THE UNIVERSITY OF BRITISH COLUMBIA

DEPARTMENT OF STATISTICS

TECHNICAL REPORT #238

Choosing the Sample Size of a Computer Experiment: A Practical Guide

Jason Leoppky, Jerome Sacks, William J. Welch

February 2008

Choosing the Sample Size of a Computer Experiment: A Practical Guide

Jason L. Loeppky
Mathematics, Statistics, and Physics
University of British Columbia, Okanagan
Kelowna, BC V1V 1V7, CANADA
(jason@stat.ubc.ca)

Jerome Sacks
National Institute of Statistical Sciences
Research Triangle Park, NC, 27709
(sacks@niss.org)

William J. Welch
Department of Statistics
University of British Columbia
Vancouver, BC V6T 1Z2, CANADA
(will@stat.ubc.ca)

February 19, 2008

Abstract

We produce reasons and evidence supporting the informal rule that the number of runs for an effective initial computer experiment should be about 10 times the input dimension. Our arguments quantify two key characteristics of computer codes that affect the sample size required for a desired level of accuracy when approximating the code via a Gaussian process (GP). The first characteristic is the total sensitivity of a code output variable to all input variables. The second corresponds to the way this total sensitivity is distributed across the input variables, specifically the possible presence of a few prominent input factors and many impotent ones (effect sparsity). Both measures relate directly to the correlation structure in the GP approximation of the code. In this way, the article moves towards a more formal treatment of sample size for a computer experiment. The evidence supporting these arguments stems primarily from a simulation study and via specific codes modeling climate and ligand activation of G-protein.

KEYWORDS: Computer experiment, Gaussian process, Random function, Latin hypercube design, Sample size.

1 Introduction

Choosing the sample size of any experiment is an important issue in the design of experiments, yet there is a lack of formal guidance. The reasons range from inadequate prior information about the process under study to inadequate results (and inability) for making necessary calculations. In standard regression settings, finding a sample size to

produce satisfactory predictive accuracy depends on the design points of the data collection and the error variance, but both the form of the regression model and the error variance are typically unknown a priori. Bayesian strategies can be deployed with some difficulty.

Deterministic computer experiments present a wholly different set of challenges, primarily because concepts such as randomization and replication play no role and predictive accuracy of the model is affected solely by bias. Because physical experimentation is absent, the constraints on experimental size are typically caused by the time it takes to make runs of the code. Such constraints are often vague and flexible. Where budget issues prevail ("you get this much computer time to make your runs") the choice of sample size, n, is taken out of our hands. Nevertheless, it is useful to have some practical guidance in choosing n and to know if the selected n is adequate to achieve stated goals.

In addition to guiding an experimenter in the choice of n for a specific experiment, we will consider more general questions. In particular, what is the role of dimensionality of the input space? If the curse of dimensionality applies, high-dimensional problems might require huge, even intractable, sample sizes for good prediction accuracy. On the other hand, if the total sensitivity of the function to all input variables is kept fixed, with this sensitivity just spread over more input variables, dimensionality might conceivably have a limited effect on accuracy, as in Monte Carlo integration. In this article, how total sensitivity grows with dimension and how this sensitivity is spread across the dimensions are key to understanding prediction accuracy, and hence sample size. Indeed, the article is really about defining the properties of functions that arise in practice, from which simple rules about sample size follow *for that class of problems*.

Little has been written on this topic. Among the few exceptions, Chapman et al. (1994) and Jones et al. (1998) used the often quoted rule of selecting a sample size that is 10 times the number of inputs. Although this rule has proved useful in practice it lacks theoretical underpinning. One theoretical exploration by Chen (1996) showed that, for a single varying input to the computer code whose output is under study, the order of the prediction error is n^{-n} for very smooth output functions and for an equally

spaced design. In higher dimensions, Chen (1996) produced results on rates for product designs. Though these rates are instructive, product designs are impractical and more precise understanding of prediction error is needed for choosing a sample size in practical settings. The key conclusion we arrive at is that the empirically based recommendation of $n = 10d$ is a good path to follow for a large class of problems.

An Example

Yi et al. (2005) studied a computer model of ligand activation of G-protein in yeast where the computer code takes four inputs and solves a system of ordinary differential equations (details are in Section 3). Following the path taken in the literature since 1989 (Sacks et al., 1989b; Currin et al., 1991), approximate the computer output using a Gaussian Process (GP) constructed from a set of code runs. The question that concerns us here is: How many runs are needed to obtain adequate prediction accuracy at untried test points? In the G-protein example, the code is relatively quick to run and we are able to investigate the effect of n on the prediction error by making runs for various values of n. For each value of n the code was evaluated at inputs from an n-point maximin Latin hypercube design (LHD) in 4 dimensions (McKay et al., 1979; Morris and Mitchell, 1995). The plot in Figure 1 shows the square root of the integrated mean squared error (RIMSE) for predictions using the GP model, for various choice of n. (RIMSE is computed for both a set of 120 hold-out points and by leave-one-out cross-validation.) The minor improvement in RIMSE for sample sizes greater than $40 = 10d$ is a feature of many problems.

In general, characterization of the factors affecting approximation accuracy, and hence sample size, requires precise formulation of the goals of the experiment. Such a formulation is often elusive, however. We restrict attention to the experimental objective of approximating the code on the basis of sample runs. Even here, the choice of measures of accuracy is open to subjective judgment. Those we use are given in (5) and (6) below. Issues such as optimization of a target criterion could bring other considerations, especially that of fully sequential experimentation.

We have obscured the role of the design of the location of inputs in this process. Con-

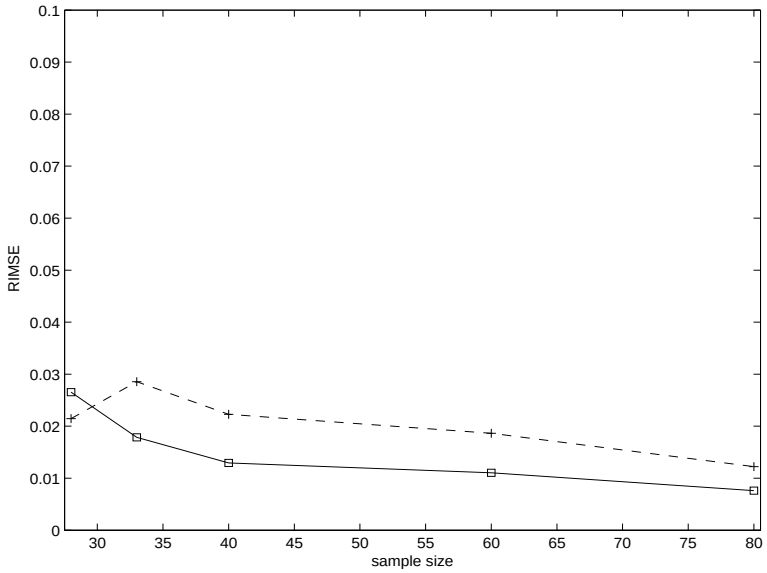

Figure 1: Root integrated mean squared error (RIMSE) of prediction against n for the G-protein example. The solid line shows RIMSE computed for a hold-out sample; the dashed line shows RIMSE from leave one out cross-validation.

siderable experience built up over a number of applications leads us to restrict attention to designs that are space-filling and, for the problems we address this is well managed by maximin LHDs (which we used in the G-protein example), but simpler to construct zero-correlation LHDs (Gough and Welch, 1994; Owen, 1994) could also be deployed.

While there are many issues that can be addressed in determining sample size we focus on these:

- Is $n = 10d$ a good rule? What are the limitations of such a rule?
- How does accuracy increase with n? When are feasible sample sizes available?
- What impact do criteria have on assessing accuracy?
- What should be done when a criterion for accuracy is not met?

We will partially answer these questions, enough to provide useful practical advice for the choice of n. Our approach to this problem investigates properties of the GP and the effect of n on prediction by first finding connections between the design and the complexity of

the problem and then conducting a simulation study. The simulations focus on deciding if $n = 10d$ is a reasonable rule and characterizing the complexity of problems that can be dealt with using $n = 10d$.

The paper is organized as follows. Section 2 reviews the GP model and gives specific formulations of the measures of accuracy we use. Section 3 explores the G-protein example in more detail. Section 4 investigates the relationships among dimension, sample size and complexity of the problem that guide the simulation study in Section 5 and 6. Section 7 discusses strategies for a follow-up experiment to augment an initial design and the implications for several examples. Finally, in Sections 8 and 9 we comment on open and future issues and summarize our conclusions.

2 The Gaussian Process Model

A complex computer code mathematically describes the relationship between several input variables and one or more (possibly functional) output variables. Usually, the computer model of interest is computationally demanding, and scientific objectives like optimization would require too many evaluations if the code is used directly. As a consequence, strategies relying on computationally efficient statistical approximation (emulation) of the code have been developed and have proved effective. Following the path taken in the literature since 1989 (Sacks et al., 1989a,b; Currin et al., 1991; O'Hagan, 1992), we place a homogeneous Gaussian process prior on the possible output functions, which leads to an approximator given by the posterior mean conditional on the data from the computer experiment. Although the output from the computer model is often multivariate, we will restrict our attention to scalar output. The results for scalar output can be carried over by using principal component analysis or wavelet decompositions of functional output as in Higdon et al. (2005) and Bayarri et al. (2007).

The computer code output is denoted by $y(\mathbf{x})$, where the code's vector-valued input, $\mathbf{x} = (x_1, \ldots, x_d)$, is assumed to be a point in a d-dimensional unit cube. As long as the input space is rectangular, there is no loss of generality here because any rectangle can

be transformed simply to the unit cube with only trivial implications for the analysis method to be described.

The GP model places a prior on the class of possible $y(\mathbf{x})$ functions. Let $Y(\mathbf{x})$ denote the random function whose distribution is determined by the prior. Specifically, we take

$$Y(\mathbf{x}) = \mu + Z(\mathbf{x}),$$

where μ is a mean parameter and $Z(\mathbf{x})$ is a Gaussian stochastic process with mean zero and constant variance σ^2. In this model, the correlation structure is crucial to prediction. At two input vectors, $\mathbf{x}$ and $\mathbf{x}'$, we take the correlation between $Y(\mathbf{x})$ and $Y(\mathbf{x}')$ as

$$R(\mathbf{x}, \mathbf{x}') = \exp\left(-h(\mathbf{x}, \mathbf{x}')\right), \tag{1}$$

where

$$h(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{d} \theta_j |x_j - x'_j|^{p_j}, \tag{2}$$

is a measure of distance between $\mathbf{x}$ and $\mathbf{x}'$ with weights $\theta_j \geq 0$ and distance-metric parameters $1 \leq p_j \leq 2$.

Experience in a variety of circumstances (Higdon et al., 2004; Linkletter et al., 2006) suggests that very smooth, even analytic, output is typical, especially in engineering contexts. As such it is often the case that p_j is fixed at 2 for all j, leading to the Gaussian correlation function. We adopt this special case for most of the article, but return to the issue of $p_j < 2$ in Sections 7 and 8. With $p_j = 2$, it is easily shown that

$$E\left|\frac{\partial Y(\mathbf{x})}{\partial x_j}\right|^2 = 2\sigma^2\theta_j.$$

Hence, the weight θ_j may be interpreted as a measure of the "sensitivity" of $Y(\mathbf{x})$ to x_j. Characterizing the distribution of the distances in (2) across design points as a function of the values of the sensitivity measures, $\theta_1, \ldots, \theta_d$, (Section 4) leads to an understanding of the the factors affecting prediction accuracy and hence sample size.

Suppose we make n runs of the code at a design D of input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ in $[0,1]^d$, leading to the data $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)}))^T$. The predictor $\hat{Y}(\mathbf{x})$ of $Y(\mathbf{x})$ is the

posterior mean of $Y(\mathbf{x})$ given the data and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$:

$$\hat{Y}(\mathbf{x}) = E\left(Y(\mathbf{x})|\mathbf{y}, \boldsymbol{\theta}\right) = \hat{\mu} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \hat{\mu}\mathbf{1}), \tag{3}$$

where $\mathbf{r}(\mathbf{x}) = (R(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, R(\mathbf{x}, \mathbf{x}^{(n)}))^T$ is an $n \times 1$ vector, $\mathbf{R}$ is an $n \times n$ matrix with element i, j given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and $\hat{\mu}$ is an estimate of μ, often from the method of maximum likelihood. The mean squared error (MSE) of $\hat{Y}(\mathbf{x})$, taking account of the uncertainty from estimating μ by maximum likelihood, is given by

$$\text{MSE}(\hat{Y}(\mathbf{x})) = E\left(\hat{Y}(\mathbf{x}) - Y(\mathbf{x})\right)^2 = \sigma^2\left(1 - \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + \frac{(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}))^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right), \tag{4}$$

where $\mathbf{1}$ is an $n \times 1$ vector with all elements equal to 1. In practice, σ^2 and $\boldsymbol{\theta}$ also have to be estimated, again often by maximum likelihood (Welch et al., 1992).

MSE in (4) can be directly computed given an experimental design and $\boldsymbol{\theta}$, and is used in Section 4 for theoretical arguments. However, for our empirical studies we take a different path to define prediction accuracy by using leave-one-out cross-validation(CV) (Currin et al., 1991; Chapman et al., 1994; Gough and Welch, 1994) as follows.

Given a design D with sample size n and code runs $\mathbf{y}$, denote the cross-validated prediction of $y(\mathbf{x}^{(i)})$ by $\hat{Y}_{-i}(\mathbf{x}^{(i)})$, which is the predictor (3) from the $n-1$ runs excluding run i. Then the cross-validated error of prediction is $\hat{Y}_{-i}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)})$ for $i = 1, \ldots, n$. Average and maximum measures of error based on cross-validation are given by

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_{-i}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)})\right)^2}$$

and

$$\max_{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}}\left|\hat{Y}_{-i}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)})\right|.$$

We also normalize for the scale of the function by dividing by the range of the values of y in the data, leading to the following inaccuracy summaries:

$$e_{\text{avg}} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_{-i}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)})\right)^2}}{\text{range of } y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})} \tag{5}$$

and

$$e_{\max} = \frac{\max_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}} \left| \hat{Y}_{-i}(\mathbf{x}^{(i)}) - y(\mathbf{x}^{(i)}) \right|}{\text{range of } y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})}. \tag{6}$$

The tolerable level of inaccuracy will be application-specific, but we will typically take $e_{\text{avg}} < 0.1$ as the target for a "useful" approximation of the code.

For questions relating to the sample size of an initial design, we do not have data available, at least not code data. But we can simulate data using the GP model with given $\boldsymbol{\theta}$. We will distinguish e depending on whether the data are from code runs or from simulations by $e_{\text{avg|code}}$ and e_{avg} respectively.

Before code runs are made, we can perform replicate simulations of the random function and obtain a collection of e_{avg} values, which will then provide an empirical distribution of (5). The average of the simulations will then be an estimate of expected inaccuracy as formulated in (5). Similarly we can get an estimate of expected accuracy as formulated in (6).

Why proceed with simulations rather than attempt direct computation of expected values, for example? There are four reasons:

1. The ratios in (5) and (6) are appealing measures of accuracy. Producing expected values or other quantities of these measures are hopeless without simulation; they are readily estimated via simulation.

2. As described in Section 7, after the sample size is selected and the computer experiment is run we can evaluate $e_{\cdot|code}$ and, with the information from the simulations, especially their empirical distribution, we can gauge whether the GP model and sample size are well matched to the actual code.

3. Even if we take expectation and remove all randomness in the data, there are other sources of randomness in practice. Most experimental designs are isomorphic with respect to various symmetries such as interchanging the columns. Different versions of the design within the equivalence class would lead to different measures of prediction error, even after taking expectation with respect to the data.

3 G-protein Computer Code

The ligand activation of G-protein in yeast is described by Yi et al. (2005). The computer code solves a system of ordinary differential equations (ODEs) with nine parameters that can vary. The system dynamics, the differential equations, are given by:

$$\begin{aligned}
\dot{\eta}_1 &= -u_1\eta_1 x + u_2\eta_2 - u_3\eta_1 + u_5 \\
\dot{\eta}_2 &= u_1\eta_1 x - u_2\eta_2 - u_4\eta_2 \\
\dot{\eta}_3 &= -u_6\eta_2\eta_3 + u_8(G_{\text{tot}} - \eta_3 - \eta_4)(G_{\text{tot}} - \eta_3) \\
\dot{\eta}_4 &= u_6\eta_2\eta_3 - u_7\eta_4 \\
y &= (G_{\text{tot}} - \eta_3)/G_{\text{tot}},
\end{aligned}$$

where $\eta_1, \ldots, \eta_4$ are concentrations of four chemical species, $\dot{\eta}_i \equiv \frac{\partial \eta_i}{\partial t}$; x is the concentration of the ligand; $u_1, \ldots, u_8$ is a vector of 8 kinetic parameters; G_{tot} is the (fixed) total concentration of G-protein complex after 30 seconds; and y is the normalized concentration of a relevant part of the complex. In one study (Feeley et al., 2007), five of these kinetic parameters are fixed (only allowing x, u_1, u_6, and u_7 to vary). The GP model is used to construct an approximation as a function of the transformed variables $\log(x), \log(u_1), \log(u_6), \log(u_7)$ each of which is further transformed to $[0, 1]$.

The ODE solver is quick to run and enables us to evaluate the affect of n on the criterion in (5) using a real model. The design points at which the code is run are selected by using maximin LHDs. These space-filling designs have proved to be highly effective in the study and application of computer experiments.

The values of n we use are multiples $(7, 10, 15, 20)$ of the dimension, 4, and also include 33, the number of runs made in the Feeley et al. (2007) study. For each choice of n, we run the ODE solver to obtain data $\{y(\mathbf{x}^{(i)}); \mathbf{x}^{(i)} \in D\}$. The data are modeled as if they were the realizations of a GP, and maximum likelihood estimates of $\hat{\mu}, \hat{\sigma}$ and $\hat{\boldsymbol{\theta}}$ are obtained for the parameters of the GP (see Section 2) for each choice of n. For each value of n, we use the code runs to calculate $e_{\text{avg}|\text{code}}$ from (5), except no normalization for the range is made here. (In any case, the normalization factor is close to 1 at about 0.8 and

makes little difference.) We also compare this measure with the analog from a set of new test points. We generate an additional independent 120-point maximin LHD, D_0, and evaluate the ODE solver to obtain data for the out-of-sample test points. The same 120 test runs are used for all evaluations. Using the test sample, the analogous version of (5) is computed by replacing the average in the numerator by $\frac{1}{120}\sum_{i=1}^{120}\left(\hat{Y}(\mathbf{x})-y(\mathbf{x})\right)^2$.

The plot in Figure 1 shows how the two unnormalized $e_{\text{avg|code}}$ measures behave as n changes. A major point is that $e_{\text{avg|code}}$ changes little as n increases past 40, nor is there any substantial difference between using cross validation instead of a new test sample. That cross validation leads to larger errors is not surprising, since leaving out one point can produce a big gap making it hard to predict the omitted point. This is relevant because the use of new test data is a luxury, only enjoyed if the code can be run quickly, and so we rely on cross validation for measuring accuracy.

Judging the quality of prediction over a wide range of scenarios is simply not possible through runs of the computer code unless the code is very quick to run. Therefore, we will rely on simulated data generated by using a GP. Because a GP model often has similar properties to those of the computer codes we expect to encounter in practice, we are at least close to mimicking reality. Before simulating, however, we need to know the important factors in function complexity, so that an efficient and insightful simulation study may be conducted.

4 Effect of d, $\boldsymbol{\theta}$, and n on prediction accuracy

Intuitively, we know that when we predict $Y(\mathbf{x})$ at some $\mathbf{x}$, the design-point neighbors of $\mathbf{x}$ will tend to be closer as n becomes larger, improving accuracy. If $\boldsymbol{\theta}$ has many large values, however, the correlation between $Y(\mathbf{x})$ and Y for the neighbors will be low, even for nearby points, leading to poorer prediction accuracy. Here, we develop this intuition into some quantitative rules relating d, $\boldsymbol{\theta}$, and n to distances and the correlation structure, shedding some light on how prediction accuracy depends on these quantities.

First, we consider how the theoretical mean squared error, $\text{MSE}(\hat{Y}(\mathbf{x}))$ in (4), depends

on d, $\boldsymbol{\theta}$, and n. Recall that the empirical definitions of prediction accuracy in (5) and (6) are normalized for scale. Similarly, without loss of generality, we can ignore σ^2 in a normalized version of mean squared prediction error:

$$\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x})) = 1 - \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + \frac{(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}))^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}. \tag{7}$$

We see that $\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x}))$ is determined by $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$ only. Thus, it is a function of n—as $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$ are an $n \times n$ matrix and an $n \times 1$ vector, respectively—and the correlations in $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$. Dimensionality, d, affects $\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x}))$ only indirectly via these correlations.

For simplicity, we will explore the factors affecting $\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x}))$ for completely random Latin hypercube designs (where the columns are permuted independently). We consider the case where n is fixed and look at the effect of d and $\theta_1, \ldots, \theta_d$ on the correlation structure and on $\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x}))$ averaged over $\mathbf{x}$. Our argument establishes two main results:

1. For moderately large d and a random LHD, the distribution of inter-point squared distance (weighted by $\theta_1, \ldots, \theta_d$) in (2) can be approximated by a normal distribution, with mean and variance given by simple functions of $\theta_1, \ldots, \theta_d$. The approximate distribution of (off-diagonal) correlations in $\mathbf{R}$ follows from the transformation in (1).

2. Under the same conditions, the distribution of correlations in $\mathbf{r}(\mathbf{x})$ for $\mathbf{x}$ drawn randomly from $[0, 1]^d$ is similar to the distribution of correlations in $\mathbf{R}$.

We recognize that the matrix inverse in (7) makes MSE_{norm} much more complicated than can be explained by these distributions of correlations. Nonetheless, the simulations in Section 5 and 6 show that the factors affecting the correlation distribution explain much of the effect of d and $\theta_1, \ldots, \theta_d$ on our empirical accuracy measures. Indeed, understanding these factors leads to simulation studies with straightforward interpretation.

Take two points, $\mathbf{x}$ and $\mathbf{x}'$, at random from a random LHD. An LHD is defined here

to have fixed grid points $\{0, 1/(n-1), \ldots, 1\}$ for each variable x_j. Let

$$h_j = |x_j - x'_j|$$

be the unweighted distance in dimension j appearing in h in (2). The first two moments of h_j^2 are given by Lemma 1.

Lemma 1: Let h_j be the distance between two randomly chosen points x_j and x'_j in dimension j for a random LHD. Then

$$P(h_j = i/(n-1)) = \frac{n-i}{\binom{n}{2}} \quad \text{for } i = 1, \ldots, n-1,$$

$$E(h_j^2) \equiv m_1(n) = \frac{1}{6}\frac{n(n+1)}{(n-1)^2},$$

and

$$\mathrm{Var}(h_j^2) \equiv m_2(n) = \frac{1}{180}\frac{n(n-2)(n+1)(7n+9)}{(n-1)^4}.$$

The proof of Lemma 1 can be found in Appendix A. Note that the two moments converge to 1/6 and 7/180 as $n \to \infty$.

If $d = 1$ than the probability distribution $P(h_j = i/(n-1))$ in Lemma 1 exactly describes the distribution of all possible distances between distinct points x and x'. That is, since the design points are on the grid, every possible distance $i/(n-1)$ occurs $n - i$ times.

If $d > 1$, however, not all of the possible distances over all dimensions will be observed in any one design, and we rely on the moments given in Lemma 1 to describe behavior. Specifically, for two randomly chosen points, the squared distance in (2) across all dimensions which arises if $p_j = 2$ has expectation

$$E(h) = m_1(n) \sum_{j=1}^{d} \theta_j. \tag{8}$$

For a completely random LHD, which has independently permuted columns,

$$\mathrm{Var}(h) = m_2(n) \sum_{j=1}^{d} \theta_j^2. \tag{9}$$

Furthermore, as d increases, the central limit theorem applies unless there are a few θ_j weights that dominate, so that h is approximately normal with mean and variance given by (8) and (9). Hence, the correlation in (1) is approximately log-normal with these moments (after a change of sign). Figure 2 compares the empirical distributions for a single random LHD with the approximations, for $d = 10$, $n = 100$, and $\boldsymbol{\theta} = (2.71, 2.17, 1.69, 1.27, 0.91, 0.61, 0.37, 0.19, 0.07, 0.01)$ the approximations are seen to be good. The values chosen for θ_j comprise a canonical configuration, to be explained in Section 5.

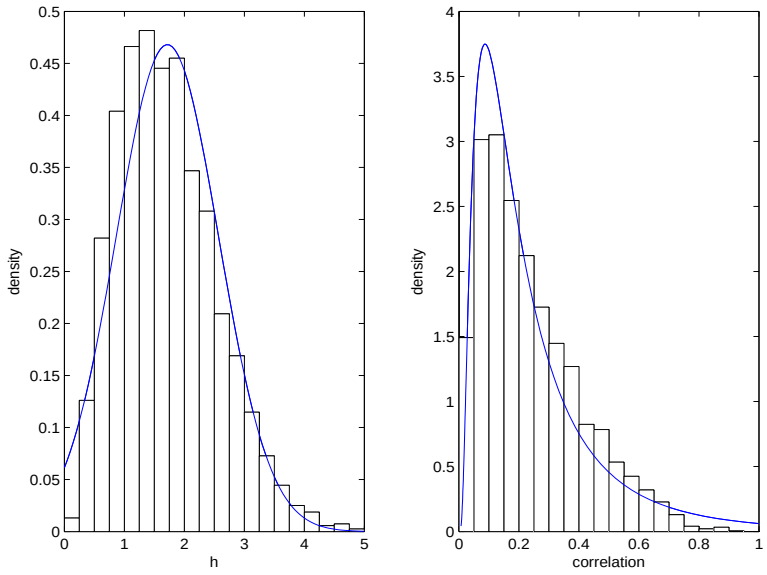

Figure 2: Distribution of squared distance (left panel) and correlation (right panel) for a randomly chosen pair of points from a random Latin hypercube design with $d = 10$, $n = 100$, and $\boldsymbol{\theta} = (2.71, 2.17, 1.69, 1.27, 0.91, 0.61, 0.37, 0.19, 0.07, 0.01)$

Similarly, Figure 3 shows the analogous distributions for the vector $\mathbf{r}(\mathbf{x})$. The empirical distribution of the distance and the correlation are taken over a single random LHD, D, and the distance between $\mathbf{x}$ and all n points in D is computed for 50 randomly chosen test point $\mathbf{x} \in [0, 1]^d$. The same normal or log-normal approximations established above for inter-point distance or correlation are transferred to test-point to design-point distance or correlation. It is seen that the correlations in $\mathbf{r}(\mathbf{x})$ behave like those in $\mathbf{R}$.

Note that there is a negative impact on prediction accuracy when the mean distance

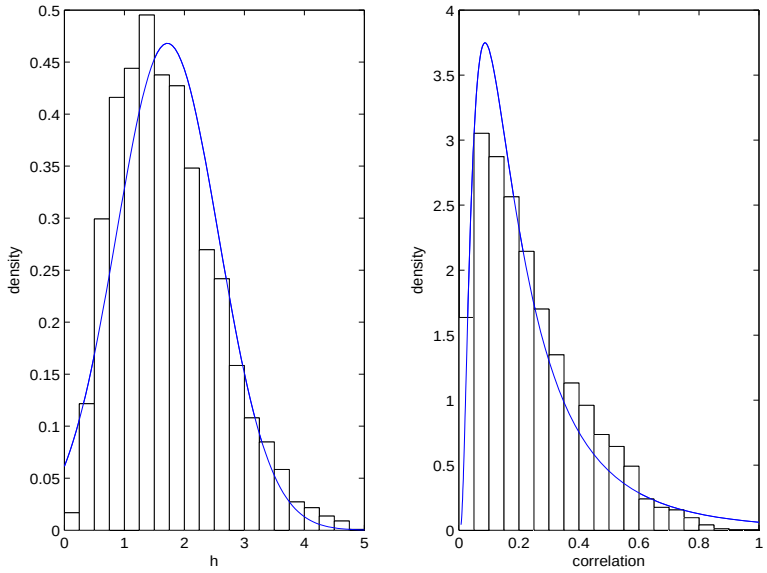

Figure 3: Distribution of squared distance (left panel) and correlation (right panel) for a randomly chosen test point and a random Latin hypercube design with $d = 10$, $n = 100$, and $\boldsymbol{\theta} = (2.71, 2.17, 1.69, 1.27, 0.91, 0.61, 0.37, 0.19, 0.07, 0.01)$

in (8) increases or when the variance in (9) *decreases*. When the variance decreases, small squared distances (high correlations) are less likely, whereas high-correlation neighboring points lead to good prediction accuracy. Indeed, if we want to predict $Y(\mathbf{x})$, just one close design neighbor, $\mathbf{x}^{(i)}$, of $\mathbf{x}$, close in the sense of correlation, may by itself give good prediction accuracy. Let $\rho = \exp(-h(\mathbf{x}, \mathbf{x}^{(i)}))$ be the correlation for these two points. Predicting using only $Y(\mathbf{x}^{(i)})$ provides an upper bound on MSE_{norm} obtained from (7). As $\mathbf{R}$ is the scalar 1, and we have

$$\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x})) < 1 - \rho^2 + (1 - \rho)^2 = 2(1 - \rho). \tag{10}$$

If ρ is larger than about 0.95, or equivalently h is less than about 0.05, $\text{MSE}_{\text{norm}}(\hat{Y}(\mathbf{x})) < 0.1$, the target we have set.

The magnitudes of the correlations in $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$, which lead to MSE_{norm} in (7), depend on $\tau = \sum_{j=1}^{d} \theta_j$ and $\psi = \sum_{j=1}^{d} \theta_j^2$ to a good approximation. There are two practical consequences. First, these two quantities are used to plan the simulations in Section 5 and 6. We find that the behavior of the empirical analog, e_{avg}, of MSE_{norm} is largely dependent on τ and ψ. Secondly, the distributions of the correlations in $\mathbf{r}(\mathbf{x})$

(for random test points) and in $\mathbf{R}$ (between design points) are similar. The implication is that accuracy estimates will be similar from cross validation based on leaving out one design point at a time versus random test points.

There are many possible $\theta_1, \ldots, \theta_d$ configurations, and we examine three special cases, ordered from worst to best behavior in terms of the effect of dimensionality.

1. Suppose $\theta_1 = \cdots = \theta_d = \theta$, i.e., as dimensionality increases, further equally active variables are added. Then, $\tau = d\theta$ and $\psi = d\theta^2$. Thus, the mean of the distribution of h increases linearly with d, the standard deviation of the distribution increases as $\sqrt{d}$, and the h distribution becomes stochastically larger with d. For large enough d, prediction accuracy will be poor, even if θ is small.

2. Suppose τ is kept constant, i.e., a fixed amount of total sensitivity is spread across all dimensions. Clearly, ψ takes its minimum value of $\psi = \tau^2/d$ when $\theta_1 = \cdots = \theta_d = \tau/d$. Thus, equally active factors are worst for prediction accuracy. Moreover, as $\psi = \tau^2/d$ decreases with d, this effect becomes worse as d increases. For large enough d, the h distribution will become concentrated at its mean, $m_1(n)\tau$, and the limiting accuracy depends on τ. In this sense, if the total amount of sensitivity is kept constant, the *worst-case* effect of dimensionality is small.

3. Alternatively, suppose we keep τ *and* ψ constant as d increases. Write

$$\psi = \sum_{j=1}^{d} \theta_j^2 = \sum_{j=1}^{d} (\theta_j - \bar{\theta})^2 + \frac{1}{d}\tau^2. \tag{11}$$

   Because the second term on the right decreases with d, $\sum_{j=1}^{d}(\theta_j - \bar{\theta})^2$ must *increase* to keep ψ constant. Another way of looking at the fact that $\theta_1, \ldots, \theta_d$ must become more variable with d to maintain prediction accuracy is that some dimensions are more active than others, or there is *effect sparsity*.

The argument that accuracy decreases as $\tau = \sum_{j=1}^{d} \theta_j$ increases or as $\psi = \sum_{j=1}^{d} \theta_j^2$ decreases is borne out by the simulations in Section 5 and 6.

The quantitative effect of n on accuracy is less obvious, however. The mean and variance of the squared distance distribution in (8) and (9) do not depend on n in the limit. Thus, $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$ in (7) have elements that depend only weakly on n in this statistical sense. Rather, MSE_{norm} depends on n because $\mathbf{R}$ and $\mathbf{r}(\mathbf{x})$ have *more* elements. The closest neighbor in the bound (10) will tend to become closer with larger n, thus driving the bound down. A full analysis of the impact of using all n design points is complicated by the inverse of $\mathbf{R}$ in (7). All that we can say is that harder problems (larger τ and smaller ψ) will require larger sample sizes, regardless of dimensionality to a large extent. This insight greatly facilitates quantification of the impact of n by simulation in Section 5 and 6.

If $p_1 = \cdots = p_d$, but the common value is less than 2, $m_1(n)$ and $m_2(n)$ in Lemma 1 will change. The mean and variance of h in (8) and in (9) will still depend on τ and ψ, however.

5 Simulation Results for Average Error

The arguments in Section 4 suggest that the effect of the correlation parameters on e_{avg} is through τ and ψ, thereby diminishing the role of d. To investigate this further we will engage a simulation study that changes dimension but keeps τ, ψ fixed.

But before doing so we must decide on the configurations of the $\boldsymbol{\theta}$ vectors to be explored. Past experience has indicated that for well-behaved outputs there may be a few large components of $\boldsymbol{\theta}$, a few moderately sized, and the remainder small. For example, for the G-protein model and the 33-run experiment, $\hat{\boldsymbol{\theta}} = (1.71, 0.29, 0.27, 0.25)$ has one moderate value and the other three are small. From this point of view, we will adopt a two-parameter class of *canonical configurations* of $\boldsymbol{\theta}$, defined by

$$\theta_j = \tau \left[\left(1 - \frac{j-1}{d}\right)^b - \left(1 - \frac{j}{d}\right)^b \right] \quad \text{for } j = 1, \ldots, d, \text{and } b \geq 1, \tau > 0. \tag{12}$$

Here θ_j decreases in j and $\sum_{j=1}^{d} \theta_j = \tau$. The generated $\boldsymbol{\theta}$ vector tends to have the characteristics we expect, especially as d gets large. Examples of $\boldsymbol{\theta}$ configurations for

$d = 10$ and $\tau = 1$ are given in Table 1. When $\tau \neq 1$, the value of $\boldsymbol{\theta}$ is found by multiplying each θ_j in the table by τ.

b	ψ	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
3	0.18	0.271	0.217	0.169	0.127	0.091	0.061	0.037	0.019	0.007	0.001
9	0.45	0.613	0.253	0.094	0.030	0.008	0.002	0	0	0	0

Table 1: Configurations of $\boldsymbol{\theta}$ for $d = 10$

Data for the simulation study are generated as follows. Given d and n, select a maximin LHD D of n points in $[0,1]^d$. Fix values of $\mu = 0, \sigma^2 = 1$, $\mathbf{p} = \mathbf{2}$ and select a canonical $\boldsymbol{\theta}$ (as specified above) for the parameters of the GP given in (2). Generate 50 independent realizations of the GP resulting in 50 different sets of observations $\{y(\mathbf{x}^{(i)}); \mathbf{x}^{(i)} \in D\}$. Since, the measure of accuracy in (5) or (6) is standardized by the range, the particular value of $\sigma^2 = 1$ is largely irrelevant.

For each data set form a predictor using (3) with the value of $\boldsymbol{\theta}$ the same as that used to generate the simulated data. Alternatively, for each data set, we could estimate $\boldsymbol{\theta}$ and construct a predictor with $\hat{\boldsymbol{\theta}}$. We found that there is no essential difference between predictors based on $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ in terms of our summary measures of prediction accuracy, and using the fixed $\boldsymbol{\theta}$ takes much less time in our extensive study. The predictor leads to a value of e_{avg} in (5) for each data set.

We start with $d = 5$ and $b = 1$ in (12), which results in $\theta_j = \tau/5, j = 1, \ldots, 5$. As argued in Section 4, this choice of $\boldsymbol{\theta}$ minimizes ψ for a fixed τ and represent a "worst case" starting point. For a given τ value, $\psi = \tau^2/5$ when $d = 5$. If τ and ψ are kept constant as d changes, then the canonical $\boldsymbol{\theta}$ vector must satisfy $\sum \theta_j^2 = \tau^2/5$. For $d = 10$, 15, and 20, this means that $b = 3.445$, 5.507, and 7.55, respectively, in (12). Values of $\tau = 3$, 10, 20, and 40 are chosen to cover problems from "easy" to "very hard".

The arguments in Section 4 suggest that similar accuracy should be obtained in any dimension for fixed values of n, τ and ψ, but intuitively we expect that n must increase

with d. Our results for the first two moments may not fully explain the behavior of the tails of the distribution of h, and small distances in particular play a prominent role. Thus, we allow n to increase modestly with d, specifically linearly. We also allow different rates, i.e., $n = kd$, where $k = 7$, 10, 15, or 20.

The four panels in Figure 4 correspond to $\tau = 3$, 10, 20, and 40, respectively. In each panel, four curves are plotted, for $d = 5$, 10, 15, and 20, respectively. A curve shows the mean of e_{avg} computed from the 50 realizations of the GP, which we denote by $\bar{e}_{\text{avg}}$, plotted against k (recall $n = kd$). Several features of these plots are worth singling out:

- All curves lie below 0.20 suggesting that even in very hard problems ($\tau = 40$) the average error does not get extremely large.
- The case of $\tau = 3$ represents an "easy" problem owing to the small components of $\boldsymbol{\theta}$.
- When ψ is fixed, the curves for $d = 5$, 10, 15, and 20 are all quite close.
- The choice of $n = 10d$ leads to predictions that on average are accurate to within 10% of the range of the data providing that $\tau \leq 10$; reliable fits are barely obtainable or not obtainable for $\tau \geq 20$.
- The improvement in fit for sample sizes greater than $n = 10d$ is marginal.

Suppose $\bar{e}_{\text{avg}}$ decreases with n approximately at the convergence rate n^{-c}. The rate c can be estimated from the points shown in Figure 4 from the slope of the least squares fit of $\log(\bar{e}_{\text{avg}})$ regressed on $\log(k)$. The estimated rates are in Table 2.

There are a few interesting things to notice in Table 2. For easy problems ($\tau = 3$) convergence rates close to 1 are achievable for dimensions as large as $d = 20$ so that doubling sample size can reduce e_{avg} by half. On the other hand, in hard problems the rates of convergence can be very small. For example, when $d = 15$ and $\tau = 20$, it takes about 8 times as many runs to reduce e_{avg} by half. When $\tau = 40$ it appears hopeless to reduce e_{avg} substantially without enormous sample sizes. In such situations, the computer experiment may have to be reformulated and restricted.

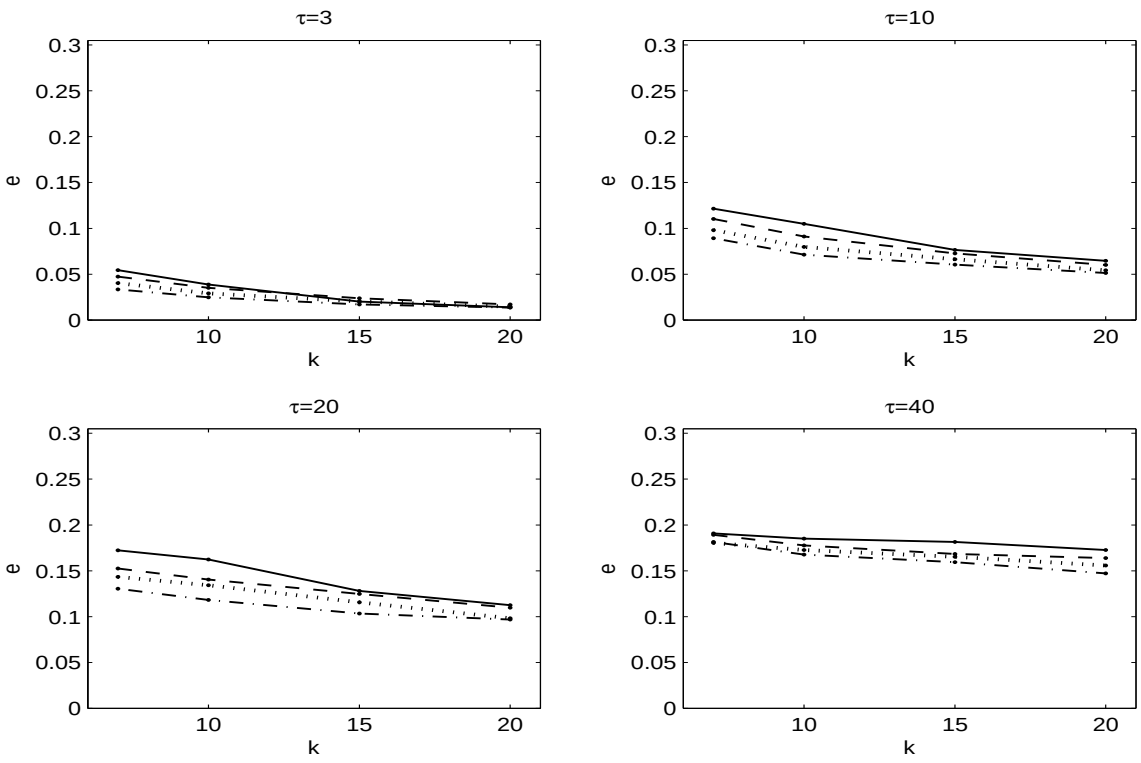

Figure 4: The four panels correspond to four values of τ, and $\psi = \tau^2/5$. In each panel, $\bar{e}_{\text{avg}}$ is plotted against k for $d = 5$ (solid line), $d = 10$ (dashed line), $d = 15$ (dotted line) and $d = 20$ (dot-dashed line).

	τ			
d	3	10	20	40
5	1.34	0.63	0.43	0.08
10	0.97	0.57	0.31	0.14
15	0.96	0.53	0.36	0.13
20	0.87	0.51	0.28	0.19

Table 2: Estimated convergence rates for $\bar{e}_{\text{avg}}$

The arguments in Section 4 suggest that for fixed total sensitivity τ, dividing τ equally across the d input variables is the worst case for prediction accuracy, i.e., $\psi = \tau^2/d$. Figure 5 explores worst-case problems by plotting e_{avg} against τ. There is a separate

plot for $d = 5, 10, 15, 20$, and $n = 10d$ throughout. Fifty simulated realizations are made for each value of τ. The lines in Figure 5 drawn through the averages of e_{avg} show

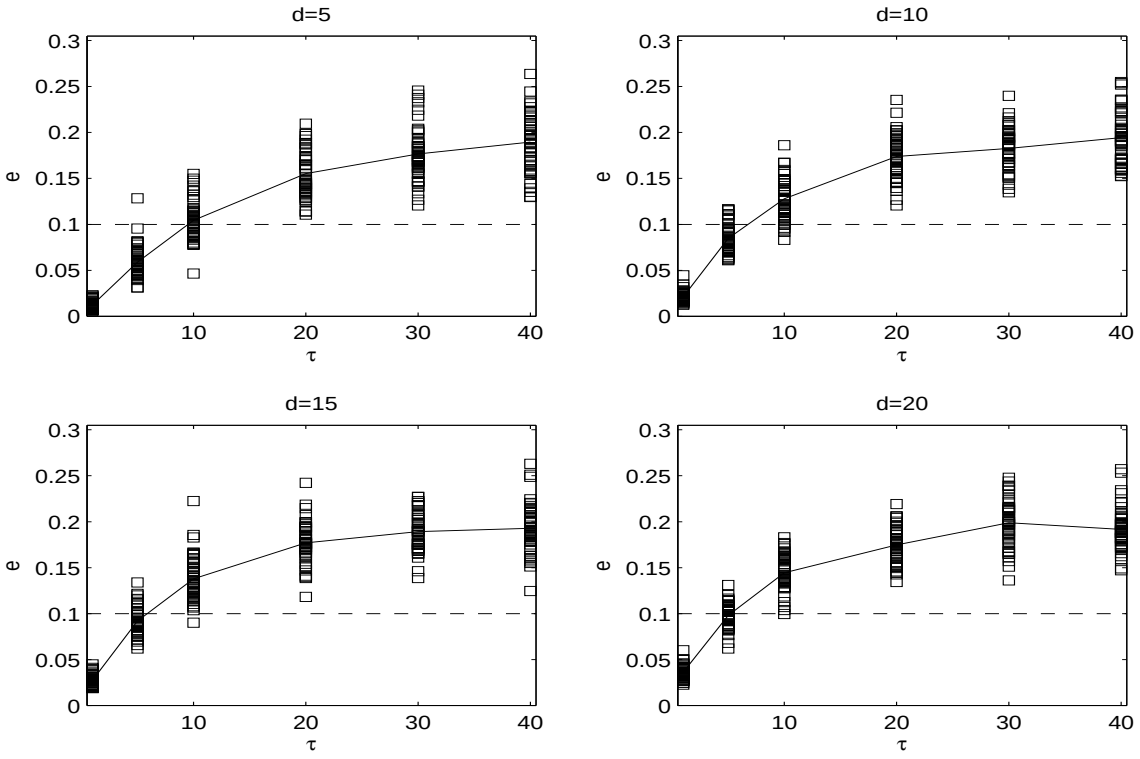

Figure 5: The four panels correspond to $d = 5$, 10, 15, and 20, respectively. In each panel, e_{avg} (squares) from 50 realizations and $\bar{e}_{\text{avg}}$ (solid line) are plotted against τ. The horizontal line indicates accuracy to within 10% of the range of the data.

little difference as d increases, as predicted in Section 4 for the worst case studied here. There is a small dimensionality effect (and $n = 10d$ is increasing with d), but the total sensitivity, τ, is the important factor. For $\tau \geq 20$, e_{avg} is above the target of 0.1 for all d studied, though it tends to remain below about 0.2 to 0.25. This latter fact is a somewhat surprising feature and, in fact, holds for τ as large as 100 (not shown).

To investigate the more realistic situation where the problem has some degree of sparsity we allow ψ to vary. We fix $d = 10$ and $n = 100$. As suggested by Figure 4, for fixed values of τ and ψ, results for other values of d (with $n = 10d$) are similar. For each fixed value of τ we increase the value of ψ so that the sparsity is increased, and the total sensitivity of the function is shifted to fewer and fewer dimensions.

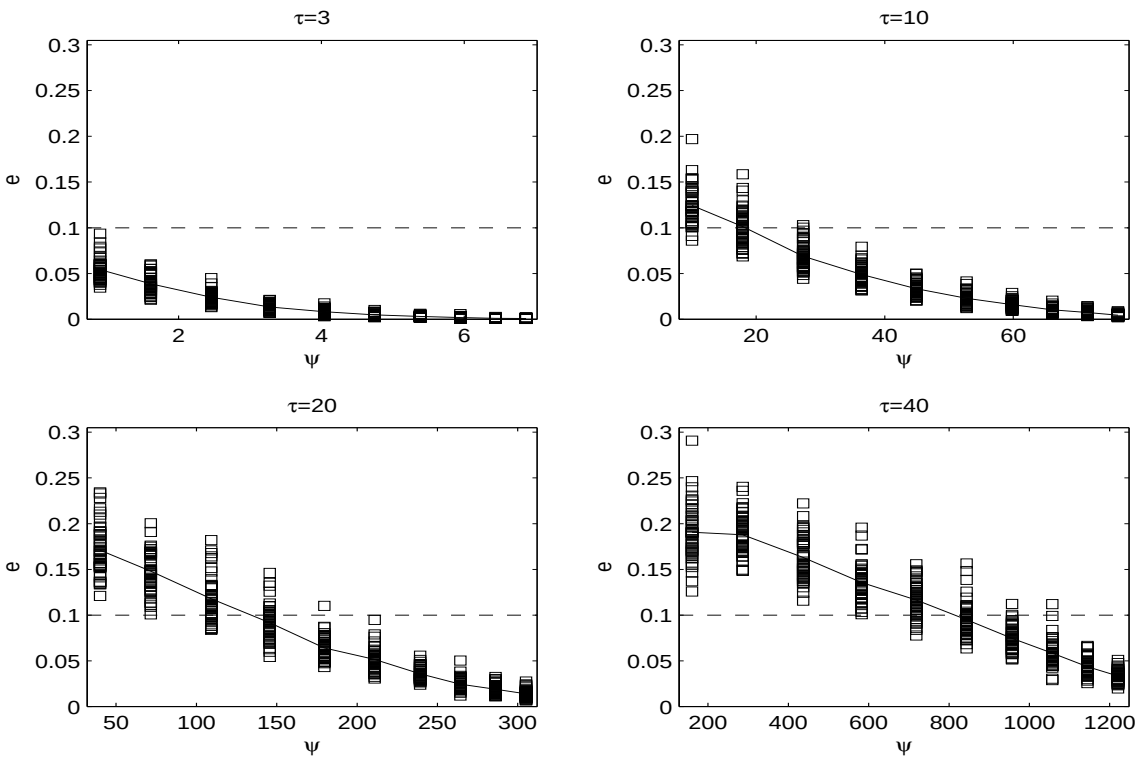

Figure 6: The four panels correspond to four values of τ. In each panel, e_{avg} (squares) from 50 realizations and $\bar{e}_{\text{avg}}$ (solid line) are plotted against ψ. The horizontal line indicates accuracy to within 10% of the range of the data.

Even a moderate degree of sparsity can result in drastic reduction of error. The $\tau = 40$ panel in Figure 6 is interesting, since even in such a complex problem reasonable accuracy can be obtained when there is a degree of sparsity. In particular the last few values of ψ represent situations where the 10-dimensional problem contains five or fewer active dimensions.

6 Simulation Results for Maximum Error

In Section 5 results were presented corresponding to $\bar{e}_{\text{avg}}$; in this section we discuss the similarities and differences that arise in using $\bar{e}_{\text{max}}$, i.e. the average of e_{max} in (6) across simulations. The four panels in Figure 7 correspond to $\tau = 3$, 10, 20, and 40, respectively. In each panel, curves for $\bar{e}_{\text{max}}$ (averaged across 50 simulations) are plotted for $d = 5$, 10,

15, and 20. Clearly, one would not expect to see the same level of accuracy as obtained using $\bar{e}_{\text{avg}}$ so we set a threshold of 0.2 as opposed to the threshold of 0.1 used for e_{avg}. Comparing this to the plots in Figure 4 there a few things to notice:

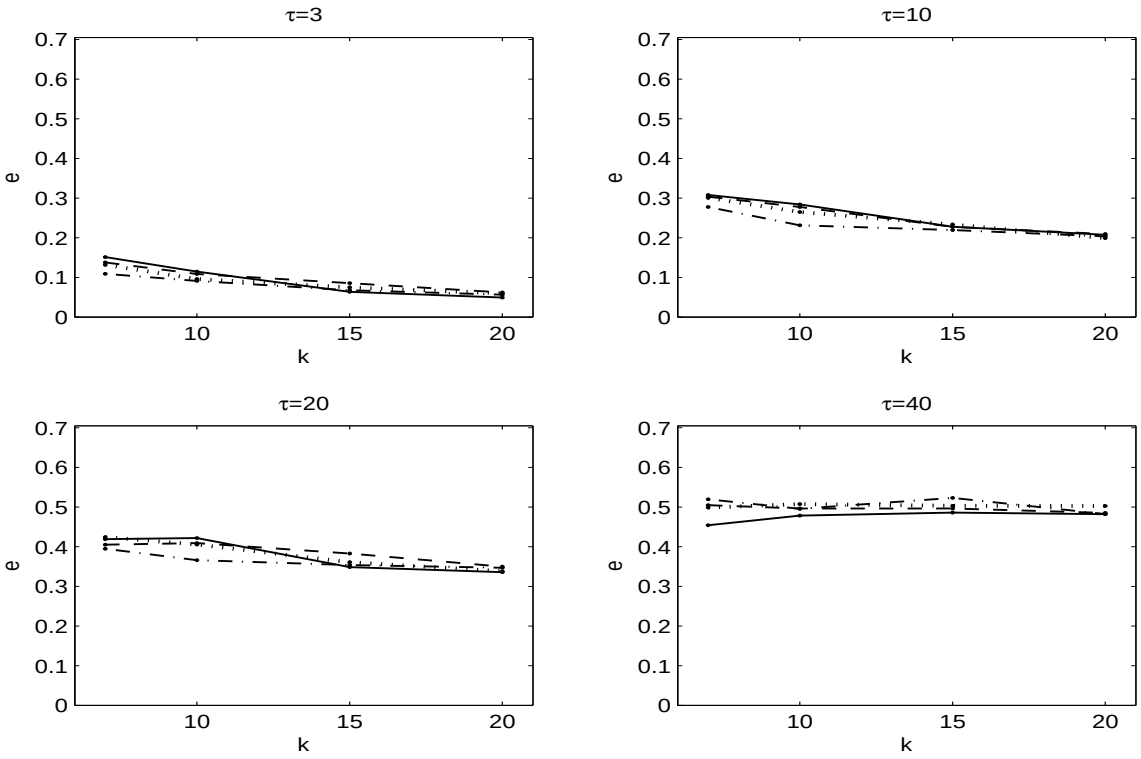

Figure 7: The four panels correspond to four values of τ and $\psi = \tau^2/5$. In each panel $\bar{e}_{\text{max}}$ is plotted against k for $d = 5$ (solid line), $d = 10$ (dashed line), $d = 15$ (dotted line) and $d = 20$ (dot-dashed line).

- The case of $\tau = 3$ again represents an "easy" problem owing to the small components of $\boldsymbol{\theta}$.
- When τ is fixed, the curves for $d = 5$, 10, 15, and 20 are all quite close.
- The choice of $n = 10d$ leads to predictions that on average are accurate to within 20% to 30% of the range of the data providing that $\tau \leq 10$; reliable fits are barely obtainable or not obtainable for $\tau \geq 20$. When $\tau = 10$ thresholds for e_{max} are somewhat harder to reach compared to those using e_{avg}.

- The improvement in fit for sample sizes greater than $n = 10d$ is marginal.

The analysis leading to Table 2 can be duplicated for $\bar{e}_{\max}$; the result is Table 3. The convergence rates, as expected, are lower than for $\bar{e}_{\text{avg}}$ and are essentially 0 when $\tau = 40$. Table 3 can be used in the same way as Table 2 is used in Section 7 to derive sample sizes needed to reduce $\bar{e}_{\max}$.

	τ			
d	3	10	20	40
5	1.11	0.41	0.26	0.00
10	0.74	0.37	0.14	0.03
15	0.72	0.37	0.22	0.00
20	0.64	0.28	0.11	0.04

Table 3: Estimated convergence rates for $\bar{e}_{\max}$

Figure 8 explores worst-case problems (equal θ_j) by plotting $e_{\max}$ against τ. There is a separate plot for $d = 5, 10, 15, 20$, and $n = 10d$ throughout. Fifty simulated realizations are made for each value of τ. The lines in Figure 8 drawn through the averages of $e_{\max}$ show little difference as d increases. Comparing this to the analogous plot in Figure 5 we see the same general trend using $e_{\max}$ as opposed to e_{avg}. There is again a small dimensionality effect, but the total sensitivity, τ, is the important factor. For $\tau \geq 20$, $e_{\max}$ is above the target of 0.2 for all d.

The Figure 9 shows the effect of sparsity when using $e_{\max}$ and corresponds to the plot in Figure 6 for e_{avg}. Even a moderate degree of sparsity can result in drastic reduction of error. The $\tau = 40$ panel in Figure 9 is interesting, since even in such a complex problem reasonable accuracy can be obtained when there is a degree of sparsity. In particular the last few values of ψ represent situations where the 10-dimensional problem contains five or fewer active dimensions. This is similar to what was found in Figure 6.

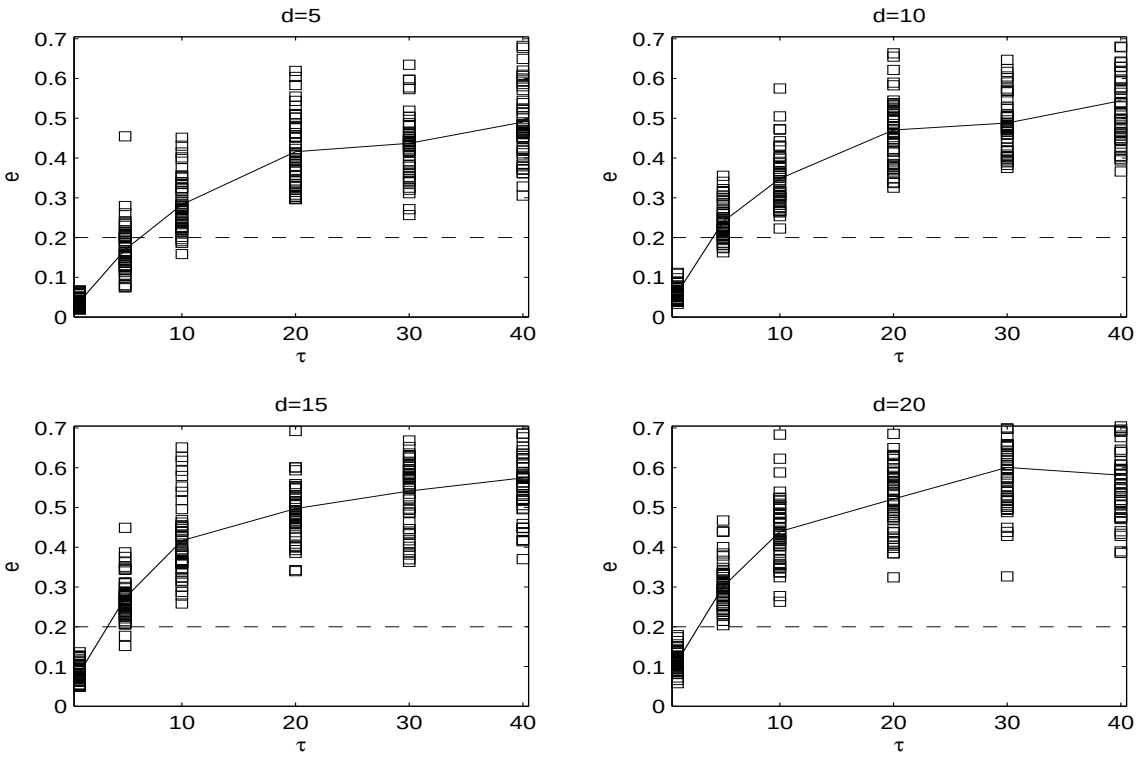

Figure 8: The four panels correspond to $d = 5$, 10, 15, and 20, respectively. In each panel, $e_{\max}$ (squares) from 50 realizations and $\bar{e}_{\max}$ (solid line) are plotted against τ. The horizontal line indicates accuracy to within 20% of the range of the data.

7 Follow-up Experiments

Suppose an initial experiment of given sample size has been conducted. We now have real data from running the code to fit a GP model following the methodology described in Section 2. Estimates of the correlation parameters, $\boldsymbol{\theta}$ and $\mathbf{p}$, are available, as well as values of $e_{\text{avg}|\text{code}}$ and $e_{\max|\text{code}}$ computed from (5) and (6).

What should be done to augment the initial design, if anything? Set e_A as a threshold value for acceptable e_{avg} (for example, $e_A = 0.1$). If $e_{\text{avg}|\text{code}} < e_A$ then nothing more needs to be done to increase accuracy. If $e_{\text{avg}|\text{code}} > e_A$ then we propose the following follow-up strategy:

1. Do a simulation study using the estimated correlation parameters, $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{p}}$, and the initial sample size. Compute e_{avg} (and $e_{\max}$) for each realized data set.

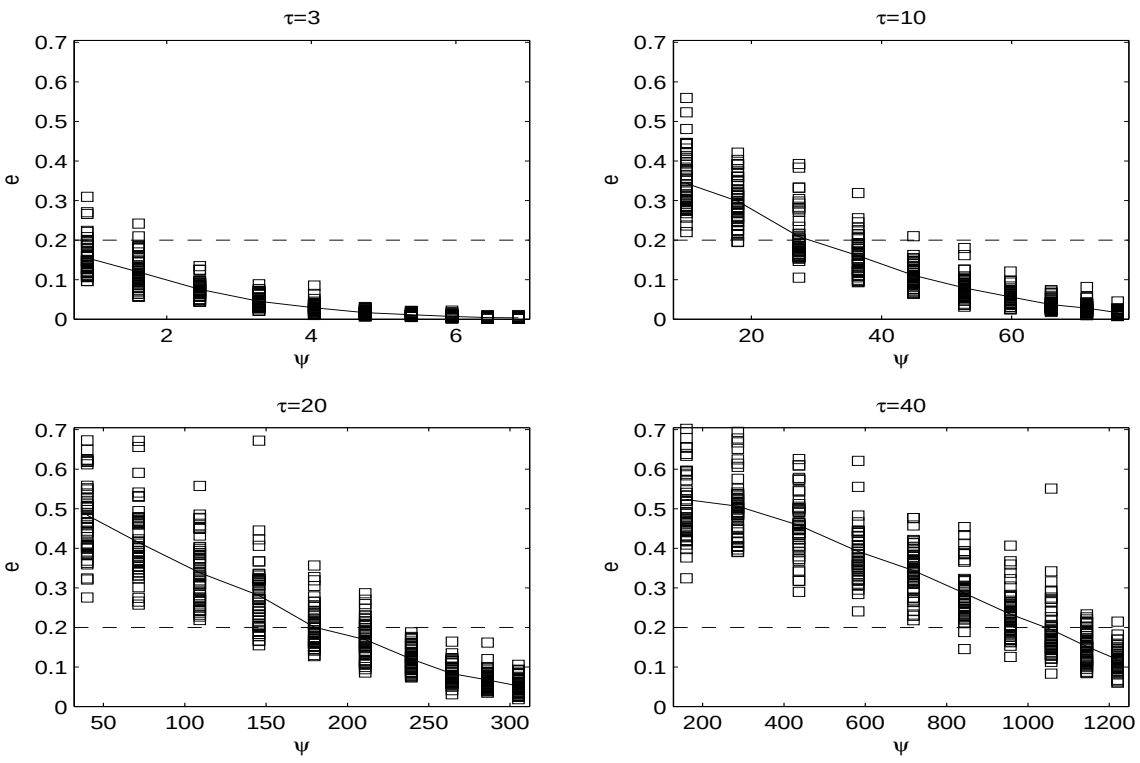

Figure 9: The four panels correspond to four values of τ. In each panel, $e_{\max}$ (squares) from 50 realizations and $\bar{e}_{\max}$ (solid line) are plotted against ψ. The horizontal line indicates accuracy to within 20% of the range of the data.

2. If the distribution of the e_{avg} values from the simulations suggests a non-trivial probability of exceeding e_A, it is plausible that the initial sample size is inadequate and we could go to Step 4. Otherwise, continue with Step 3.

3. If the distribution of the e_{avg} values from the simulations is inconsistent with $e_{\text{avg|code}}$, the appropriateness of the GP model is in question; see Section 8. Of course, it would be foolish to worry about small inconsistencies, and some subjectivity is inevitable in assessing what is substantial.

4. To decide on a follow-up sample size, explore plausible choices of n through a simulation study using the estimated correlation parameters, $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{p}}$.

In order to illustrate the points above, we revisit the G-protein example in Section 3. The initial sample size of $n = 33$ led to $e_{\text{avg|code}} = 0.0221$ (see Figure 1). For most

examples this would be considered small, and there would be nothing further to do. On the other hand, if we wanted to reduce the error rate by a half, a simulation could be performed using the outline above. Since $\hat{\mathbf{p}} = \mathbf{2}$, we could also consult Table 2 to calculate the desired sample size. From the analysis of the $n = 33$ runs, $\hat{\tau} = 2.52$ and $\hat{\psi} = 3.15$. For $d = 5$ and $\tau = 3$, Table 2 gives a rate of convergence of 1.34, which is very likely to be conservative. Thus, a sample size of at least $n = 33(2^{1/1.34}) = 55$ is required. From Figure 1, $e_{\text{avg|code}}$ at $n = 60$ is 0.0137, which is just slightly larger than half of the observed value when $n = 33$.

Gough and Welch (1994) considered a model for ocean circulation with $d = 7$ inputs and seven different outputs, $y_1, \ldots, y_7$. From an initial experiment yielding 36 good runs, a GP was fit separately for each output. In each case $\mathbf{p} = \mathbf{2}$ for each output $y_1, \ldots, y_7$. Estimated values of $\boldsymbol{\theta}$ are provided in Table 4. The $\hat{\tau}$ and $\hat{\psi}$ rows in Table 4 summarize

	y_1	y_2	y_3	y_4	y_5	y_6	y_7
$\hat{\theta}_1$	0.319	0.544	7.899	0.301	0.100	1.249	2.512
$\hat{\theta}_2$	0	0.026	0.001	0.552	0.827	0.001	0.003
$\hat{\theta}_3$	0.686	0.012	0.012	0	0.115	0.066	0.037
$\hat{\theta}_4$	0.267	1.202	0.021	0	0.156	0.478	0
$\hat{\theta}_5$	0.029	0.197	0.003	0.060	0.044	0.119	1.463
$\hat{\theta}_6$	0	0.008	0.006	1.136	0.685	0.433	0.665
$\hat{\theta}_7$	0.229	0.031	0.129	0.069	0.009	0.082	1.001
$\hat{\tau}$	1.53	2.02	8.07	2.12	1.94	2.43	5.68
$\hat{\psi}$	0.69	1.78	62.42	1.69	1.21	2.01	9.89

Table 4: Estimates of $\boldsymbol{\theta}$ for the ocean-circulation model.

the GP fits from the 36 runs.

If $e_A = 0.1$, the values for $e_{\text{avg|code}}$ in Table 5 show that a reasonable approximation has been obtained. Moreover, for all output variables, $e_{\text{avg|code}}$ is within $\bar{e}_{\text{avg}} \pm 2\widehat{\text{sd}}(e_{\text{avg}})$, i.e., $e_{\text{avg|code}}$ lies within the support of the empirical distribution of simulated e_{avg}. The

n		y_1	y_2	y_3	y_4	y_5	y_6	y_7
36	$e_{\text{avg}\|\text{code}}$	0.034	0.041	0.017	0.037	0.034	0.039	0.078
	$\bar{e}_{\text{avg}}$	0.028	0.033	0.022	0.030	0.042	0.051	0.072
	$\widehat{\text{sd}}(e_{\text{avg}})$	0.008	0.011	0.008	0.009	0.011	0.026	0.021
70	$\bar{e}_{\text{avg}}$	0.008	0.011	0.007	0.009	0.018	0.021	0.032
	$\widehat{\text{sd}}(e_{\text{avg}})$	0.004	0.003	0.003	0.007	0.006	0.006	0.009

Table 5: Actual and simulated accuracy measures for the ocean-circulation model

accuracy for y_7 is lower than for the other output variables, however, and we consider reducing $e_{\text{avg}|\text{code}}$ from 0.078 to, say, 0.05. In this case $\hat{\mathbf{p}} = \mathbf{2}$, and again we can use Table 2 for guidance in choosing an appropriate sample size.

Interpolating the convergence rates in the table suggest that $\bar{e}_{\text{avg}}$ should be decreasing at rate roughly $1/n$, and reducing the sample size to 0.05 would require a sample size of approximately 56 runs. Alternatively, cutting $\bar{e}_{\text{avg}}$ by half would require doubling the run size. As no further code runs are available we investigate this strategy by simulating what would have happened if $n = 70$ runs had been performed: for y_7, $\bar{e}_{\text{avg}}$ is reduced by just over a factor of 2, as predicted.

Chapman et al. (1994) analyzed a computer code describing the seasonal growth and decline of Arctic sea ice. The code had $d = 13$ input variables and four outputs, $y_1, \ldots, y_4$. From an initial design of $n_1 = 69$ runs, GPs were fit separately for each output. Every fitted GP had at least one input variables with $\hat{p_j} < 2$. Estimated values of θ_j and $\alpha_j = 2 - p_j$ for $n_1 = 69$ runs are provided in Table 6.

Estimated vales of θ_j and $\alpha_j = 2 - p_j$ for $n = 157$ runs are provided in Table 7.

The values of $e_{\text{avg}|\text{code}}$ are in Table 8; each is below 0.1, and if $e_A = 0.1$ we would be tempted to stop.

The $e_{\text{avg}|\text{code}}$ values for y_3 and y_4 are below 0.1 and within $2\widehat{\text{sd}}(e_{\text{avg}})$ of $\bar{e}_{\text{avg}}$ but the $e_{\text{max}|\text{code}}$ values of about 0.5 are of concern. The simulated e_{max} distributions are inconsistent with the observed $e_{\text{max}|\text{code}}$ values, as evidenced by $\bar{e}_{\text{max}}$ and $\widehat{\text{sd}}(e_{\text{max}})$ in Table 8,

	y_1		y_2		y_3		y_4	
j	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$
1	0.010	0.000	0.023	0.000	0.021	0.000	0.045	0.000
2	0.069	0.000	0.011	0.000	0.000	0.344	0.000	1.000
3	0.455	0.000	0.000	0.000	0.533	0.000	0.445	0.000
4	0.020	0.000	0.021	0.000	0.215	0.000	0.266	0.000
5	0.001	0.000	0.126	0.000	0.000	1.000	0.386	0.000
6	0.024	0.000	0.105	0.000	0.000	0.000	0.053	0.038
7	0.168	0.000	0.030	0.000	1.883	0.000	0.872	0.000
8	0.301	0.000	2.171	0.000	0.252	0.094	0.181	0.316
9	0.058	0.000	0.359	0.000	0.001	0.000	0.028	0.000
10	0.064	0.000	0.000	0.000	0.094	0.864	0.554	0.000
11	0.015	1.000	0.386	0.000	0.990	0.513	1.182	0.188
12	0.000	0.907	0.003	0.000	0.000	0.344	0.011	0.000
13	0.000	1.000	0.416	0.135	0.000	0.000	0.000	0.000

Table 6: Estimates of θ_j and $\alpha_j = 2 - p_j$ for the sea-ice code using $n_1 = 69$ runs.

although for y_3 the range of the simulated $e_{\max}$ values covers $e_{\max|\text{code}}$. The sea-ice code failed to converge for 12 of 81 attempted runs (hence the 69 good runs), a suggestion of erratic behavior of the code and a possible explanation of the difference between actual and simulated error in some regions of the input space.

Faced by similar concerns about the approximation accuracy from the initial experiment, Chapman et al. (1994) opted to make additional runs and ended up with a total of 157 good code runs. As these are the only follow-up runs available, we restrict our analysis to seeing whether we can predict by simulation the impact of such a follow-up experiment.

The accuracy measures for the $n = 157$ runs conducted and from simulation are

	y_1		y_2		y_3		y_4	
j	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$	$\hat{\theta}_j$	$\hat{\alpha}_j$
1	0.032	0.313	0.030	0.132	0.048	0.579	0.314	0.000
2	0.015	0.115	0.022	0.013	0.003	1.000	0.067	0.000
3	0.421	0.000	0.014	0.158	0.324	0.546	0.316	0.503
4	0.007	0.000	0.027	0.375	0.001	0.000	0.000	0.000
5	0.008	0.100	0.033	0.594	0.000	0.820	0.220	0.000
6	0.043	0.000	0.182	0.000	0.017	0.351	0.289	0.000
7	0.216	0.000	0.013	0.000	2.272	0.010	1.711	0.015
8	0.563	0.000	1.273	0.141	0.214	0.322	0.476	0.085
9	0.093	0.000	0.347	0.000	0.000	0.000	0.060	0.000
10	0.182	0.322	0.006	0.536	0.403	0.652	0.331	0.255
11	0.184	0.059	0.023	0.000	0.908	0.307	0.357	0.525
12	0.000	0.907	0.000	0.907	0.000	0.907	0.000	0.344
13	0.003	1.000	0.002	0.817	0.000	0.903	0.000	0.344

Table 7: Estimates of θ_j and $\alpha_j = 2 - p_j$ for the sea-ice code using $n = 157$ runs.

compared in Table 8. Relative to $n = 69$, simulation suggests only modest reduction in e_{avg}. For y_4, even this modest reduction is not realized by $e_{\text{avg|code}}$. With $n = 157$ runs, the simulated values of $e_{\max}$ are again inconsistent with $e_{\max|\text{code}}$ for the troublesome y_3 and y_4. Although the magnitude of the maximum error is underestimated, the simulations correctly predict that there will be little impact on $e_{\max|\text{code}}$ from the further runs. Thus, the simulation study leads to the same conclusion that Chapman et al. (1994) reached after the follow-up experiment: Taking more runs is not effective. Alternative ways of proceeding are discussed in Section 8.

n		y_1	y_2	y_3	y_4
		Average error			
69	$e_{\text{avg}\|\text{code}}$	0.043	0.044	0.093	0.099
	$\bar{e}_{\text{avg}}$	0.048	0.044	0.079	0.089
	$\widehat{\text{sd}}(e_{\text{avg}})$	0.011	0.013	0.019	0.018
157	$e_{\text{avg}\|\text{code}}$	0.032	0.031	0.079	0.096
	$\bar{e}_{\text{avg}}$	0.029	0.029	0.056	0.062
	$\widehat{\text{sd}}(e_{\text{avg}})$	0.008	0.009	0.011	0.011
		Maximum Error			
69	$e_{\text{max}\|\text{code}}$	0.249	0.124	0.466	0.559
	$\bar{e}_{\text{max}}$	0.139	0.128	0.225	0.263
	$\widehat{\text{sd}}(e_{\text{max}})$	0.039	0.052	0.071	0.079
157	$e_{\text{max}\|\text{code}}$	0.189	0.116	0.446	0.494
	$\bar{e}_{\text{max}}$	0.103	0.096	0.182	0.203
	$\widehat{\text{sd}}(e_{\text{max}})$	0.035	0.033	0.045	0.055

Table 8: Actual and simulated accuracy measures for the sea-ice code

8 Comments and Open Issues

There are several open issues, concerned mainly with follow-up once an initial set of code runs has been collected and analyzed.

Effective dimensionality

The ocean-circulation model (Gough and Welch, 1994) had an initial sample size of $n = 36$, about half the recommended value of $n = 10d$. Even so, a good fit was obtained. A closer look at this application shows that $\hat{\boldsymbol{\theta}}$ has elements that are near zero for three of the input variables. Thus, the input space is effectively reduced to $d = 4$ dimensions, leading to a recommendation of $n = 40$. If there are good a priori reasons to expect that

the number of active dimensions, d_0, is less than d then choosing $n = 10d_0$ could be a useful complement to the recommended strategy, especially if there are serious budget constraints.

The GP model is a poor fit

Good general strategies to cope with lack of fit of the GP model are not readily available. There is interesting work by Gramacy and Lee (2007) which could be useful when runs are plentiful. The approach used extensively by Aslett et al. (1998) and by Gramacy and Lee (2007), of narrowing the space of inputs, better enables approximation of code output by a homogeneous GP; the assumption of homogeneity is less sustainable when the input space is too large. But how to do this in a measured way is not clear and needs further research.

Canonical configurations of $\boldsymbol{\theta}$

For $\mathbf{p = 2}$, we chose a simple two-parameter family in our analyses in Section 5 and 6. Other sets of values for $\boldsymbol{\theta}$ can be explored, but we find little incentive to do so for the purpose of settling on initial sample size. We have found that even if $\boldsymbol{\theta}$ is not a canonical configuration there is little to no difference in distributions of e_{avg} or $e_{\max}$ relative to a canonical $\boldsymbol{\theta}$ provided τ and ψ are the same.

Treating a GP with $\mathbf{p \neq 2}$ (as in the sea-ice example)

We have not discussed the relevance, nor the use, of τ and ψ when $\mathbf{p \neq 2}$. The interpretation of τ and ψ values need to be reexamined.

In the case of the exponential correlation function (all $p_j = 1$), the implied prior distribution is on a much larger class of functions and achieving good accuracy is more difficult. It is easy to work out the mean and variance of h_j^1 as in Lemma 1, and again we find that τ and ψ should be important. The exact values are given in Lemma 2.

Lemma 2: Let h_j be the distance between two randomly chosen points for variable x_j in a random LHD. Then

$$E(h_j) = \frac{1}{3}\frac{(n+1)}{(n-1)},$$

and

$$\mathrm{Var}(h_j) = \frac{1}{18}\frac{(n-2)(n+1)}{(n-1)^2}.$$

The proof of Lemma 2 can be found in Appendix A. Note that the two moments converge to 1/3 and 1/18 as $n \to \infty$, i.e., they do not depend on n in the limit. The mean of h_j^1, is now approximately twice that for the case $p_j = 2$, indicating that larger samples could be needed to achieve desired accuracy. How this all plays out in analogues of the analyses in Section 5 and 6, to enable follow-up recommendations has yet to be explored.

When $1 < p_j < 2$, exact calculations of the mean and variance of $h_j^{p_j}$ are not available. Approximations are obtainable as follows, however. Assume that x_j and x_j' are approximately independent and uniform on $[0, 1]$, and again let $h_j = |x_j - x_j'|$. We find that $h_j^{p_j}$ has

$$E(h_j^{p_j}) = \frac{2}{(p_j+1)(p_j+2)}$$

and

$$E(h_j^{2p_j}) = \frac{1}{(p_j+1)(2p_j+1)}.$$

For $p_j = 2$ this produces $E(h_j^2) = 1/6$ and $\mathrm{Var}(h_j^2) = 7/180$, which are the asymptotic values found in Lemma 1.

For the general case, with p_j varying with x_j, assume the design is a completely random LHD. Asymptotically, $h(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^d \theta_j |x_j - x_j'|^{p_j}$ in (2) has mean depending on both $\boldsymbol{\theta}$ and $\mathbf{p}$:

$$E(h) = \sum_{j=1}^d \theta_j \frac{2}{(p_j+1)(p_j+2)}.$$

Similarly, it has asymptotic variance

$$\mathrm{Var}(h) = \sum_{j=1}^d \theta_j^2 \left(\frac{1}{(p_j+1)(2p_j+1)} - \frac{4}{(p_j+1)^2(p_j+2)^2} \right).$$

Defining canonical sets of correlation parameters is now much more complicated. Some preliminary calculations for the sea-ice application suggest that the convergence rates for $\mathbf{p} \neq \mathbf{2}$ are different from those obtained when $\mathbf{p} = \mathbf{2}$ and thus one must examine rates for various combinations of both $\boldsymbol{\theta}$ and $\mathbf{p}$. This too calls for additional examination.

9 Discussion

In the introduction we raised a set of issues that should be treated. In the subsequent sections we have provided evidence that:

- "$n = 10d$" is a viable and valuable rule-of-thumb for choosing an initial sample size for a computer experiment.
- Criteria can make a difference for post-experimental analysis but have less influence on initial sample size. The sea-ice example shows that the conflict between the e_{avg} and $e_{\max}$ criteria has implications, as spelled out in Section 7. However, as seen in Section 6 both criteria support the "$n = 10d$" rule.
- When $\mathbf{p = 2}$ there is good information about rates at which error decreases with n and when feasible sample sizes are available. These depend on the parameters τ and ψ, whose values are not known until the post-experimental stage and are then useful for deciding how to follow-up. When $\mathbf{p \neq 2}$ much remains to be done.
- In the case that accuracy goals are not met with an initial sample size, a follow-up strategy is needed, but a full analysis is lacking and is a topic for further inquiry.

APPENDIX A: Proof of Lemma 1

Proof of Lemma 1:

Let D be an $n \times d$ random LHD, and let x_j and x'_j be any two randomly chosen runs of the design in dimension j. The construction of the LHD ensures that $x_j \neq x'_j$, and hence x_j and x'_j are dependent random variables. There are a total of $\binom{n}{2}$ possible pairs of points and each pair is equally likely. Clearly, $P(x_j = i/(n-1)) = 1/n$ and $P(x'_j = k/(n-1)|x_j = i/(n-1)) = 1/(n-1)$. Consider any two points that are an absolute distance of $i/(n-1)$ apart. By a simple counting argument, there are $n - i$

pairs giving rise to this distance. This establishes

$$P(h_j = i/(n-1)) = \frac{(n-i)}{\binom{n}{2}} = \frac{2(n-i)}{n(n-1)}, \quad i = 1, \ldots, n-1.$$

The expected value of h_j^2 is

$$\begin{aligned} E(h_j^2) &= E\left(\frac{i^2}{(n-1)^2}\right) = \frac{1}{(n-1)^2}\left(\frac{2}{n(n-1)}\sum_{i=1}^{n-1} i^2(n-i)\right) \\ &= \frac{2}{n(n-1)^3}\left(n\sum_{i=1}^{n-1} i^2 - \sum_{i=1}^{n-1} i^3\right) = \frac{1}{6}\frac{n(n+1)}{(n-1)^2}. \end{aligned}$$

Similarly,

$$\begin{aligned} \mathrm{Var}(h_j^2) &= E(h_j^4) - E(h_j^2)^2 = E\left(\frac{i^4}{(n-1)^4}\right) - \left(\frac{1}{6}\frac{n(n+1)}{(n-1)^2}\right)^2 \\ &= \frac{1}{(n-1)^4}\left(\frac{2}{n(n-1)}\sum_{i=1}^{n-1} i^4(n-i)\right) - \left(\frac{1}{6}\frac{n(n+1)}{(n-1)^2}\right)^2 \\ &= \frac{1}{180}\frac{n(n-2)(n+1)(7n+9)}{(n-1)^4}. \end{aligned}$$

Algebra was carried out in Maple. □

Proof of Lemma 2:

Following Lemma 1, the expected value is:

$$\begin{aligned} E(h_j) &= E\left(\frac{i}{(n-1)}\right) = \frac{1}{(n-1)}\left(\frac{2}{n(n-1)}\sum_{i=1}^{n-1} i(n-i)\right) \\ &= \frac{2}{n(n-1)^2}\left(n\sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} i^2\right) = \frac{1}{3}\frac{(n+1)}{(n-1)} \end{aligned}$$

Similarly, the variance is:

$$\begin{aligned} \mathrm{Var}(h_j^2) &= E(h_j^2) - E(h_j)^2 = \frac{1}{6}\frac{n(n+1)}{(n-1)^2} - \left(\frac{1}{3}\frac{(n+1)}{(n-1)}\right)^2 \\ &= \frac{1}{18}\frac{(n-2)(n+1)}{(n-1)^2}. \end{aligned}$$

Algebra was carried out in Maple. □

ACKNOWLEDGEMENTS

The research of Loeppky and Welch was supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J., and Welch, W. J. (1998), "Circuit Optimization via Sequential Computer Experiments: Design of an Ouput Buffer," *Applied Statistics*, 47, 31–48.

Bayarri, M. J., Berger, J. O., Garcia-Donato, G., Sacks, J., Walsh, D., Cafeo, J., and Parthasarathy, R. (2007), "Computer Model Validation with Function Output," *Annals of Statistics*, 35, 1874–1906.

Chapman, W. L., Welch, W. J., Bowman, K. P., Sacks, J., and Walsh, J. E. (1994), "Arctic sea ice variability: Model sensitivities and a multidecadal simulation," *Journal of Geophysical Research*, 99, 919–936.

Chen, X. (1996), "Properties of Models for Computer Experiments," Ph.D. thesis, University of Waterloo.

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.

Feeley, R., Frenklach, M., Paulo, R., and Sacks, J. (2007), "A Study of the G-protein Computer Model," Tech. rep., Unpublished.

Gough, W. A. and Welch, W. J. (1994), "Parameter Space Exploration of an Ocean General Circulation Model Using an Isopycnal Mixing Parameterization," *Journal of Marine Research*, 52, 773–796.

Gramacy, R. B. and Lee, H. K. H. (2007), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *Journal of the American Statistical Association*, to appear.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulation for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Higdon, D., Williams, R., Moore, L., McKay, M., and Keller-McNulty, S. (2005), "Uncertainty Quantification for Combining Experimental Data and Computer Simulations," in *Society for Modeling and Simulation International*, eds. Pace, D. and Stevenson, S.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998), "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13, 455–492.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006), "Variable Selection for Gaussian Process Models in Computer Experiments," *Technometrics*, 48, 478–490.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Ouput from a Computer Code," *Technometrics*, 21, 239–245.

Morris, M. D. and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402.

O'Hagan, A. (1992), "Some Bayesian Numerical Analysis," in *Bayesian Statistics 4*, eds. Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., Oxford University Press, pp. 345–363.

Owen, A. B. (1994), "Controlling Correlations in Latin Hypercube Samples," *Journal of the American Statistical Association*, 89, 1517–1522.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989a), "Designs for Computer Experiments," *Technometrics*, 31, 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989b), "Designs and Analysis of Computer Experiments (with Discussion)," *Statistical Science*, 4, 409–435.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.

Yi, T.-M., Fazel, M., Liu, X., Otitoju, T., Papachristodoulou, A., Prajna, S., and Doyle., J. (2005), "Application of Robust Model Validation Using SOSTOOLS to the Study of G-Protein Signaling in Yeast," in *Proceedings of Foundations of Systems Biology and Engineering*.