Omega)})^{1/p}$. Let v denote the exact solution to the PDEs given by the physical model. Suppose there exists a Sobolev space $W^m_p(\Omega)$ where v lies. Let v_h denote the solution of the finite-element variational problem with the mesh density h. If the solution to the PDEs exists in the classical sense, v_h is the approximate solution given by the finite-element method. Then, the error is $v - v_h$. According to the theorems in Brenner and Scott (2007, p. 64 and p. 110), for $s \leq m$, the $\|\cdot\|_{W^s_p(\Omega)}$ norm of the error can be controlled by the following inequality

$$\|v - v_h\|_{W^s_p(\Omega)} \leq Ch^{m-s}|v|_{W^m_p(\Omega)}, \tag{1}$$

where C is independent of h and v. By specifying $m = p = 2, s = 1$ and $m = p = 2, s = 0$ in Equation (1), respectively, and defining the H^1 norm to be the $\|\cdot\|_{W^1_2(\Omega)}$ norm, we can get two important special cases of (1):

$$\|v - v_h\|_{H^1} \leq Ch\|v''\|_{L^2}, \tag{2}$$

and

$$\|v - v_h\|_{L^2} \leq Ch^2\|v''\|_{L^2}, \tag{3}$$

where v' is the generalized gradient of v, $\|v''\|_{L^2} = (\sum_{i,j} \|\frac{\partial^2 v}{\partial x_i \partial x_j}\|^2_{L^2})^{1/2}$, and the two norms are defined as $\|u\|_{H^1} = (\int_\Omega (u^T u + (u')^{\mathrm{T}} u'))^{\frac{1}{2}}$, $\|u\|_{L^2} = (\int_\Omega u^T u)^{\frac{1}{2}}$, for any u. These two norms have different physical meanings. Note that the convergence rate varies with the norm being used. In the present context, the computer output is a scalar value. This value is a functional of the underlying approximate solution. If the functional only uses the approximate solution itself like the integral operator, the L^2 norm would be appropriate. If the functional involves the derivative of the approximate solution, one should use the H^1 norm. In practice, the norm should be chosen to suit a particular need. We will revisit this topic in Section 5.

3. NONSTATIONARY GAUSSIAN PROCESS MODEL

Before proposing novel nonstationary Gaussian process models in Section 3.2, we review in Section 3.1 the required Gaussian processes.

3.1 Gaussian Processes

Stationary Gaussian process models have been extensively discussed in Santner, Williams, and Notz (2003), and Banerjee, Carlin, and Gelfand (2004). The stochastic properties of a Gaussian process $Z(\mathbf{x})$ with zero mean are determined by its covariance function $C(\mathbf{x}_1, \mathbf{x}_2)$. A Gaussian process $Z(\mathbf{x})$ with zero mean is said to be stationary if $C(\mathbf{x}_1, \mathbf{x}_2)$ can be expressed as a function of the difference between $\mathbf{x}_1$ and $\mathbf{x}_2$ (Santner, Williams, and Notz 2003, pp. 29–30), that is,

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 K(\mathbf{x}_1 - \mathbf{x}_2), \tag{4}$$

where σ^2 is the variance and K is a correlation function satisfying $K(0) = 1$. Otherwise, we call it a nonstationary Gaussian process.

For simplicity, we use the separable Gaussian correlation function throughout this article, that is,

$$K_\phi(\mathbf{x}) = \prod_{i=1}^{k} \exp\left\{-\phi_i x_i^2\right\}, \tag{5}$$

where x_i is the ith component of $\mathbf{x}$ and the ϕ_i are the correlation parameters. Other correlation function families can be considered, which will require parallel development of the methodology.

Several methods were given for constructing nonstationary covariances in Banerjee, Carlin, and Gelfand (2004, pp. 149–157). In this article, two types of nonstationary Gaussian processes on $\mathbf{R}_+ = \{t \geq 0\}$ are considered. The first is the simplest nonstationary Gaussian process, the Brownian motion (also known as the Wiener process, see Durrett 2010). The covariance function of a Brownian motion $\{B(t); t \geq 0\}$ is

$$\operatorname{cov}(B(t_1), B(t_2)) = \min(t_1, t_2). \tag{6}$$

The second one is constructed by the following strategy. Banerjee, Carlin, and Gelfand (2004, p. 150) presented this method to introduce nonstationarity through the scaling of a stationary process. Assume $Z(t)$ with the covariance (4) has variance $\sigma^2 = 1$. Let $V(t) = t^{\frac{1}{2}} Z(t)$, $(t \geq 0)$. Then, $V(t)$ is a nonstationary Gaussian process with covariance function

$$\operatorname{cov}(V(t_1), V(t_2)) = (t_1 t_2)^{\frac{1}{2}} K_\phi(t_1 - t_2). \tag{7}$$

It is clear that $\operatorname{var}(B(t)) = \operatorname{var}(V(t)) = t$. The main differences between $B(t)$ and $V(t)$ lie in the following aspects. First, the sample path of a Brownian motion is nondifferentiable, while $V(t)$ is infinitely differentiable (Santner 2003, p. 40). In addition, if we fix t_1 and let t_2 go to infinity, the asymptotic behavior of the two covariances are quite different. The covariance of $B(t)$ will stay constant because

$$\lim_{t_2 \to +\infty} \operatorname{cov}(B(t_1), B(t_2)) = \lim_{t_2 \to +\infty} \min(t_1, t_2) = t_1,$$

while that of $V(t)$ goes to 0 because

$$\lim_{t_2 \to +\infty} \operatorname{cov}(V(t_1), V(t_2)) = \lim_{t_2 \to +\infty} (t_1 t_2)^{\frac{1}{2}} \exp\{-\phi(t_1 - t_2)^2\} = 0.$$

Thus, the correlation of $B(t)$ can be much stronger than that of $V(t)$.

The best linear unbiased predictor (BLUP) for stationary Gaussian process models can be found in Santner, Williams, and Notz (2003). These results can be extended to nonstationary Gaussian process models without much difficulty. We discuss the Bayesian analysis for the nonstationary Gaussian process models based on (6) and (7) in the supplementary material (available online).

3.2 Modeling the Mesh Density

Let $\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$ be the vector of the input variables and $\mathbf{t} = (t_1, \ldots, t_k)^{\mathrm{T}}$ the vector of the tuning parameter values such as mesh densities. We assume $t_i > 0$ for each i, and a smaller t_i indicates a higher accuracy. Suppose the experimental region of interest is $\mathcal{X} \times \mathcal{T}$, where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{t} \in \mathcal{T}$. Because our interest is to predict the exact solution, we should include 0 in the closure $\overline{\mathcal{T}}$ of $\mathcal{T}$, that is, $0 \in \overline{\mathcal{T}}$. Denote the response of a computer code run by $(y, \mathbf{x}, \mathbf{t})$, where y is the computer output for the input $(\mathbf{x}, \mathbf{t})$. Since the computer code is deterministic, y is a deterministic function of $(\mathbf{x}, \mathbf{t})$, that is, $y = \eta(\mathbf{x}, \mathbf{t})$. Recall the concepts we describe in Section 2. The approximate solution is $\eta(\mathbf{x}, \mathbf{t})$. The exact solution to this physical model is denoted by $\varphi(\mathbf{x})$. As $\mathbf{t}$ gets closer to zero, the output of the computer experiment gets closer to the exact solution $\varphi(\mathbf{x})$. We can thus use the following equation to describe this relationship:

$$\eta(\mathbf{x}, \mathbf{t}) = \eta(\mathbf{x}, \mathbf{0}) + \delta(\mathbf{x}, \mathbf{t}) = \varphi(\mathbf{x}) + \delta(\mathbf{x}, \mathbf{t}), \tag{8}$$

where $\delta(\mathbf{x}, \mathbf{t})$ denotes the error with respect to the mesh density $\mathbf{t}$ at input $\mathbf{x}$.

We assume $\varphi(\mathbf{x})$ and $\delta(\mathbf{x}, \mathbf{t})$ are realizations of two mutually independent Gaussian stochastic processes $\{V(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$ and $\{Z(\mathbf{x}, \mathbf{t}) : (\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \overline{\mathcal{T}}\}$. Note that neither $E(V)$ nor $E(Z)$ is identifiable, since we can only observe $\varphi(\mathbf{x}) + \delta(\mathbf{x}, \mathbf{t})$. For convenience, we assume the separable form

$$E(V(\mathbf{x})) = f_1^{\mathrm{T}}(\mathbf{x})\beta_1,\ E(Z(\mathbf{x}, \mathbf{t})) = f_2^{\mathrm{T}}(\mathbf{t})\beta_2, \tag{9}$$

where $f_1^{\mathrm{T}}(\mathbf{x})$ and $f_2^{\mathrm{T}}(\mathbf{t})$ are vectors of known regression functions and β_1 and β_2 are vectors of unknown regression coefficients. Since the computational resource is limited, only data with t_i larger than a positive constant, say t_0, are observed. Recall that the objective is to predict for $\phi(\mathbf{x}) = \eta(\mathbf{x}, \mathbf{0})$. If the regression function $f_2^{\mathrm{T}}(\mathbf{t})$ is nonzero, the prediction will extrapolate this function to $\mathbf{t} = 0$ using only the observations with $t_i \geq t_0$. Therefore, a careful examination is needed while choosing $f_2^{\mathrm{T}}(\mathbf{t})$. We will discuss this issue further in Section 4.3.

Now we turn to the variance structure of $Z(\mathbf{x}, \mathbf{t})$. First, $Z(\mathbf{x}, \mathbf{t})$ must be a *nonstationary* process since it should satisfy the limiting condition $\lim_{\mathbf{t} \to 0} Z(\mathbf{x}, \mathbf{t}) = 0$ for any $\mathbf{x}$. Therefore, we propose the following variance structure:

$$\operatorname{var}(Z(\mathbf{x}, \mathbf{t})) = g(\mathbf{t}; \varrho), \tag{10}$$

where $g(\cdot; \varrho)$ can be a general increasing function with respect to each component of $\mathbf{t}$, and ϱ is a vector of parameters.

As discussed earlier, g should satisfy $\lim_{\mathbf{t}\to 0} g(\mathbf{t};\varrho) = 0$. We can assume that g is a polynomial function with little loss of effectiveness in modeling.

To further develop the modeling approach, we assume for the rest of the article that $\mathbf{t}$ is one-dimensional, denoted by t. This is partly justified by the fact that there is no general error bound for multivariate $\mathbf{t}$ in numerical analysis. For a typical computer experiment, the tuning parameter should be relatively small. Otherwise, its code cannot give a useful answer. Thus, to simplify the model, we assume that the higher order terms in the polynomial function are negligible, that is, we can assume the following monomial function,

$$\mathrm{var}(Z(\mathbf{x}, t)) = \sigma^2 t^l. \tag{11}$$

For limited data, which is commonly the case in expensive simulations, l is a difficult parameter to estimate and can be sensitive to the choice of t. As an alternative to the data-driven approach, we can resort to the mathematical theory in numerical analysis to guide the choice of l. Since l dominates the convergence rate of $\mathrm{var}(Z(\mathbf{x}, t))$ to 0 as $t \to 0$, its choice affects the convergence rate of the numerical algorithm to the exact solution. As discussed in Section 2 (see (1)–(3)), the error bound, denoted by e, is usually given in the form

$$|e| \le Ct^{\kappa}, \tag{12}$$

where C is independent of t. By Equation (11), we have

$$\mathbb{P}(|e| < 3\sigma t^{\frac{l}{2}}) \approx 99.7\%. \tag{13}$$

The theoretical error bound (12) and the error of the statistical model (13) must have the same order, which leads to $3\sigma t^{\frac{l}{2}} \sim Ct^{\kappa}$, as $t \to 0$. Thus, we have

$$l = 2\kappa. \tag{14}$$

For the finite-element methods, if the input variables are fixed, $\|v''\|_{L^2}$ in the upper bound of Equations (2) and (3) remains constant for different h. Then, we can obtain an appropriate l by applying Equation (2) or Equation (3) to (14): for the H^1 norm, $l = 2$ and for the L_2 norm, $l = 4$.

Nonstationary Gaussian processes with variance (11) can be flexible. Here, we consider two types of covariance structures. One is derived from the Brownian motion (see Equation (6)),

$$\mathrm{cov}(Z(\mathbf{x}_1, t_1), Z(\mathbf{x}_2, t_2)) = \sigma^2 K_{\phi}(\mathbf{x}_1 - \mathbf{x}_2)\min(t_1, t_2)^l, \tag{15}$$

where K is defined by Equation (5). The other one is derived from Equation (7),

$$\mathrm{cov}(Z(\mathbf{x}_1, t_1), Z(\mathbf{x}_2, t_2)) = \sigma^2 K_{\phi_1}(\mathbf{x}_1 - \mathbf{x}_2) K_{\phi_2}(t_1 - t_2)(t_1 t_2)^{l/2}. \tag{16}$$

Let $W(\mathbf{x}, t) = V(\mathbf{x}) + Z(\mathbf{x}, t)$. By Equation (8), $\eta(\mathbf{x}, t)$ is a realization of $W(\mathbf{x}, t)$. Since V and Z are mutually independent, the covariance function of W equals the sum of the covariance functions of V and Z. For example, if we choose the Brownian motion type covariance function (15), the covariance function of W has the form

$$\begin{aligned}&\mathrm{cov}(W(\mathbf{x}_1, t_1), W(\mathbf{x}_2, t_2))\\ &\quad= \mathrm{cov}(V(\mathbf{x}_1), V(\mathbf{x}_2)) + \mathrm{cov}(Z(\mathbf{x}_1, t_1), Z(\mathbf{x}_2, t_2))\\ &\quad= \sigma_1^2 K_{\phi_1}(\mathbf{x}_1, \mathbf{x}_2) + \sigma_2^2 K_{\phi_2}(\mathbf{x}_1, \mathbf{x}_2)\min(t_1, t_2)^l.\end{aligned} \tag{17}$$

The second model given by Equation (16) can be treated similarly.

For inference, we adopt a fully Bayesian approach. The posterior density for each parameter can be drawn via a Markov chain Monte Carlo (MCMC) algorithm (Liu 2001). Bayesian prediction for Gaussian process models is discussed in Banerjee, Carlin, and Gelfand (2004) and Qian and Wu (2008). In our model, the interest lies in predicting the exact solution φ rather than the computer output η. The predictive distribution of φ can be obtained along with the MCMC iterations. Bayesian inference for Gaussian process models is discussed in details in Santner, Williams, and Notz (2003) and Banerjee, Carlin, and Gelfand (2004). The reparameterization-based MCMC approaches (Cowles, Yan, and Smithet 2009) are helpful in sampling from the posterior. To save space, details on the Bayesian computation are given in the supplementary material (available online).

4. NUMERICAL STUDIES

In this section, we use three examples to study the behavior of the predictive mean of the proposed method. In each of the first two examples, we choose an explicit function for the exact solution and another explicit function for the error. In Example 3, the exact solution is obtained by solving a PDE.

4.1 Example 1

Suppose the true function of concern is $y(x) = \exp(-1.4x)\cos(3.5\pi x)$ (Santner, Williams, and Notz 2003, pp. 56–57), where seven points are selected as the training data. The first point is drawn randomly from [0, 1/7]. Each of the remaining six points is the previous one plus 1/7. In our context, the response is the true function plus a high-frequency noise function. We assume this noise is $e(x, t) = t^2 \sin(40x)/10$, which is a quadratic function of the tuning parameter t for fixed x. Here, we consider a three resolution experiment, by assigning a different resolution parameter to each design point in Santner, Williams, and Notz (2003). The highest resolution with $t = 1$ consists of $x_1 = 0.2152$ and $x_2 = 0.7866$. The second resolution with $t = 2$ consists of $x_3 = 0.0723$, $x_4 = 0.5009$, and $x_5 = 0.9294$. The lowest resolution with $t = 3$ consists of $x_6 = 0.3580$ and $x_7 = 0.6437$. This particular arrangement of the resolution parameters is for illustration only.

We compare the proposed method with the stationary model which ignores t. Fully Bayesian analysis is implemented for both models. For the stationary model and the stationary part of the proposed model, we consider the ordinary kriging, that is, kriging model with an unknown constant mean, which is the same as that in the example of Santner, Williams, and Notz (2003). The proposed model assumes the Brownian motion type covariance structure. The priors for both models are $1/\sigma_1^2 \sim \mathrm{Gamma}(2, 1)$, $1/\sigma_2^2 \sim \mathrm{Gamma}(2, 40)$, $\phi_1, \phi_2 \sim \mathrm{Gamma}(2, 0.1)$, where σ_2^2 and ϕ_2 only appear in the nonstationary model (see Equation (17)). We run MCMC to compute the predictive curves. After 5000 burn-in runs, 10,000 samples are drawn for inference.

Figure 1 gives the true function and predictive curves from using the two methods. The seven observations are marked in

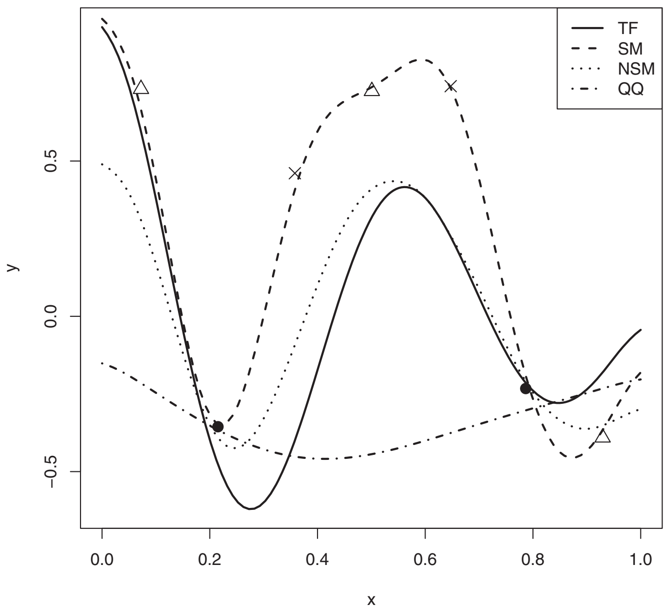

Figure 1. True and predictive curves for Example 1. TF = true function, SM = stationary model, NSM = nonstationary model, QQ = model with both qualitative and quantitative factors. The seven observations are marked by • (for $t = 1$), △ (for $t = 2$), and × (for $t = 3$).

Figure 1 by three symbols (•, △, ×) according to their t values. It is easily seen that those with the highest resolution ($t = 1$) are closest to the true functions, and those with the lowest resolution ($t = 3$) are farthest to the true function. The predictive curve of the stationary model should interpolate these points. But due to the numerical error of kriging modeling, there is a small distance between the points and the curve. Clearly, the prediction based on the proposed method tracks the true curve more closely than that based on the stationary model except near 0 or 1, which are outside the range of data. Figure 1 also shows the predictive result given by the Gaussian process model with both qualitative and quantitative factors proposed by Qian, Wu, and Wu (2008) (see also Zhou, Qian, and Zhou 2011). In the figure, it is referred to as the QQ model. It can be seen that this method has the worst performance. While the predictive curve interpolates at the two high-fidelity points x_1 and x_2, it is far away from the true curve at all the low-fidelity points (i.e., x_3 to x_7). This result suggests that the QQ model can behave poorly when the true function (i.e., the exact solution) is nonlinear and oscillatory.

4.2 Example 2

In this example, we simulate 1000 different realizations of a random function to examine the overall performance of the proposed method. Suppose the design region is $(u, v, t) \in [0, 1] \times [0, 1] \times [0.5, 1.5]$, where u, v are the model parameters and t is the tuning parameter. The true function f and the error e are chosen to be $f(u, v) = \sum_{i=1}^{10} w_{1i} \sin(w_{2i}u + w_{3i}v + w_{4i}uv + w_{5i})$ and $e(u, v, t) = w'_1 t^2 \sin(w'_2 u + w'_3 v + w'_4 uv + w'_5)$, respectively, where $w_{1,1}, \dots, w_{5,10}, w'_1, \dots, w'_5$ are independent samples from the uniform distribution over $[-1, 1]$, denoted as $U(-1, 1)$. We use a maximin Latin-hypercube of 30 points as the design for (u, v) and for each design point we sample $t \sim U(0.5, 1.5)$ independently.

Table 1. Comparison between nonstationary and stationary models using random functions

	NSM	SM_0	$SM_{0.5}$
Averaged ISE	0.1236	0.2680	0.2018
Frequency of best performance	951	2	47

Recall that the proposed model is nonstationary with respect to t. To demonstrate the effect of nonstationarity, we compare the proposed method with the stationary model with the Gaussian correlation function $K(u, v, t) = \exp\{-\phi_1 u^2 - \phi_2 v^2 - \phi_3 t^2\}$ (see Equation (5)). The proposed model assumes the Brownian motion covariance structure in Equation (15). Let $F(u, v, t) = f(u, v) + e(u, v, t)$ be the approximate solution. For the stationary model, we consider two prediction methods. They share the same stationary model but predict for different values of t. In the first method, similar to the proposed method, we predict for points with $t = 0$, which give values for the exact solution according to Equation (8). This is referred to as SM_0 in Table 1. Note that this method involves extrapolation because the experimental region for t is $[0.5, 1.5]$. Noting the fact that a stationary Gaussian process model usually performs poorly in extrapolation, we also consider another method, referred to as $SM_{0.5}$. In this method, we predict for $F(u, v, 0.5)$ with $(u, v) \in [0, 1] \times [0, 1]$, which has the highest accuracy since $t = 0.5$ is the smallest value in the region $[0.5, 1.5]$. To take advantage of a possible trend effect in t, we let each of the three models contain a linear term βt.

For the Bayesian analysis, the same priors as in Example 1 are used. In each simulation, MCMC is used to obtain the posterior distribution. After 5000 burn-in runs, 5000 samples are drawn for inference. The simulation results are summarized in Table 1. First, we compare them in terms of the integrated squared error (ISE). We define ISE as $\int_{[0,1]\times[0,1]}(f - \hat{f})^2$, where $\hat{f}$ is the predictive curve given by each of the three methods. The second row gives the averaged value of the ISE over 1000 simulations for each method. It is seen that the proposed method reduces the ISE by 53.9% and 38.8%, respectively, over the two stationary methods. The third row gives the frequency for each method with the smallest ISE. The proposed method beats the two stationary methods 951 out of 1000 times. Again, the nonstationary model outperforms the stationary model. We also observe that SM_0 behaves worse than $SM_{0.5}$. This is mostly due to the extrapolation effect, that is, SM_0 predicts for F with $t = 0$ while the training data have $t \geq 0.5$.

4.3 Example 3

Here, we choose a PDE with an analytical solution so that the behavior of the proposed method can be easily examined. Consider the following Poisson's equation:

$$\begin{cases} \Delta u = (a^2 - 2\pi^2)e^{ax}\sin(\pi x)\sin(\pi y) & \text{on } D, \\ \qquad + 2a\pi e^{ax}\cos(\pi x)\sin(\pi y), & \\ u = 0, & \text{on } \partial D, \end{cases}$$

Table 2. Numerical solutions of Poisson's equation

Run #	a	h	Exact	CPU time	Error	ΔNSM	ΔARM
1	−1	0.005	0.252	30.767	−0.006	0.001	−0.006
2	−0.8	0.01	0.276	2.304	−0.010	0.015	0.002
3	−0.6	0.008	0.303	5.469	−0.009	0.008	−0.007
4	−0.4	0.008	0.333	5.517	−0.009	0.008	−0.012
5	−0.2	0.01	0.367	2.400	−0.011	0.014	−0.023
6	0	0.005	0.405	33.111	−0.006	0.000	−0.006
7	0.2	0.01	0.448	2.434	−0.012	0.013	−0.026
8	0.4	0.008	0.497	5.785	−0.011	0.006	−0.016
9	0.6	0.008	0.552	5.783	−0.012	0.005	−0.011
10	0.8	0.01	0.614	2.592	−0.015	0.010	−0.003
11	1	0.005	0.684	35.886	−0.008	−0.002	−0.008

NOTE: Columns 4–6 are the exact results, the CPU time (in seconds), and the numerical errors, respectively. The last two columns give the difference between the exact solution and the predictive results given by the proposed nonstationary model (NSM) and the autoregressive model (ARM) of Kennedy and O'Hagan (2000).

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, $D = [0, 1] \times [0, 1]$ and a is an input variable. It is easy to verify that the exact solution given a is

$$u_a(x, y) = e^{ax} \sin(\pi x) \sin(\pi y).$$

Suppose our interest lies in computing the integral $I(a) = \int_D u_a(x, y)$, which has an analytical form $I(a) = \frac{2(e^a+1)}{a^2+\pi^2}$.

We implement a *finite-difference method* (Kincaid and Cheney 2002) to solve this equation numerically. Denote the step length by h, which is a tuning parameter of the finite-difference method. Similar to Equation (3), the L^2 norm of the numerical error can be controlled by a quadratic function of h. Here, h plays the role of the tuning parameter t.

We choose a design with 11 points for a and three levels for h. The design, the exact results, the CPU time, and the numerical results are concluded in Table 2. For convenience, we give the numerical errors in the column "Error" rather than the actual values of the numerical output. Note that the numerical error is the δ term in Equation (8), that is, the difference between the numerical output and the exact solution. The simulation is carried out on a 2.7-GHz PC. The CPU time has a small variation if we repeat running the code, but the trend with respect to the step length h is clear. From Table 2, we can see that the CPU time grows rapidly as h decreases.

We assume the regression function defined by Equation (9) has the form $\beta_0 + a\beta_1 + f(h)\beta_2$. Without loss of generality, we can assume $f(0) = 0$. The regression term should also satisfy the numerical error bound given by Equation (3). This implies that $|f(h)\beta_2| \le Ch^2$ for some constant C. Following the idea similar to that in establishing Equation (11), we can also ignore the high-order terms and assume $f(h) = h^2$. For the Bayesian analysis of the proposed method, the same priors as in Example 1 are used. MCMC is used to obtain the posterior distribution. After 5000 burn-in runs, 10,000 samples are drawn for inference.

We compare the proposed method with the autoregressive model proposed by Kennedy and O'Hagan (2000). The results are given in the last two columns of Table 2. Similar to the "Error" column, we give the difference between the predictive result and the exact solution in columns 7 and 8 for the nonstationary model (NSM) and the autoregressive model (ARM), respectively. We also consider an overall comparison, based on the mean square error (MSE) for the 11 runs. For the proposed model, it is 8.33×10^{-5}, which is much smaller than 1.77×10^{-4} for the autoregressive model. If we use the numerical results directly as predictor of the exact solution, the MSE is 1.03×10^{-4}, which is smaller than 1.77×10^{-4} for the ARM. Thus, only the proposed method achieves an improvement over the numerical results. This improvement is mainly due to the use of the regression trend term $h^2\beta_2$. An important side observation is that, if we use $h\beta_2$ instead of $h^2\beta_2$, the performance would not be satisfactory. The MSE would be 5.54×10^{-4}, much worse than the other three. Noting that $h\beta_2$ is not a correct error bound according to the results of numerical analysis (see end of Section 2), this example shows the importance of importing correct knowledge from applied mathematics in building statistical models. We also consider the stationary model, which is included in Example 2. Under the same trend term $h^2\beta_2$, this model gives an MSE value of 6.45×10^{-3}, which is much worse than the proposed method and the autoregressive model.

To show the overall performance of the proposed method, we plot in Figure 2 the curves of posterior mean and 95% limits of Bayesian credible intervals given by the proposed method. As in ordinary kriging, the credible interval should shrink near the design points. However, the length of the credible interval does not go to 0 because we do not observe an exact solution. The length of the credible interval on each design point also depends on the accuracy of the corresponding computer run. For example, we have high-accuracy runs on $a = -1, 0, 1$. Therefore, the lengths of the credible intervals on those points are the smallest. The credible intervals also shrink slightly on $a = -0.6, -0.4, 0.4, 0.6$, because on those points the computed outputs have the second-best accuracy. For the low accuracy data on $a = -0.8, -0.2, 0.2, 0.8$, the shrinkage of the credible interval is hard to discern from the figure. The overall picture suggests that the proposed method captures the prediction uncertainty faithfully.

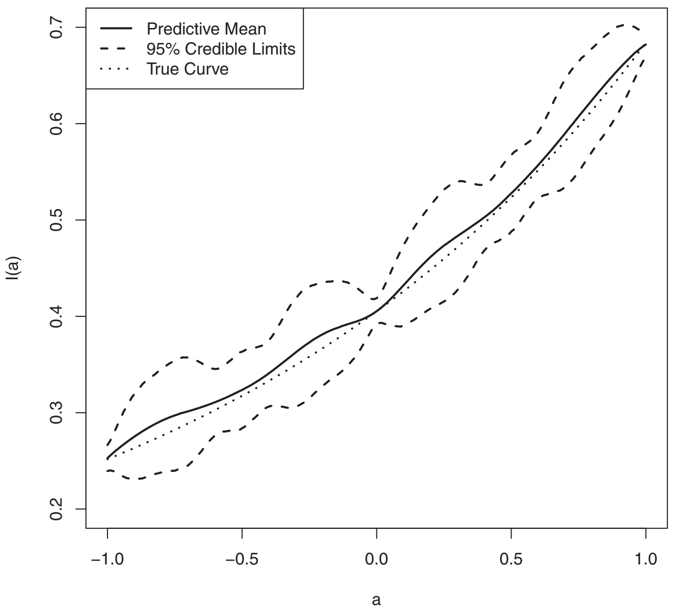

Figure 2. 95% limits of Bayesian credible intervals given by the proposed method for Example 3.

5. CASTING PROCESS SIMULATION

In this section, we examine a computer experiment problem in casting to illustrate the proposed methodology.

Shrinkage defects appear frequently in casting operations. They occur when feed metal is not available to compensate for shrinkage as the metal solidifies. Casting strength is low in the region where shrinkage defects occur. Even a slight shrinkage defect can reduce the quality of the casting. Therefore, we hope to eliminate serious shrinkage defects from the casting process. See Stefanescu (2008) for a detailed discussion. We want to study the relationship between shrinkage defect and a control variable for a specific casting problem through a computer experiment. Through the Niyama criterion proposed by Niyama et al. (1982), we can infer the possible shrinkage defects in the casting product. The Niyama criterion is a local thermal parameter defined as

$$N_y = G/\sqrt{\dot{T}}, \tag{18}$$

where G is the temperature gradient and $\dot{T}$ is the cooling rate. In the region where the Niyama value is low, serious shrinkage defect is likely to occur. To compute the Niyama function, the flow and temperature fields are needed. The simulator we use is a commercial software called InteCAST (website: *http://www.intecast.com/En/e-cae.asp*). This simulator computes the flow and temperature fields via a finite-element method.

The response y of interest is the volume of the region where the Niyama value is below a critical value of 200, which was recommended to us by a collaborating engineer. We choose a single control variable (temperature x) and a tuning parameter (the mesh size t) as inputs. The experimental region is [710°C, 750°C] × [1.5 mm, 2.5 mm].

For the choice of design points, we adopt the maximum entropy criterion. The definition, justification, and construction of the maximum entropy designs for nonstationary models are given in the appendix. In the construction, we use the Brownian motion model in Equation (15) with $\sigma_1^2/\sigma_2^2 = 30, \phi_1 = \phi_2 = 0.01$, with σ_1^2, σ_2^2 given in Equation (17). Figure 3 plots the design points from a 20-run design. We can see that the maximum entropy design tends to put more points on the low-accuracy region. This makes sense because low-fidelity runs are cheaper.

Besides these 20 training points, we also compute the value of y at the point (725, 1.5) as testing data. This point is shown in Figure 3 by the solid point. The mesh size of the testing data is chosen to be 1.5 because a mesh size smaller than 1.5 would give an out-of-memory error and thus 1.5 is the highest accuracy result which can be obtained by our computer. Table 3 gives the simulation results for both training and testing data. From Table 3, we can see that the difference between the responses with the same x but different t values can also be very large.

Here, we assume the mean of the Gaussian process is a constant, that is, $\beta = \beta_0$ and $f_V^{\mathrm{T}}(\mathbf{x}) = 0$. By Equation (18), the definition of the Niyama value involves the derivative of the thermal field. Therefore, the H^1-type error bound is more appropriate because it measures the discrepancy between the derivatives of two functions. In this case, by Equations (2), (12), and (14), we have $l = 2$. Two models are considered here. Model I is the Brownian motion model in Equation (15). Model II has the

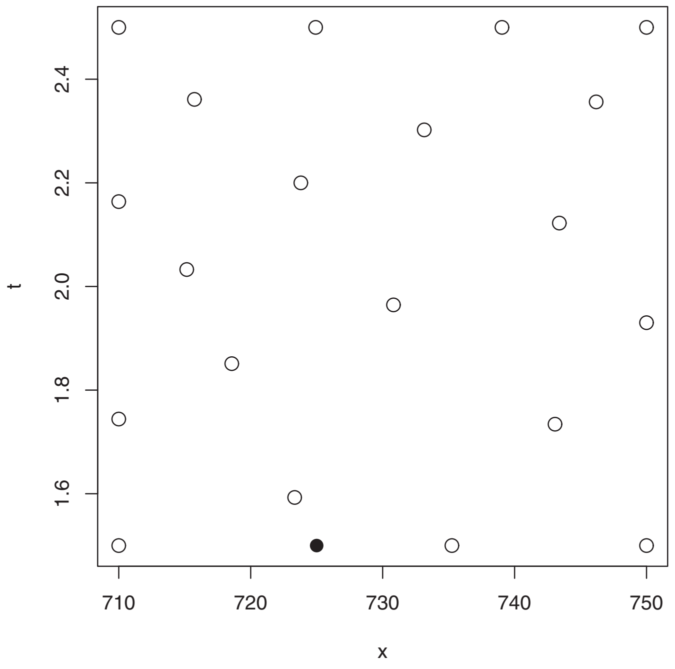

Figure 3. Design points for the casting experiment are shown by the circles and the testing point is given by the solid point.

covariance structure in Equation (16). The priors for Model I are $1/\sigma_1^2, 1/\sigma_2^2 \sim \text{Gamma}(2, 1), \phi_1, \phi_2 \sim \text{Gamma}(2, 0.1)$. The priors for Model II are $\phi_3 \sim \text{Gamma}(2, 0.1)$, while the other priors remain the same as Model I. Note that this choice of hyperparameters differs from that used in Examples 1–3. From our experience, some of the hyperparameters need to be chosen carefully to obtain reasonable results. In fact, other kriging methods for multifidelity data (e.g., Kennedy and O'Hagan

Table 3. Casting experiment data

Run #	x(°C)	t(mm)	y(cm^3)
1	710	2.5	189.67
2	739.05	2.5	178.16
3	724.92	2.5	184.05
4	750	2.5	175.42
5	715.74	2.36	149.94
6	746.18	2.36	137.79
7	733.15	2.30	153.91
8	723.80	2.20	160.19
9	710	2.16	197.92
10	743.39	2.12	190.15
11	715.15	2.03	208.49
12	730.82	1.96	137.17
13	750	1.93	149.20
14	718.57	1.85	196.04
15	710	1.74	195.00
16	743.07	1.73	175.31
17	723.33	1.59	161.76
18	710	1.5	172.94
19	735.25	1.5	165.85
20	750	1.5	159.53
21	725	1.5	167.00

NOTE: Runs 1–20 were used to train the model, and run 21 for testing.

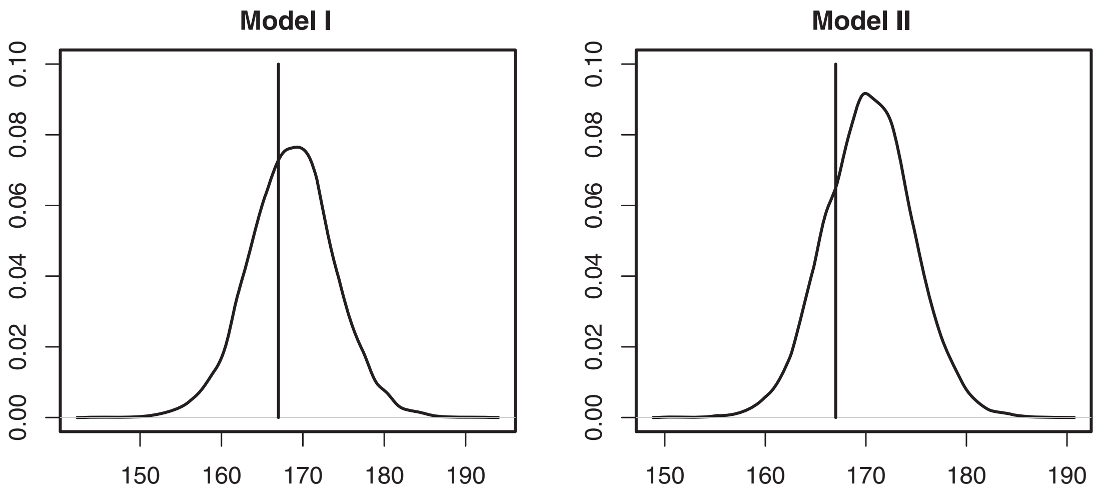

Figure 4. Predictive density, casting experiment.

2000; Qian and Wu 2008) are also sensitive to the choice of the hyperparameters. Multifidelity data do not generally contain enough information for the kriging modeling. Therefore, informative priors should be specified to complement this information. More discussions on choosing the hyperparameters for the proposed method are presented in the supplementary material (available online). For both models, we want to predict the high-accuracy output $y(725, 1.5)$.

Through Slice–Gibbs sampling (Agarwal and Gelfand 2005), we obtain 10,000 production runs for posterior calculations after 5000 burn-in iterations for each model. Prediction for the testing data is done simultaneously in each MCMC iteration. Figure 4 plots the prediction densities obtained by a kernel density smoother for the two models, where the vertical line in each plot indicates the true value (= 167) of the high-accuracy output $y(725, 1.5)$. From the results, both models give appropriate predictions. The predicted result with Model II has a smaller variance than Model I but is slightly biased if we regard the testing data $y(725, 1.5)$ as the exact solution.

6. DISCUSSION AND FUTURE WORK

Mesh density in finite-element analysis (FEA) is one of the most commonly used tuning parameters. Choice of the mesh density affects the numerical performance and has implications on the computational cost. In this article, by using the concept of physical model and exact solution, we state the goal of the computer experiment as that of finding a good approximation to the exact solution to the physical model.

To model the exact solution, we propose a new kriging model based on a nonstationary Gaussian process. The model integrates the computer outputs of different mesh densities and provides approximation to the exact solution. Concepts and results in numerical analysis are imported to build and justify this model. For FEA, we consider the error bounds given by Equation (1). But in some extreme conditions, Equation (1) may not be satisfied. Thus, further investigation on extending Equation (1) and the associated variance structure in Equation (11) will be of interest. Another important issue is multidimensional tuning parameters. To develop an appropriate statistical model, one needs to study how these parameters control the accuracy of the output and the joint effect of the parameters.

Given the variance structure, there are various choices of the covariance function. We believe that the choice does not matter much in the prediction performance. In this article, we suggest two covariance structures, given by Equations (15) and (16). In practice, we prefer Equation (15) because of the Markovian property of the Brownian motion. For example, we may obtain the computer outputs of the same input variable with different accuracies, for example, an iterative algorithm returns a sequence of outputs with increasing accuracies. The common practice in computing is to use the finest result only. A nonstationary process with the Markovian property can be used to justify this practice because the low-accuracy results for the same input variable are not used or needed for prediction.

APPENDIX

MAXIMUM ENTROPY DESIGNS FOR NONSTATIONARY MODELS

The design problem for multifidelity computer experiments has received considerable attention in the last few years. For two-fidelity data, Qian and Wu (2008) proposed a nested design constructed by the maximin distance criterion. Qian, Ai, and Wu (2009), Qian (2009), and Qian and Wu (2009) proposed nested Latin hypercube designs, which can be extended to more than two fidelities. The generic strategy for these designs is to take a large number of lower fidelity observations to obtain a macroscopic understanding of the response and select a small subset for higher fidelity runs to supplement some detailed information. Because a low-fidelity run is much cheaper than a high-fidelity run, the total cost can still be kept low. Note that, for each level of fidelity, the designs employed in the aforementioned papers have the space-filling property.

Space-filling designs (Santner, Williams, and Notz 2003) such as Latin hypercube designs (McKay et al. 1979), maximin distance designs (Johnson et al. 1990), and uniform designs (Fang, Li, and Sudjianto 2006) are widely used in computer experiments. Recall that in a space-filling design, observations are spread evenly throughout the experimental region. An explanation or justification for this approach is that interest lies in the whole experimental region because we have no knowledge to decide in which part to take more observations. Because of the absence of information on the relative importance of each observation, we assume they are homogeneous. Thus, a stationary Gaussian process model is adopted. However, observations with different accuracies should not be assumed to be homogeneous, which is why nonstationary Gaussian process models are considered in Section 3.2. Thus, design nonuniformity should follow from model nonstationarity.

Here, we consider the maximum entropy criterion (Shewry and Wynn 1987), which is based on information-theoretical ideas. It can facilitate design construction for various statistical models. It works by finding a design to maximize the expected change in information after the experiment is run. Sacks et al. (1989) and Santner, Williams, and Notz (2003) discussed its applications in computer experiments and showed that for stationary Gaussian process models the maximum entropy criterion can be reduced to the maximization of det(K), where K is the correlation matrix.

Here, we extend the usage of maximum entropy designs to nonstationary Gaussian process models. Through algebraic calculations similar to Santner, Williams, and Notz (2003, pp. 166–167), the entropy criterion for our nonstationary model can be reduced to

$$\det(\mathrm{cov}(y(\mathbf{X}, \mathbf{T})), \tag{A.1}$$

where $(x, t) \in D = \mathcal{X} \times \mathcal{T}$. Here, a design maximizing Equation (A.1) is called a *maximum entropy design*.

Exchange algorithms can be applied to maximize Equation (A.1). For instance, Currin et al. (1991) described an algorithm adopted from DETMAX (Mitchell 1974) for finding maximum entropy designs.

SUPPLEMENTARY MATERIALS

Bayesian Analysis: Details of the Bayesian analysis of the model, including discussion of choice of prior hyperparameters. (PDF file)

Code and data: C++ code implementing the model and input data for Example 3. (zip file)

ACKNOWLEDGMENTS

The authors are grateful to Peter Z. Qian for his initial suggestion which inspired us to work on the problem, to Ke Yin on a discussion about partial differential equations, to the Editor, Associate Editor, and two referees for very valuable comments. Tuo's research is partially supported by the National Natural Science Foundation of China 11271355. Wu's research is supported by NSF grants DMS 0705261 and 1007574. The work was completed while Tuo was visiting Georgia Institute of Technology.

[Received April 2012. Revised August 2013.]

REFERENCES

Agarwal, D. K., and Gelfand, A. E. (2005), "Slice Sampling for Simulation Based Fitting of Spatial Data Models," *Statistics and Computing*, 15, 61–69. [379]

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004), *Hierarchical Modeling and Analysis for Spatial Data*, London: Chapman and Hall/CRC Press. [374,375]

Brenner, S. C., and Scott, L. R. (2007), *The Mathematical Theory of Finite Element Methods* (3rd ed.), New York: Springer. [373]

Cowles, M., Yan, J., and Smith, B. (2009), "Reparameterized and Marginalized Posterior and Predictive Sampling for Complex Bayesian Geostatistical Models," *Journal of Computational and Graphical Statistics*, 18, 262–282. [375]

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, 86, 953–963. [380]

Durrett, R. (2010), *Probability: Theory and Examples* (4th ed.), New York: Cambridge University Press. [374]

Fang, K. T., Li, R., and Sudjianto, A. (2006), *Design and Modeling for Computer Experiments*, London: Chapman and Hall/CRC Press. [379]

Han, G., Santner, T. J., and Rawlinson, J. J. (2009), "Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments," *Technometrics*, 51, 465–474. [373]

Kennedy, M. C., and O'Hagan, A. (2000), "Predicting the Output From a Complex Computer Code When Fast Approximation Are Available," *Biometrika*, 87, 1–13. [373,377,379]

——— (2001), "Bayesian Calibration of Computer Models" (with discussion), *Journal of the Royal Statistical Society,* Series B, 63, 425–464. [372]

Kincaid, D. R., and Cheney, E. W. (2002), *Numerical Analysis: Mathematics of Scientific Computing* (3rd ed.), Providence: American Mathematical Society. [377]

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer. [375]

Mitchell, T. J. (1974), "An Algorithm for the Construction of 'D-Optimal' Experimental Designs," *Technometrics*, 16, 203–210. [380]

Niyama, E., Uchida, T., Morikawa, M., and Saito, S. (1982), "A Method of Shrinkage Prediction and Its Application to Steel Casting Practice," *International Cast Metals Journal*, 7, 52–63. [378]

Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2013), "Optimization of Noisy Computer Experiments With Tunable Precision," *Technometrics*, 1, 2–13. [373]

Qian, P. Z. G. (2009), "Nested Latin Hypercube Designs," *Biometrika*, 96, 957–970. [379]

Qian, P. Z. G., Ai, M., and Wu, C. F. J. (2009), "Construction of Nested Space-Filling Designs," *The Annals of Statistics*, 37, 3616–3643. [379]

Qian, P. Z. G., and Wu, C. F. J. (2008), "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, 50, 192–204. [373,375,379]

Quian, P. Z. G., and Wu, C. F. J. (2009), "Sliced Space-Filling Designs," *Biometrika*, 96, 945–956. [379]

Qian, P. Z. G., Wu, H, and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396. [376]

Reese, S., Wilson, A., Hamada, M., Martz, H., and Ryan, K. (2004), "Integrated Analysis of Computer and Physical Experiments," *Technometrics*, 46, 153–164. [373]

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435. [380]

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer Verlag. [372,374,375,379,380]

Shewry, M. C., and Wynn, H. P. (1987), "Maximum Entropy Sampling," *Journal of Applied Statistics*, 14, 165–170. [380]

Stefanescu, D. M. (2008), *Science and Engineering of Casting Solidification* (2nd ed.), New York: Springer. [378]

Tuo, R., Qian, P. Z. G, and Wu, C. F. J. (2013), "A Brownian Motion Model for Stochastic Simulation With Tunable Precision," Comments on "Quantile-Based Optimization of Noisy Computer Experiments With Tunable Precision," by Picheny et al., *Technometrics*, 1, 29–31. [373]

Zhou, Q., Qian, P. Z. G., and Zhou, S. (2011), "A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors," *Technometrics*, 53, 266–273. [376]