Modeling of Computer Experiments with Different Mesh Densities

Rui Tuo*
Academy of Mathematics and Systems Science
Chinese Academy of Sciences

C. F. Jeff Wu
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dan Yu
Academy of Mathematics and Systems Science
Chinese Academy of Sciences

April 18, 2012

Abstract

This article considers deterministic computer experiments with real-valued tuning parameters which determine the accuracy of the numerical algorithm. The aim is to integrate computer outputs with different tuning parameters. Finite element method with mesh density as the tuning parameter is an important case. Novel nonstationary Gaussian process models are proposed to establish a framework consistent with the results in numerical analysis. Maximum entropy designs are considered for experimental planning. Numerical studies show the advantages of the proposed method. The methodology is illustrated with a problem in casting simulation.

Keywords: Brownian Motion; finite element analysis; kriging; multi-resolution data; nonstationary Gaussian process models; tuning parameters

*The authors are grateful to Peter Z. G. Qian for inspiring our interests in the research problem and to Ke Yin on a discussion about partial differential equations. Wu's research is supported by NSF grants DMS 0705261 and 1007574.

1 Introduction

Numerical computations like finite element analysis (FEA) are commonly used in simulating real world phenomena like soil erosion, hourly temperature, 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 take the respective advantages of the two. This is particularly useful if many combinations of the input variables should be considered as 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 paper is to develop a framework for doing this and to propose a class of nonstationary Gaussian process models to link the outputs of simulation runs with different mesh densities in order 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 PDE but with different initial or boundary conditions. Input variables can be either control variables (Santner, Williams and Notz 2003, p. 5) 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 paper 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. And stationary Gaussian process models would be suitable for modeling the computer outputs (Santner et al. 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.

In order 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 theory of error bounds in finite element method. 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 is usually not obtainable with a reasonable cost, a statistical approach can be used to find a good approximation to the exact solution with reasonable cost. To this end, the proposed nonstationary Gaussian process model is used as an *emulator* which is based on the output of the simulators with different mesh densities. When the simulators are 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 wide-spread use of FEA, the proposed approach can have a wide range of applications.

This work is related to the modeling of computer experiments with multi-fidelity. The existing work focus on using the *qualitative* information of the resolution level, e.g., 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 utilizes a real-valued tuning parameter. Another related method is Han et al. (2009), which chooses 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. is not applicable here.

This paper is organized as follows. In Section 2 we discuss the nature of tuning parameters and introduce some concepts and results borrowed from numerical analysis and finite element method. In Section 3 we introduce nonstationary Gaussian process models and construct a new model for the mesh density. Numerical studies given in Section 4 show the advantages of the proposed models over existing ones which do not incorporate real-valued tuning variables. In Section 5 a maximum entropy design strategy is considered and shown to be more suitable for multi-fidelity problems. The methodology is illustrated in Section 6 using a casting process simulation problem. Concluding remarks and future work are given in Section 7.

2 Physical Model and Mesh Density

The prevailing statistical approaches for computer experiments treat computer simulation codes as black-box functions and provide surrogate models for these functions. Since the tuning parameter is part of the algorithm, a reasonable model for the tuning parameter should consider the mechanism in the "black-box functions". 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. A physical model is given by a set of PDEs which can be solved by some numerical algorithm. 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 non-uniform mesh, we can also parameterize the mesh size by a multi-dimensional variable. In this work, we only focus on the uniform case.

The mathematical theory of the finite element methods governs the quantitative relationship between the error and the mesh density. Here we introduce some concepts and results from Brenner and Scott (2007). Suppose $\Omega \in \mathbf{R}^n$. Let $L^1_{loc}(\Omega)$ denote the set of locally integrable functions on Ω, i.e., its elements are integrable on any compact subset of the interior of Ω. Let k be a non-negative integer and $f \in L^1_{loc}(\Omega)$. 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 $\|f\|_{W^k_p(\Omega)} = \Big(\sum_{|\alpha| \leq k} \|D^\alpha_w f\|^p_{L^p(\Omega)} \Big)^{1/p}$ 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 $W^k_p(\Omega) = \Big\{ f \in L^1_{loc}(\Omega) : \|f\|_{W^k_p(\Omega)} < \infty \Big\}$. For a non-negative integer k and $f \in W^k_p(\Omega)$,

define the Sobolev semi-norm $|f|_{W_p^k(\Omega)} = \Big(\sum_{|\alpha|=k} \|D_w^\alpha f\|_{L^p(\Omega)}^p \Big)^{1/p}$. Let v denote the exact solution to the PDEs given by the physical model. Suppose there exists a Sobolev space $W_p^m(\Omega)$ 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 $\|\cdot\|_{W_p^s(\Omega)}$ norm of the error can be controlled by the following inequality

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

where C is independent of h and v. By specifying $m = p = 2, s = 1$ and $m = p = 2, s = 0$ in (1) respectively and defining the H_1 norm to be the $\|\cdot\|_{W_2^1(\Omega)}$ 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, $\|v''\|_{L^2} = \Big(\sum_{i,j} \big\| \frac{\partial^2 v}{\partial x_i \partial x_j} \big\|_{L^2}^2 \Big)^{1/2}$ and the two norms are defined as $\|u\|_{H^1} = \big(\int_\Omega \big(u^2 + (u')^{\mathrm{T}} u' \big) \big)^{\frac{1}{2}}$, $\|u\|_{L^2} = \big(\int_\Omega u^2 \big)^{\frac{1}{2}}$, for any u. These two norms have different physical meanings. Because the convergence rate varies with the norm being used, the requirement for smoothness affects the accuracy. In the present context, the computer output is a single 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 6.

3 Nonstationary Gaussian Process Model

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

3.1 Gaussian Processes

Stationary Gaussian process models have been extensively discussed in Santner et al. (2003), and Banerjee et al. (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 et al. 2003 pp. 29-30), i.e.,

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

where σ^2 is the variance and K is the 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, i.e.,

$$K_\phi(\mathbf{x}_1, \mathbf{x}_2) = \prod_{i=1}^{k} \exp\{-\phi_i (x_{i1} - x_{i2})^2\}, \tag{5}$$

where x_{i1} and x_{i2} are the ith components of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. 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 et al. (2004, pp. 149-157). In this paper, 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

$$\mathrm{Cov}(B(t_1), B(t_2)) = \min(t_1, t_2). \tag{6}$$

The second one is constructed by the following strategy. Banerjee et al. (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 1. Let $V(t) = t^{\frac{1}{2}} Z(t), (t \geq 0)$. Then $V(t)$ is a nonstationary Gaussian process with covariance function

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

It is clear that $\mathrm{Var}(B(t)) = \mathrm{Var}(V(t)) = t$. 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 goes to infinity, the approximate performances of the two covariances are quite different. The covariance of $B(t)$ will stay constant because

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

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

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

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 et al. (2003). These results can be extended to nonstationary Gaussian process models without much difficulty. In Section 4, we will focus on the Bayesian analysis for the nonstationary Gaussian process model to be proposed as follows.

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 that represents a specific mesh for a computer experiment run. 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 $\mathcal{T}$, i.e., $0 \in \overline{\mathcal{T}}$. 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 $(\mathbf{x}, \mathbf{t})$, i.e., $y = \eta(\mathbf{x}, \mathbf{t})$. 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 following separable form

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

where $f_1^{\mathrm{T}}(\mathbf{x})$ and $f_2^{\mathrm{T}}(\mathbf{t})$ are vectors of known regression functions, 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 $f_Z^{\mathrm{T}}(\mathbf{x}, \mathbf{t})$ involves a term with t, 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 consider this issue in Section 4.2.

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

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

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 above, g should statisfy $\lim_{\mathbf{t} \to 0} g(\mathbf{t}; \varrho) = 0$. 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 paper 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 in order to simplify the model, we assume that the higher order terms in the polynomial function are negligible, i.e., 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 following form:

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

where C is independent of t. By (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 (2) and (3) remains constant for different h. Then we can obtain an appropriate l by applying (2) or (3) to (14): for the H^1 norm, $l = 2$ and for the L^2 norm, $l = 4$.

We assume $\mathbf{t}$ is one dimension, because there is no general error bounds for multivariate $\mathbf{t}$ in the numerical analysis literature. 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 (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 (5). The other one is derived from (7),

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

$\eta(x, t) \to \varphi(x)$ as $t \to 0$.

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 et al. (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. Baysian inference for Gaussian process models is discussed in

details in Santner et al. (2003) and Banerjee et al. (2004). The reparameterization-based MCMC approaches (Cowles et al. 2009) are helpful in sampling from the posterior. To save space, details on the Bayesian computation are given in the supplemental materials.

4 Numerical Studies

In this section, we use two examples to study the behavior of the predictive mean of the proposed method. The first one is constructed under the assumption of the proposed model. The second one is from a PDE problem.

4.1 Example 1

Suppose the true function of concern is $y(x) = \exp(-1.4x)\cos(3.5\pi x)$ (Santner et al., 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 6 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 et al. (2003). The highest resolution with $t = 1$ consists of $x_1 = .2152$ and $x_2 = .7866$. The second resolution with $t = 2$ consists of $x_3 = .0723$, $x_4 = .5009$ and $x_5 = .9294$. The lowest resolution with $t = 3$ consists of $x_6 = .3580$ and $x_7 = .6437$. This particular arrangement of the resolution parameters is for illustration only. The 21 testing data are $i/20$ for $i = 0, \ldots, 20$.

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, i.e., kriging model with an unknown constant mean, which is the same as that in the example of Santner et al. (2003). The proposed model takes the Brownian Motion type covariance structure. The priors for both models are $\sigma_1^2 \sim Gamma(2,1), \sigma_2^2 \sim Gamma(2,40), \phi_1, \phi_2 \sim Gamma(2, 0.1)$, where σ_2^2 and ϕ_2 only appear in the nonstationary model. We run MCMC to compute the means.

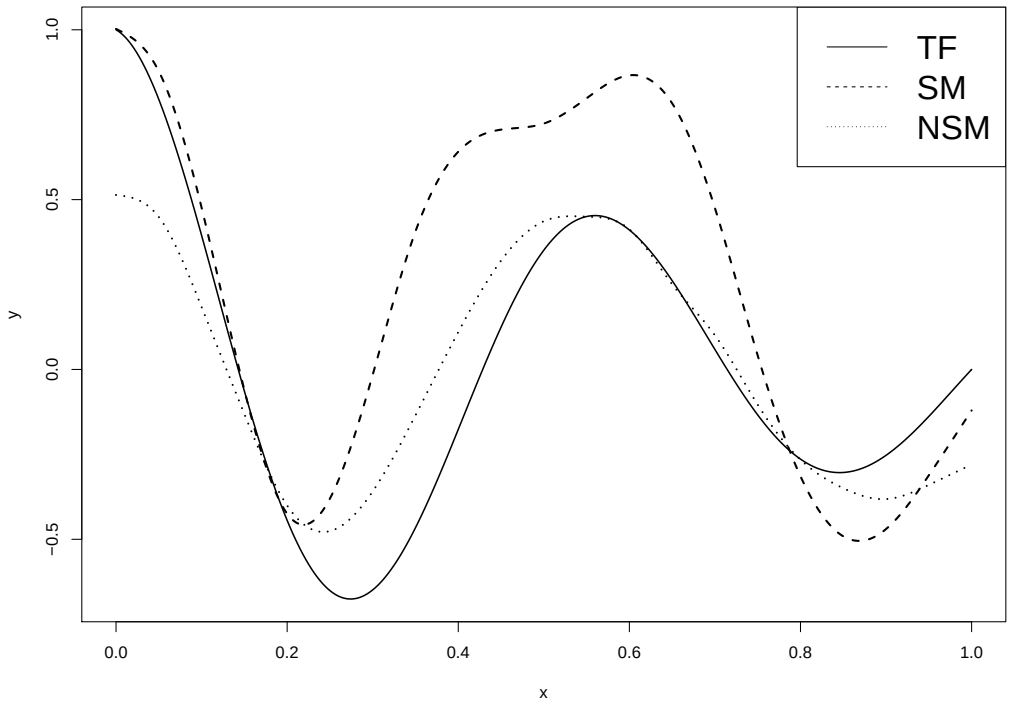

Figure 1: True and predictive curves for the synthetic example. TF = True Function, SM = Stationary Model, NSM = Nonstationary Model.

Figure 1 gives the true function and predictive curves from using the two methods. The predictive curves are plotted using smoothing spline. 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.

4.2 Example 2

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

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

where $D = [0,1] \times [0,1]$ and a is an input variable. The exact solution given a is

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

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

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 (3), the L^2 norm of the numerical error can be controlled by a quadratic function of h.

We choose a design with 11 points for a and three resolutions for h. The design, the exact results, the CPU time and the numerical results, are concluded in Table 1. 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 $\delta(x,t)$ term in (8), i.e., the difference between the numerical output and the exact solution. 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 1 we can see that the CPU time grows rapidly as h decreases.

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

Table 1: Numerical solutions of Poisson's equation. Column 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).

We assume the regression function defined by (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 (3). This implies that $|f(h)\beta_2| \leq Ch^2$ for some constant C. Following the idea similar to that in establishing (11), we can also ignore the high order terms and assume $f(h) = h^2$.

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 in Table 1. 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 in numerical analysis (see end of Section 2), it shows the importance of importing correct knowledge from applied mathematics in building statistical models.

5 Design of Experiment

The design problem for computer simulations with different accuracies has received considerable attentions in the literature. 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-accuracy observations to obtain a macroscopic understanding of the response and select a small subset for higher-accuracy runs to supplement some detailed information. Because a low-accuracy run is much cheaper than a high-accuracy run, the

total cost can still be kept low.

In this section, an experimental design methodology will be presented for computer experiments with real-valued tuning parameters and arbitrary simulation runs based on the proposed model. The goal is to obtain more information from the experiment and still reduce its cost.

In traditional computer experiments, space-filling designs (Santner et al. 2003) such as Latin hypercube designs and maximin distance designs are widely used. In a space-filling design observations are spread evenly throughout the experimental region. An explanation for this approach is that our interests are in the whole experimental region because we have no knowledge to decide in which part we should take more observations. In the modeling, because of the absence of information about the 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. And this is why we have proposed a nonstationary Gaussian process model for this type of problems. For the same reason, we should not take the observations uniformly over the experimental region and should look for alternatives to space-filling designs.

Although there is a vast literature on designs for computer experiments, none of them address the situations considered in this paper. Here we consider the maximum entropy criterion (Shewry and Wynn, 1987), which is based on information-theoretic 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 et al. (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 and its corresponding correlation function $K(\cdot)$ is defined in (4). For a known correlation function $K(\cdot)$, Currin et al. (1991) described an algorithm adopted from DETMAX (Mitchell 1974) for finding maximum entropy designs.

Here we extend the usage of maximum entropy designs to nonstationary Gaussian process models. Through algebraic calculations similar to Santner et al. (2003, pp. 166-167),

the entropy criterion for our nonstationary Gaussian process model can be reduced to

$$\det(\mathrm{Cov}(y(\mathbf{X}, \mathbf{T})), \tag{17}$$

where $(x, t) \in D = \mathcal{X} \times \mathcal{T}$. Here a design maximizing (32) is called a *maximum entropy design*. For the covariance structure (15), $\mathrm{Cov}(y(\mathbf{X}, \mathbf{T}))$ is given by (20). An interesting property of the maximum entropy designs is that they tend to place more points in the region with larger response variance. This holds because these points would provide more information since the uncertainty in the region is larger before the experiment. Formally, this property can be stated in Theorem 1 in a simple case.

Theorem 1. Suppose $\mathcal{D} = \mathcal{X} \times \{t_1, t_2\}$, $0 < t_1 < t_2$, and

$$\mathrm{Cov}(y(\mathbf{x}_1, a_1), y(\mathbf{x}_2, a_2)) = C(\mathbf{x}_1, \mathbf{x}_2) \min(a_1, a_2), \tag{18}$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, $a_1, a_2 \in \{t_1, t_2\}$, C is any covariance function on $\mathcal{X} \times \mathcal{X}$. Suppose there exist an n-point design which maximizes (17) and the maximum is positive. Denote this maximum entropy design by

$$D = \{(\mathbf{x}_{11}, t_1), \ldots, (\mathbf{x}_{1n_1}, t_1), (\mathbf{x}_{21}, t_2), \ldots, (\mathbf{x}_{2n_2}, t_2)\},$$

where $n_1 + n_2 = n$.

Then $n_1 \leq n_2$.

Proof. See Appendix. □

As is shown in Theorem 1, for the two-resolution experiment, the maximum entropy design automatically places more points on the low-accuracy level (i.e., at t_2). This helps reduce the total cost of the experiment because the low-accuracy runs are cheaper. This property is consistent with the strategy in existing methods.

6 Casting Process Simulation

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

Shrinkage defects appear frequently in foundry. 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 slight shrinkage defect can reduce the quality of the casting. Therefore casting with serious shrinkage defect should be eliminated. See Stefanescu (2008) for 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

$$N_y = G/\sqrt{\dot{T}}, \tag{19}$$

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. In order to compute the Niyama function, the flow and temperature fields are needed. The emulator 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 $[710℃, 750℃] \times [1.5\text{mm}, 2.5\text{mm}]$.

We use the maximum entropy design suggested in Section 5. In the construction, we use the Brownian Motion model in (15) with $\lambda = \frac{1}{30}, \phi_1 = \phi_2 = 0.01$. Figure 2 plots the design points of 20 runs in the design region. We can see from Figure 2 that the maximum entropy design tends to put more points on the low-accuracy region.

Besides these 20 training points, we also compute the value of y at the point $(725, 1.5)$ as testing data. 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 2 gives the simulation results for both training and testing data. From Table 2 we can see that the difference of the response for the same x but different t's can be very large.

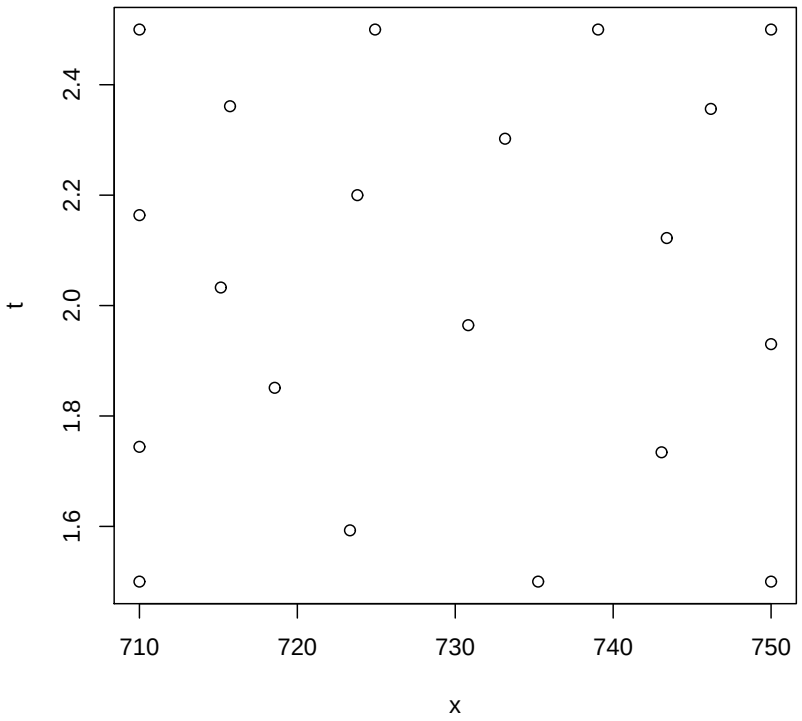

Figure 2: Design points, casting experiment.

6.0.1 Analysis

Here we assume the mean of the Gaussian process is a constant, i.e., $\beta = \beta_0$ and $f_V^{\mathrm{T}}(\mathbf{x}) = 0$. By (19), 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 (2), (12) and (14) we have $l = 2$. Two models are considered here. Model I is the Brownian Motion model in (15). Model II has the covariance structure in (16). The priors for Model I are $\sigma_1^2, \sigma_2^2 \sim Gamma(2, 1), \phi_1, \phi_2 \sim Gamma(2, 0.1)$. The priors for Model II are $\phi_3 \sim Gamma(2, 0.1)$, while the other priors remain the same with Model I. For both models, we predict the exact physical solution $y(735, 0)$.

Through Slice-Gibbs sampling (Agarwal and Gelfand 2005), we obtain 10000 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 3 plots the

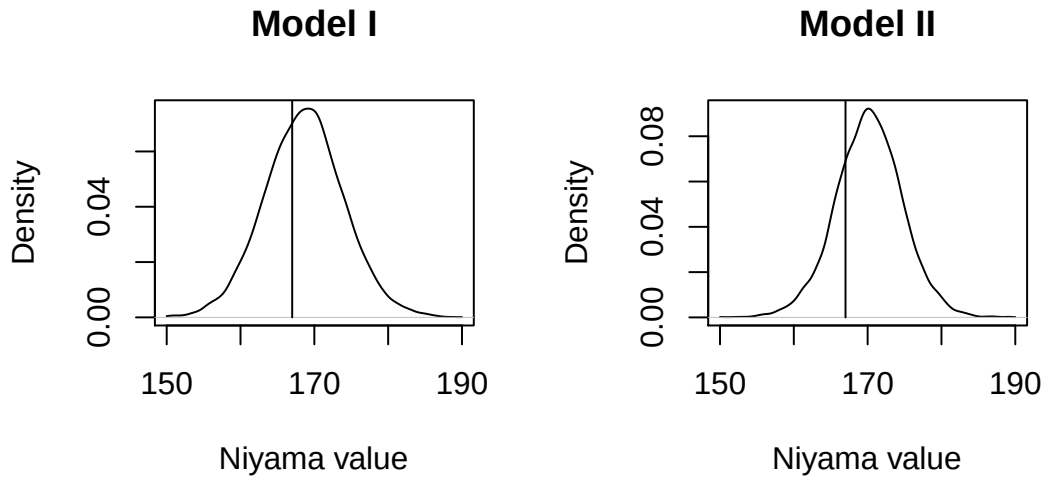

Figure 3: Predictive density, casting experiment.

prediction densities obtained by a kernel density smoother for the two cases, where the vertical line in each plot indicates the true value of the high-accuracy output $y(735, 1.5)$. 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 $y(735, 1.5)$ as the exact solution.

7 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 but is independent of the exact solution. It also has implications on the computational cost. In this article, by using the concept of physical model and exact solution, the goal of the computer experiment can be stated as that of finding good approximation to the exact solution to the physical model.

To model the exact solution, we propose a new kriging model based on a non-stationary 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 (1). But in some extreme conditions, (1) may not be satisfied. Thus further investigation on extending (1) and the associated variance structure in (11) will be of interest. Another important issue is multi-dimensional 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 (15) and (16). In practice, we prefer (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, e.g., 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.

Maximum entropy designs are suggested for the proposed models. Theorem 1 shows that these designs put more points on the low-accuracy level for a special covariance structure. Note that the condition of Theorem 1 does not hold in the casting example because the covariance function is

$$\sigma_1^2 K_{\phi_1}(x_1, x_2) + \sigma_2^2 K_{\phi_2}(x_1, x_2) \min(t_1^2, t_2^2), \tag{20}$$

which does not satisfy (18). However, this property is still demonstrated in Figure 2. Thus we expect Theorem 1 can be extended to more general nonstationary Gaussian process models. Further theoretical investigation of the maximum entropy designs is hence warranted.

Appendix

Proof of Theorem 1. We will prove the result by showing that the contrary, i.e., $n_1 > n_2$ will lead to a contradiction. if $n_1 > n_2$, for the maximum entropy design D, the covariance

matrix $\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T}))$ has the following form:

$$\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T})) = \begin{pmatrix} t_1 A_1 & t_1 A_2 \\ t_1 A_2^{\mathrm{T}} & t_2 A_3 \end{pmatrix},$$

where A_1 is a $n_1 \times n_1$ matrix and A_3 is a $n_2 \times n_2$ matrix. Consider the design point of

$$D^* = \{(\mathbf{x}_{11}, t_2), \ldots, (\mathbf{x}_{1n_1}, t_2), (\mathbf{x}_{21}, t_1), \ldots, (\mathbf{x}_{2n_2}, t_1)\}.$$

Then D^* place n_2 points on t_1 and n_1 point on t_2. Using the notation of (21), the covariance matrix can be expressed as

$$\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T}^*)) = \begin{pmatrix} t_2 A_1 & t_1 A_2 \\ t_1 A_2^{\mathrm{T}} & t_1 A_3 \end{pmatrix}.$$

A_1 is invertible because $\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T}))$ is positive definite. Thus

$$\begin{aligned} & \det \begin{pmatrix} t_1 A_1 & t_1 A_2 \\ t_1 A_2^{\mathrm{T}} & t_2 A_3 \end{pmatrix} \\ = \; & \det(t_1 A_1) \det(t_2 A_3 - t_1 (A_2^{\mathrm{T}} A_1^{-1} A_2)) \\ = \; & t_1^{n_1 - n_2} \det(A_1) \det(t_1 t_2 A_3 - t_1^2 (A_2^{\mathrm{T}} A_1^{-1} A_2)), \end{aligned}$$

and

$$\begin{aligned} & \det \begin{pmatrix} t_2 A_1 & t_1 A_2 \\ t_1 A_2^{\mathrm{T}} & t_1 A_3 \end{pmatrix} \\ = \; & \det(t_2 A_1) \det(t_1 A_3 - \frac{t_1^2}{t_2} (A_2^{\mathrm{T}} A_1^{-1} A_2)) \\ = \; & t_2^{n_1 - n_2} \det(A_1) \det(t_1 t_2 A_3 - t_1^2 (A_2^{\mathrm{T}} A_1^{-1} A_2)). \end{aligned}$$

Hence we have $\det(\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T}^*))) > \det(\mathrm{Cov}(\mathbf{y}(\mathbf{X}, \mathbf{T})))$, which is a contradiction. □

References

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

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

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.

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.

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

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

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

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.

——— (2001), "Bayesian Calibration of Computer Models" (with discussion), *Journal of the Royal Statistical Society, Ser. B*, 63, 425-464.

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

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

Mitchell, T. J. (1974), "An Algorithm for the Construction of "D-Optimal" Experimental Designs", *Technometrics* 16, 203-210.

Niyama, E., Uchida, T., Morikawa, M., and Saito, S. (1982), "A Method of Shrinkage Prediction and its Application to Steel Casting Practice," *Int. Cast Met. J.* 7, 52-63.

Qian, P. Z. G. (2009), "Nested Latin Hypercube Designs," *Biometrika*, 96, 957-970.

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.

—— (2009), "Sliced Space-Filling Designs," *Biometrika*, 96, 945-956.

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

Rawlinson, J. J., Furman, B. D., Li, S., Wright, T. M., and Bartel, D. L. (2006), "Retrieval, Experimental, and Computational Assessment of the performance of Total Knee Replacements," *Journal of Orthopaedic Research*, 24, 1384-1394.

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

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

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer Verlag.

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

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

Run #	x(℃)	$t(mm)$	$y(cm^3)$	Status
1	710	2.5	189.67	training
2	739.05	2.5	178.16	training
3	724.92	2.5	184.05	training
4	750	2.5	175.42	training
5	715.74	2.36	149.94	training
6	746.18	2.36	137.79	training
7	733.15	2.30	153.91	training
8	723.80	2.20	160.19	training
9	710	2.16	197.92	training
10	743.39	2.12	190.15	training
11	715.15	2.03	208.49	training
12	730.82	1.96	137.17	training
13	750	1.93	149.2	training
14	718.57	1.85	196.04	training
15	710	1.74	195	training
16	743.07	1.73	175.31	training
17	723.33	1.59	161.76	training
18	710	1.5	172.94	training
19	735.25	1.5	165.85	training
20	750	1.5	159.53	training
21	725	1.5	167	testing

Table 2: Data, casting experiment.