Advances in the Design of Gaussian Processes as Surrogate Models for Computer Experiments

J. Andrés Christen[1] and Bruno Sansó[2]

(1) CIMAT, Guanajuato, Mexico

(2) Department of Applied Mathematics and Statistics UCSC, USA.

Abstract

We present some advances in the design of computer experiments. A Gaussian Process (GP) model is fitted to the computer experiment data as a surrogate model. We investigate using the Active Learning (AL) strategy of finding design points that maximize reduction on predictive variance. Using a series of approximations based on standard results from linear algebra (Weyl's inequalities) we establish a score that approximates the AL utility. Our method is illustrated with a simulated example as well as with an intermediate climate computer model. The second example requires the calibration of a model that depends on three parameters that summarize important climate properties. The calibration of the model is done using one observation derived from historical records and model runs on a grid of 426 different parameter combinations. We use our score to revisit this design and also explore the possibility of weighting the score to create more adequate designs in terms of the calibration problem at hand.

KEYWORDS: Computer Experiments; Gaussian Processes; Surrogate Models; Sequential Design; Active Learning.

1 Introduction

Computer experiments have been in use since the dawn of digital computers and may be traced back to the Manhatan project in the 1940's (Feynman, 1985, Chapter 3). They are an increasingly popular method to study complex systems for which direct experimentation is either too costly, too time consuming or simply impossible. Over 400 hits are obtained by a popular scientific search engine with the phrase "computer experiments" (for years 2002-2007 only). Computer experiments

involve mathematical models that mimic reality with varying levels of accuracy and complexity. Some require very substantial computing resources involving the use of large supercomputers. In many cases though, recent advances in computer technology has made it possible for a wide group of researchers to tackle increasingly complex problems with low cost computers.

Computer models usually depend on a number of parameters that may or may not have physical relevance and need to be tuned or calibrated. Performing the computer experiment entails choosing the right combinations of parameter values in an experimental design setting. This task becomes critical when large computing resources need to be allocated for each run. An overview of the field of design and anlysis of computer experiments is presented in Santner et al. (2003). Bursztyn and Steinberg (2006) present a comprehensive discussion of design criteria in the context of computer experiments. They compare space filling designs, namely latin hypercube, lattice and rotation designs using various variance and entropy reduction criteria. The authors tend to favor the Integrated Mean Squared Error (IMSE) criterion. This is used to analyze and present algorithmically simple space filling design (eg. latin hypercube). They also propose a related but computationally simpler alias matrix "A" criterion. Latin hypercube designs are also favored by other authors. For example, Lehman, Santner and Notz (2004) elaborate on a design presented by Williams, Santner and Notz (2000) who use a latin hypercube design as initial trial. In a second stage, the posterior expected improvement is used to generate a desing that maximizes a desired criteria (eg. maximize some aspect of the computer output). Stinstra, Den Hertog, Stehouwer and Vestjens (2003) present algorithms for obtaining maximin designs in a computer experiment context (see also Mease and Bingham, 2006, Fang and Li, 2006, Cioppa and Lucas, 2007, and references therein).

When designing a computer experiment one wishes to "spread" the inputs so as to learn about the model for a wide diversity of parameter configurations (fill the space). A somewhat competing goal is that of focusing on the areas of the parameter space that correspond to high output variability. This approach corresponds to criteria based designs, see Santner, Williams and Notz (2003, Chapter 6) for a review. Once an information criterion is selected, the design proceeds in a

sequential setting in which points are chosen by learning from the output obtained by previously selected parameter values. The criterion proposed in Cohon (1996) consists of maximizing the average reduction in (predictive) variance at every point of a grid when adding a new design point. This is referred to as Active Learning and arises in robotics (we call it Active Learning Cohon or ALC, following Gramacy, 2005). In a Bayesian setting IMSE is equal to the integrated predictive variance (see Bursztyn and Steinberg, 2006) which in turn may be approximated by the average predictive variance over a grid, which would be equivalent to maximizing the ALC criterion. Therefore, IMSE, or equivalently ALC, seem to be favored by various independent sources.

Applying the ALC sequentially requires the calculation of the change in predictive variance every time a new point is added to the design. Such calculations may be extremely computationally demanding when a fine grid of parameter values is used. If a Gaussian process (GP) is used to approximate the computer output, an increasingly large covariance matrix needs to be inverted for every new point considered. Gramacy and Lee (2007) preselect points from a fine grid using a large latin hypecube design and calculate ALC designs on this subset of points. The idea is to use the latin hypercube to learn broadly about the parameter space and then use ALC to focus on higher uncertainty regions. In this paper we concentrate on making an approximation to the ALC criterion, that can be easily calculated over large grids. Our proposed design strategy can be easily updated sequentially. It tends to fill the space when small numbers of parameter combinations are available but will favor higher uncertainty regions as additional data become available. Additionally, it may be easily and effectively modified to account for specific features of the computer experiment, like obtaining a design for maximum output or, as in our case study, calibration of the parameters in the presence of substantive prior knowledge.

In Section 2 we present the general Gaussian process setting including calibration with one observation. In Section 3 we present the design problem and explain our score and sequential scheme. In Sections 4 and 5 we present a simulated 2D example and a design for an intermediate climate computer model, respectively, to show the performance of our design method. Finally, in Section 6 we present a discussion of this article.

2 GP as surrogate Models and Calibration

We assume we have a computer model that produces output $z(\boldsymbol{x}) \in \mathbb{R}$ for $\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^q$. We assume that evaluating the computer model, and perhaps post processing the output, is very costly. Therefore, we fit a GP model to $z(\cdot)$ as a surrogate model. Given m configurations of the parameters, $\boldsymbol{x}_i, i = 1, \ldots, m$,

$$z(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)\boldsymbol{\beta} + \epsilon_i,$$

with $\epsilon_i \sim N(0, \sigma^2)$ (σ^2 is the normal variance), where $f(\boldsymbol{x}) \in \mathbb{R}^q$ is some regressor function and $\beta \in \mathbb{R}^q$ a set of linear parameters. The idea is to work with the surrogate model and minimize the number of evaluations of the actual computer model $z(\cdot)$.

Letting $\boldsymbol{z} = (z(\boldsymbol{x}_1), z(\boldsymbol{x}_2), \ldots, z(\boldsymbol{x}_m))'$, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)'$ and $\boldsymbol{F} = (f(\boldsymbol{x}_i))$ we have:

$$\boldsymbol{z} = \boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

with $\boldsymbol{\epsilon} \sim N_m(\boldsymbol{0}, \sigma^2 \boldsymbol{R}_{\boldsymbol{\lambda}})$, where $\boldsymbol{R}_{\boldsymbol{\lambda}}$ is some correlation matrix arising from the covariance function $\text{cov}(y(\boldsymbol{x}_i), y(\boldsymbol{x}_j)) = c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. A common assumption is to make the covariance dependent on a low dimensional parameter, say $\boldsymbol{\lambda}$, so that $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 K_{\boldsymbol{\lambda}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This is usually achieved by imposing some restrictions on c, like stationarity or isotropy.

2.1 Calibration

We refer to calibration as the problem of inferring the combinations of parameter values that produce computer output close to avaible observations. Thus given an observation $Z \in \mathbb{R}$ we need to infer the value of $\boldsymbol{\theta} \in \mathcal{D}$ such that $z(\boldsymbol{\theta}) \approx Z$. We assume that the computer model has some inadequacies with respect to actual observations, therefore

$$Z = z(\boldsymbol{\theta}) + \delta,$$

where $\delta \sim N(0, \tau\sigma)$. This is easily generalizable to having more than one observation, but for the sake of clarity we will concentrate on having only one.

The corresponding model, as proposed in Kennedy and O'Hagan (2001) is

$$\begin{bmatrix} Z \\ \boldsymbol{z} \end{bmatrix} \sim N_m \left(\begin{bmatrix} f(\boldsymbol{\theta}) \\ \boldsymbol{F} \end{bmatrix} \boldsymbol{\beta}, \sigma^2 \begin{bmatrix} 1+\tau^2 & r_{\boldsymbol{\lambda}}(\boldsymbol{\theta})' \\ r_{\boldsymbol{\lambda}}(\boldsymbol{\theta}) & \boldsymbol{R}_{\boldsymbol{\lambda}} \end{bmatrix} \right),$$

where $r_{\boldsymbol{\lambda}}(\boldsymbol{\theta})' = (K_{\boldsymbol{\lambda}}(\boldsymbol{\theta}, \boldsymbol{t}_1), K_{\boldsymbol{\lambda}}(\boldsymbol{\theta}, \boldsymbol{t}_2), \ldots, K_{\boldsymbol{\lambda}}(\boldsymbol{\theta}, \boldsymbol{t}_m))$. Such model allows for joint estimation of the computer model parameters (θ and τ) and the GP parameters ($\sigma^2, \boldsymbol{\lambda}$ and $\boldsymbol{\beta}$).

One way to proceed would be to establish the posterior distribution of $(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}, \tau, \boldsymbol{\theta})$ and sample from it (using MCMC for example). However we do not expect to be able to learn much for the parameter τ. So a typical strategy is to fix it and perform a sensitivity analysis of the results.

Note that

$$Z|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}, \tau^2 \quad \sim N(\mu(\boldsymbol{\theta}), S(\boldsymbol{\theta})^2),$$

where $\mu(\boldsymbol{\theta}) = f(\boldsymbol{\theta})\boldsymbol{\beta} + r_{\boldsymbol{\lambda}}(\boldsymbol{\theta})'\boldsymbol{R}_{\boldsymbol{\lambda}}(Z - \boldsymbol{F}\boldsymbol{\beta})$ and $S(\boldsymbol{\theta})^2 = \sigma^2(1 + \tau^2 + r_{\boldsymbol{\lambda}}(\boldsymbol{\theta})'\boldsymbol{R}_{\boldsymbol{\lambda}} r_{\boldsymbol{\lambda}}(\boldsymbol{\theta}))$. Therefore

$$f(\boldsymbol{\theta}|Z, \sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}, \tau^2) \propto \frac{\pi(\boldsymbol{\theta})}{S(\boldsymbol{\theta})} \exp\left\{ \frac{(Z - \mu(\boldsymbol{\theta}))^2}{2S(\boldsymbol{\theta})^2} \right\} \tag{1}$$

where $\pi(\boldsymbol{\theta})$ is the prior distribution for $\boldsymbol{\theta}$. Moreover, fixing τ^2 we have that

$$f(\boldsymbol{\theta}|Z, \boldsymbol{z}) = \int f(\boldsymbol{\theta}|Z, \sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}, \tau^2) f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|Z, \boldsymbol{z}) \; d\sigma^2 \; d\boldsymbol{\lambda} \; d\boldsymbol{\beta}.$$

Therefore, if a sample from the posterior of the GP parameters, $f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|Z, \boldsymbol{z})$, is available, we can produce samples from $f(\boldsymbol{\theta}|Z, \boldsymbol{z})$ using Equation (1). One way to proceed is to resample from the prior $\pi(\boldsymbol{\theta})$ using sampling importance resampling (see Bernardo and Smith, 1994, p. 350). This consists of obtaining samples $\boldsymbol{\theta}^{(j)} \sim \pi(\boldsymbol{\theta}); j = 1, 2, \ldots, h$ and $(\sigma^2)^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\beta}^{(t)} \sim f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|Z, \boldsymbol{z}); t = 1, 2, \ldots, M$, and then resample $\boldsymbol{\theta}^{(j)}$ with probability

$$q_j = \frac{\sum_{t=1}^{M} f(\boldsymbol{\theta}_j|Z, (\sigma^2)^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\beta}^{(t)}, \tau^2)}{\sum_{t=1}^{M} \sum_{l=1}^{h} f(\boldsymbol{\theta}_l|Z, (\sigma^2)^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\beta}^{(t)}, \tau^2)}.$$

We expect that the contribution to the posterior of $(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta})$ by the single observation Z (and the prior of $\boldsymbol{\theta}$) will be marginal, in comparison the other available m data points in $\boldsymbol{z}$. Then we assume that

$$f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|Z, \boldsymbol{z}) = f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|\boldsymbol{z}). \tag{2}$$

The last simplification splits the inference problem in two. First sampling from $f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|\boldsymbol{z})$, which are the GP parameters, and second using those samples to simulate from $f(\boldsymbol{\theta}|Z, \boldsymbol{z})$. The simplification in (2) can only be acceptable when very few calibration points are available. As in the example presented in Section 5, having one or very few actual observations is typical of computer model calibration problems.

To sample from $f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|\boldsymbol{z})$ we observe that

$$\boldsymbol{\beta}|\sigma^2, \boldsymbol{\lambda}, \boldsymbol{z} \sim N_m(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}, \sigma^2(\boldsymbol{F}'\boldsymbol{R}_{\boldsymbol{\lambda}}^{-1}\boldsymbol{F})),$$

where $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ is the standard generalized least squares estimator for $\boldsymbol{\beta}$. This provides the full conditional for $\boldsymbol{\beta}$. However, the full conditional for $\boldsymbol{\lambda}$ does not have a close form and is normally quite difficult to sample from. Thus, to sample from $f(\sigma^2, \boldsymbol{\lambda}, \boldsymbol{\beta}|\boldsymbol{z})$ we propose the use of the t-walk, (Christen and Fox, 2008), a self-adjusting MCMC that is designed to cope with highly correlated posteriors adapting to various scales in several dimensions. In the examples considered in this paper, such strategy has produce very good results.

3 Designing the experiment

3.1 Active Learning

The criterion proposed in Cohn (1996) amounts to a utility function that is given as the average decrease in predictive variance of a GP, for each of the points in a designated grid, if the candidate point is included in the design. Let $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ be points on a grid; design points are to be taken from this grid. Let $\boldsymbol{D}_N = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ be points were the computer model has been evaluated. The $N+1$st point is chosen by maximizing

$$ALC(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{1}{m}\sum_{j=1}^{m} V(\boldsymbol{y}_j|\boldsymbol{D}_N) - V(\boldsymbol{y}_j|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}),$$

were $V(\boldsymbol{y}|\boldsymbol{D}_N)$ is the predictive variance of the GP at $\boldsymbol{y}$ and $V(\boldsymbol{y}_j|\boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ is the expected predictive variance at $\boldsymbol{y}$. The maximization is taken over all possible additional points $\boldsymbol{x}_{N+1}$. As explained in Section 1, in a Bayesian setting, ALC is approximately equivalent (over the grid points)

to minimizing the Integrated (predictive) Mean Square Error. This in turn is equivalent, under certain circumstances, to other design criteria (see Bursztyn and Steinberg, 2006, and references therein). After a new design point is found, ideally its output is calculated from the computer experiment and new predictive variances are obtained to search for the next design point. ALC is thus intrinsically sequential. It is reported to produce good designs but, unfortunately, for a GP, evaluating $ALC(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$ for every point in a large grid (eg. 500, 10,000 points etc.) is simply not feasible. Thus strategies to reduce the search, like the one implemented in Gramacy and Lee (2007), need to be considered.

3.2 An alternative to ALC

We present a score that represents a proxy for ALC, based on an upper bound for the reduction in predictive variance (see the Appendix for a formal argument). We propose it as a simpler alternative to ALC. The score is defined as

$$A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{\frac{1}{m}\sum_{j=1}^{m} c(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2 + \frac{1}{m} C^1_{\boldsymbol{D}_N}}{C^2_{\boldsymbol{D}_N} + \sqrt{\sum_{i=1}^{N} c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2}}. \tag{3}$$

where $C^1_{\boldsymbol{D}_N} = \sum_{j=1}^{m}\sum_{i=1}^{N} c(\boldsymbol{y}_j, \boldsymbol{x}_i)^2$ and $C^2_{\boldsymbol{D}_N} = \max_j \sum_{i=1}^{N} |c(\boldsymbol{x}_i, \boldsymbol{x}_j)|$. For a heuristic justification, we note that

$$A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{\frac{1}{m} c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1})^2 + \frac{1}{m}\sum_{\boldsymbol{y}\neq\boldsymbol{x}_{N+1}} c(\boldsymbol{y}, \boldsymbol{x}_{N+1})^2 + \frac{1}{m} C^1_{\boldsymbol{D}_N}}{C^2_{\boldsymbol{D}_N} + \sqrt{\sum_{i=1}^{N} c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2}}.$$

Therefore, the score suggests to choose $\boldsymbol{x}_{N+1}$ with high predictive variance, correlated with other points in the grid (which might be a very good idea, since by choosing $\boldsymbol{x}_{N+1}$ we could be learning about points correlated with it), but not so "close" to already selected points, given the denominator term. Note that the score has the same units as σ^2, as it is the case for ALC (the average reduction in variance).

3.3 Sequential procedure

Once we select a new design point $\boldsymbol{x}_{N+1}$ using the proposed score, we would, ideally, evaluate the computer model at $\boldsymbol{x}_{N+1}$. Given the output $z(\boldsymbol{x}_{N+1})$, one would re-estimate the parameters of

the surrogate model and reevaluate the covariance structure $c(\cdot,\cdot)$. To move to the next step, one would use again the score to obtain a new design point $\boldsymbol{x}_{N+2}$. However, in some applications, several new design points are processed in batches. In such a case, we could naively add points to the design and recalculate the score. The resulting batches, however, tend to occupy only current high variance regions according to the most recent estimate of the correlation structure, and in our experience the resulting designs are simply not satisfactory.

Updating the correlation structure is imperative to obtain a reasonable sequential procedure. Assuming that we have fixed the correlation parameters to an estimator $\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_N)$, for the GP surrogate model presented in Section 2 the predictive variance $V(\cdot|\boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ does not depend on the actual observed value $z(\boldsymbol{x}_{N+1})$ but only on the design point $\boldsymbol{x}_{N+1}$. Therefore, updating the the predictive variances may be done without waiting for the computer model output. The correlation structure to be used in our score in (3) to find the $n+1$ design point, having observed responses for the first N points, is $c(\boldsymbol{y}_i, \boldsymbol{y}_j) = \sqrt{V(\boldsymbol{y}_i|\boldsymbol{D}_{N+n})V(\boldsymbol{y}_i|\boldsymbol{D}_{N+n})}K_{\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_N)}(\boldsymbol{y}_i, \boldsymbol{y}_j)$. If responses are obtained for k points, the correlation parameters are reestimated to obtain $\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_{N+k})$ (we use the MAP estimator using the MCMC procedure sketched in Section 2.1). The correlation structure is modified to $c(\boldsymbol{y}_i, \boldsymbol{y}_j) = \sqrt{V(\boldsymbol{y}_i|\boldsymbol{D}_{N+n})V(\boldsymbol{y}_i|\boldsymbol{D}_{N+n})}K_{\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_{N+k})}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, etc. This provides a sequential scheme, to be used in our simple 2D example in section 4 and the computer model example in section 5.

In a more general setting we suppose that we have estimations of the predictive variance $V(\boldsymbol{x}|\boldsymbol{D}_N)$ and the correlation structure $K(\boldsymbol{x}, \boldsymbol{y})$, arising from some spatial model perhaps more complex than the GP model described above. One that, for example, considers complicated partitions of the parameter space. Updating the variance may not be simple or even possible since it may indeed depend on the response $z(\boldsymbol{x}_{N+1})$. An alternative is to maintain an "internal" (for design purposes only) GP model and use it for the estimation of the predictive variance $\hat{V}_{GP}(\boldsymbol{x}|\boldsymbol{D}_{N+1})$. In this fashion, a representation for the expected changes in predictive variance could be $\hat{V}(\boldsymbol{x}|\boldsymbol{D}_{N+1}) = V(\boldsymbol{x}|\boldsymbol{D}_N) + \hat{V}_{GP}(\boldsymbol{x}|\boldsymbol{D}_{N+1})$. Note that $\hat{V}(\boldsymbol{x}|\boldsymbol{D}_{N+1})$ is a proxy for the the expected behavior of the predictive variance once the point $\boldsymbol{x}_{N+1}$ is added. We have experimented

successfully with this idea, however, in the rest of the paper, we concentrate on having a GP surrogate model as in Section 2, and then update the variances directly, as described in the previous paragraph.

3.4 Purpose specific designs

Suppose there is a feature of the computer model and/or of the experimental region $u(\boldsymbol{x})$ that one wishes to maximize. So, it is necessary to explore the parameter space to find the maximum of u. After some initial trails it is possible that several parts of the design space have low $u(\boldsymbol{x})$ values, and perhaps some include rugged parts of the output. High predictive uncertainty in those areas will call for more design points in those regions but evaluating the computer experiment there may be irrelevant as far as maximizing u is concerned. Santner *et al.* (2003, Chapter 6) present a review of some optimization driven design criteria. Most of the proposed designs require an initial learning stage, that typically uses a space filling design. After the initial stage, a purpose specific search is conducted. Various levels of complexity in the search algorithms are described by Santner *et al.* (2003). Here we present a very simple alternative, that seamlessly progresses from the learning stage to the purpose specific design.

We propose to bias our score in (3) to focus on the regions that are relevant to the maximization of the feature $u(\boldsymbol{x})$. The proposed bias score is

$$\{u(\boldsymbol{x}_{N+1})\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$$

for some positive w. The score will first spread design points on the grid, to learn about the problem, but after a big enough N it will black out points with low $u(\boldsymbol{x})$ values, thus concentrating the design points on maximizing u. For example, if the design has as main purpose maximization of the computer experiment output we could consider simply $u(\boldsymbol{x}) = \hat{z}(\boldsymbol{x})$, the current predictive value at $\boldsymbol{x}$. The score then becomes $\{\hat{z}(\boldsymbol{x}_{N+1})\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$. This will first behave basically as $A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$, but will prefer high values of $z(\cdot)$, when the design size N is big.

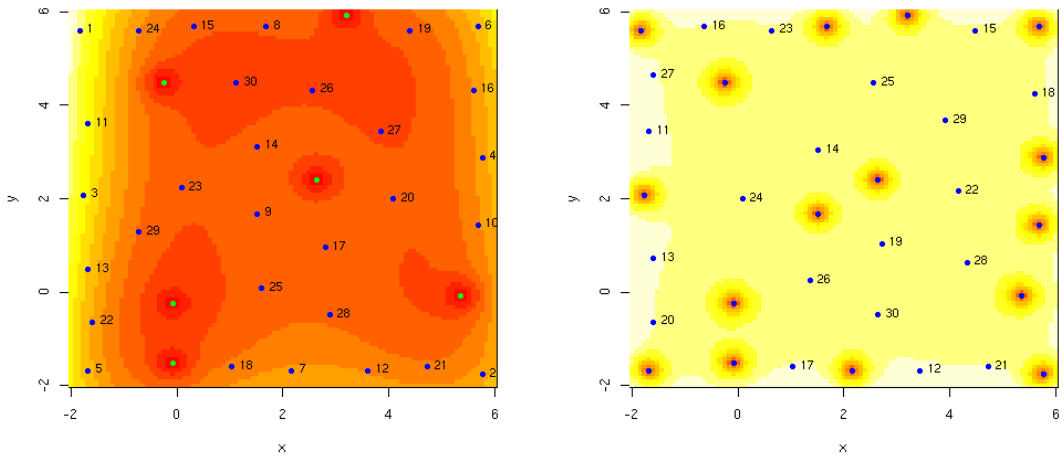
In Section 5 we use the idea of biasing the score in the case where the feature u is proportional to the prior density of $\boldsymbol{\theta}$, $u(\boldsymbol{x}) \propto \pi(\boldsymbol{x})$. This seems appropriate for calibration problems where

there is good prior information about the most likely values in the parameter space. One may also try to use a latin hyper cube design using $\pi(\boldsymbol{x})$ as underlying measure. This will produce spread out design points with high prior density. However, the advantage of the above procedure is that it will first spread points in the whole region, learning from the computer model, and it could potentially set more design points on rugged, high variance regions of the sampling space even though not having high prior density. Since our procedure is sequential we do not need to commit sampling to all parts of the design region from the onset. Rather the design will concentrate on more complicated response areas, which in general are not known *a priori*.

4 Simulated example

To test our score we simulated data according to the model $z(x_1, x_2) = (x_1 - 4.5)^2 + (x_2 - 3.5)^2 + e$, where $e \sim N(0, 3^2)$. We considered a regular grid of 10,200 points in the region $[-2, 6] \times [-2, 6]$ and used a GP (surrogate) model with a quadratic regressor, as explained in Section 2, with correlation function $(1 - \lambda_2)\exp(-3d/\lambda_1) + \lambda_2 \exp\{-3(d/\lambda_1)^2\}$, where d is the distance between any two points. We took 6 points at random as initial trial and then calculated 30 design points, see Figure 1(a). Next we evaluated the first 10 design points, recalculated the model parameters and calculated 20 design points, see Figure 1(b). Apart from small changes in the ordering, due to numerical error, the 20 remaining design points are the same in both cases. This is a most desirable sequential behavior.

Some interesting features of our design are to be highlighted. The points are well spread across the region. They first start to be allocated in the borders of the region but many do lay in the inner region. Points do not show up completely at the edge of the region, although the highest predictive variance does occur at the edge (eg. design point 1 at the left top corner of Figure 1(a) is near but not at this corner). This is due to the fact that not only high variance is required but also high correlation with other points on the grid is considered, as explained in Section 3.2. The spread of points is a particularly desirable feature (as seen in Figure 1), arising from the denominator term in our score (3).



(a) (b)

Figure 1: Data taken from the known model $(x_1 - 4.5)^2 + (x_2 - 3.5)^2$ for (a) 6 random points (green) taken as initial trail and 30 design points from our score (blue). (b) 10 additional points sampled and the score recalculated to generate 20 design points. Topographic colors represent predictive variance, with a grid of 10,200 points. Numbers represent the sequence in which design points were selected.

5 Case Study

We consider the problem of calibrating an intermediate climate computer model. We focus on the MIT2D Climate Model (MIT2DCM) described in Sokolov and Stone (1998) and Forest *et al.* (2006). The MIT2DCM provides simulations of the ocean, surface and upper atmospheric temperature over a grid of 46 different latitudes and 11 vertical layers. The model was run for the period of 1860–1995 and its output was summarized in three low dimensional statistics. We use the summary consisting of the trend of deep ocean temperatures obtained from the pentadatal averages for the 0-3 km deep layer during the period 1952–1995. The model output is controlled via three parameters, $(x_1, x_2, x_3) = \boldsymbol{x}$. These parameters are:

- x_1 - "The rate of diffusion for heat anomalies into the deep-ocean, $\mathcal{K}_v$ ".
- x_2 - "Climate sensitivity, $\mathcal{S}$ ", defined as the equilibrium global mean surface temperature response to a doubling of CO_2.
- x_3 - "The net anthropogenic aerosol forcing $\mathcal{F}_{aer}$ ".

The MIT2DCM was evaluated at an irregular grid consisting of 426 combinations of these parameters. Further details on the statistical analysis of the MIT2DCM can be found in Sansó, Forest and Zantedeschi (2007). Observational records obtained from Levitus et al. (2005) are summarized in the same fashion as the MIT2DCM output to obtain an observational temperature trend of $Z = 1.044336 \times 10^{-3}$ C°/year. The calibration question is to find a probability distribution for the climate parameters ($\mathcal{K}_v$, $\mathcal{S}$, $\mathcal{F}_{aer}$)=$\boldsymbol{\theta}$, given all the prior climate information and the observational records available.

Here we focus on the problem of choosing combinations of the climate parameter values that are useful for the calibration problem. We have very good information about the most likely values of $\boldsymbol{\theta}$ from the literature about climate change. This is used to build the following prior. For $\sqrt{\mathcal{K}_v}$ we consider a beta distribution with support on $(0, 6)$ and parameters $(3.5, 6)$. For $\mathcal{S}$ we use a beta distribution with support $(0, 15)$ and parameters (2.85,14) and for $\mathcal{F}_{aer}$ we use a beta distribution supported on (-1.5,.5) with parameters (4,4). We assume that, *a priori*, the three parameters are

independent. A more detailed discussion of these choices appears in Sansó et al. (2007).

We propose a criterion that biases the choice of parameter values towards the region of high density prior values (as explained in Sectio 3.4). This is achieved by the score

$$\left\{ \frac{f(\boldsymbol{x}_{N+1})}{\max f(\boldsymbol{\theta})} \right\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N).$$

This score depends on the quantity w which needs to be appropriately tuned in relation to the ratio of actual computed points N to possible grid values m. For the MIT2DCM example we took 30 design points at random from the 426 original grid as initial trial. With this we calculated our score to obtain 100 design points. This was done both on the original grid and in a larger regular grid. We proceeded sequentially as explained in section 3.3, using the correlation function $K_{\boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{3} \exp(-\lambda_i |x_i - y_i|)$. The value of w was set to 5 after some tuning.

Figure 2 presents a 3D view of the resulting design and in Figure 3 we present the design weighted by the prior, both using a regular grid, with 30 initial trail points and 100 design points. It is clear how design points are selected in higher prior density regions but nevertheless points are still spread across the design region. Also interesting are the two-by-two projections of Figures 2 and 3 shown in Figures 4 and 5 respectively. Note that for 130 points, less than 1/3 of the original sample, nearly the same predictive variance is expected as with the full sample (see the left bottom panels). Also, in both cases, most of the two by two projections are covered using 30% of the available grid points.

6 Discussion

We have presented a new score to evaluate points to generate sequential designs, mainly in the context of GP models used as surrogate substitutes for complex computer models. This score may be viewed as a simplification of the Active Learning strategy used in robotics (Cohon, 1996). We also generalized the sequential procedure when several design points are needed before the computer output is available. In our tests, the score performed quite well, producing reasonable designs with a far lower computational burden than the Active Learning strategy. Besides the

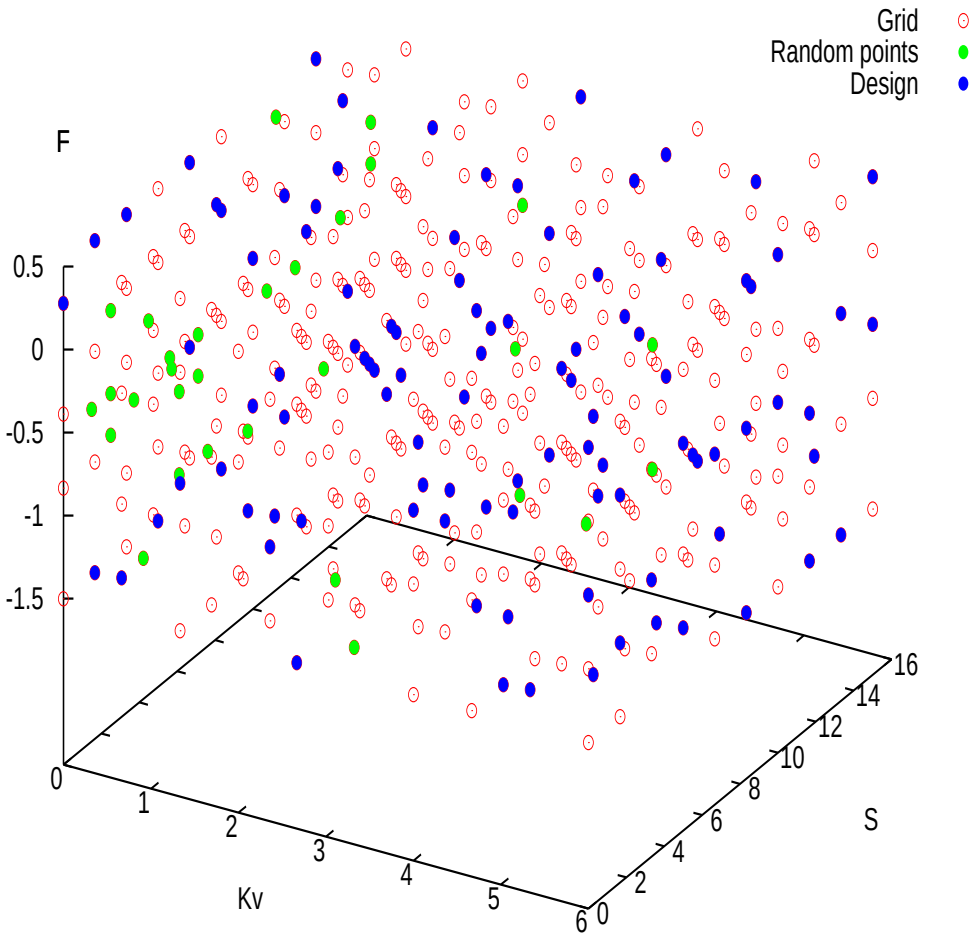


Figure 2: Regular grid (500 points) with 30 random points (green) taken as initial trail and 100 design points from our score (blue).

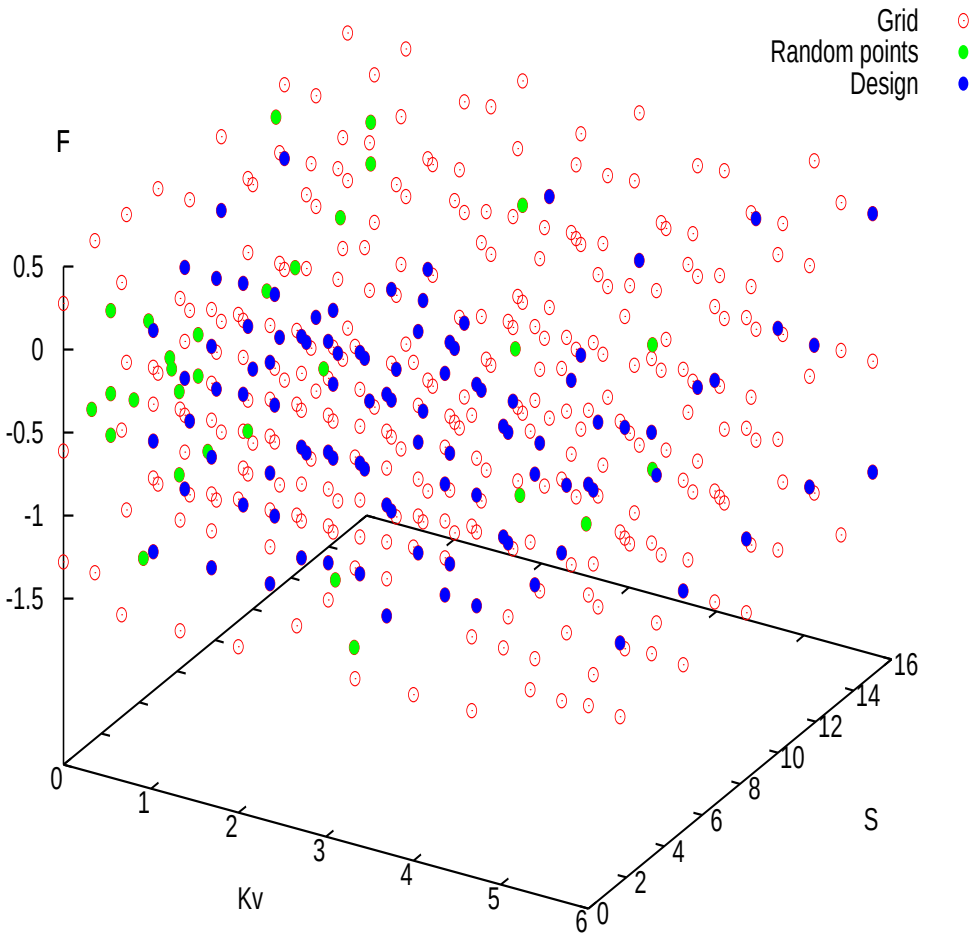


Figure 3: Regular grid (500 points) with 30 random points (green) taken as initial trail and 100 design points from our score weighted by the prior (blue).

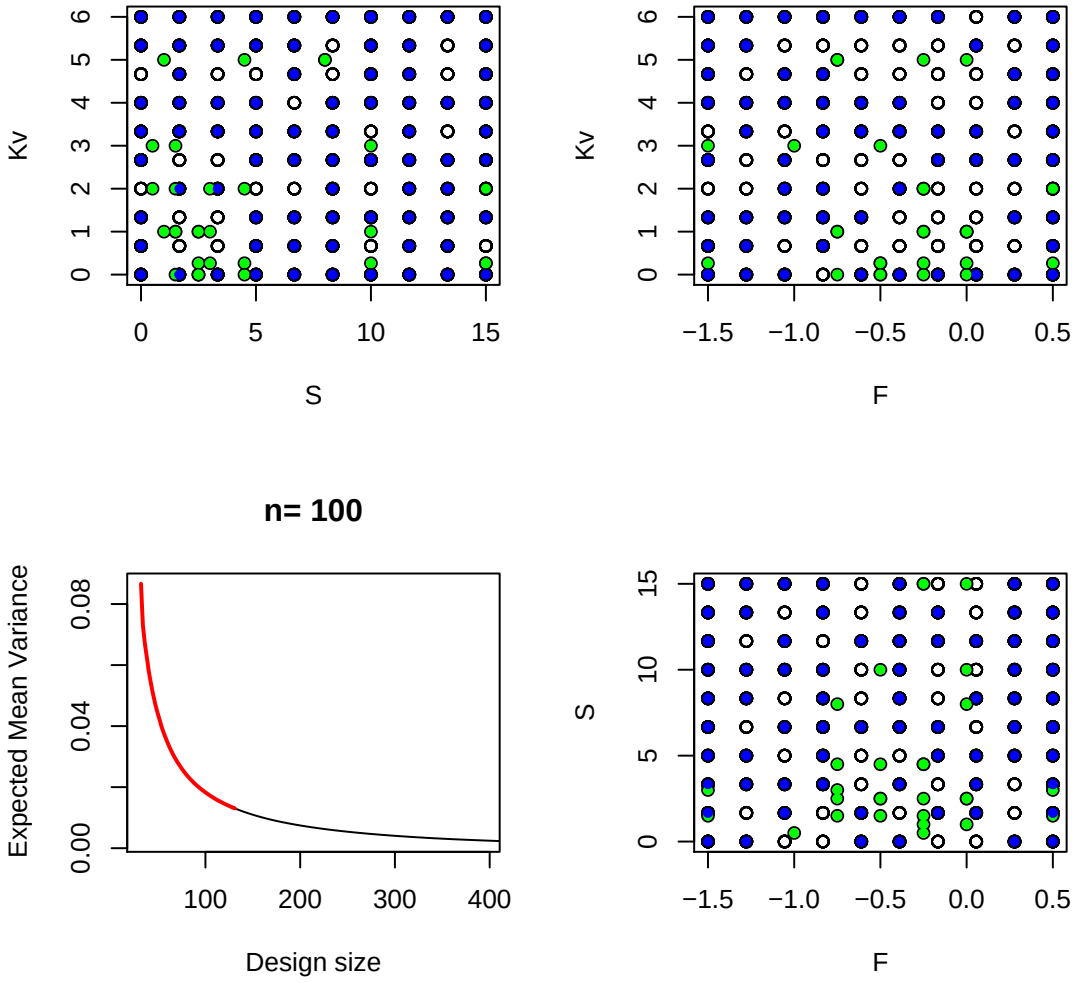


Figure 4: Two by two representation of our design, with a regular grid (500 points) with 30 random points (green) taken as initial trail and the 100 design points taken using our score (blue). Note how with only 130 points most projections are covered. In the left bottom panel we present the expected average predictive variance for all design sizes. With this 130 point design the plot is highlighted (red); the expected variance is nearly the same as for the full 426 point ddsample. Similar results are obtained using the original grid.

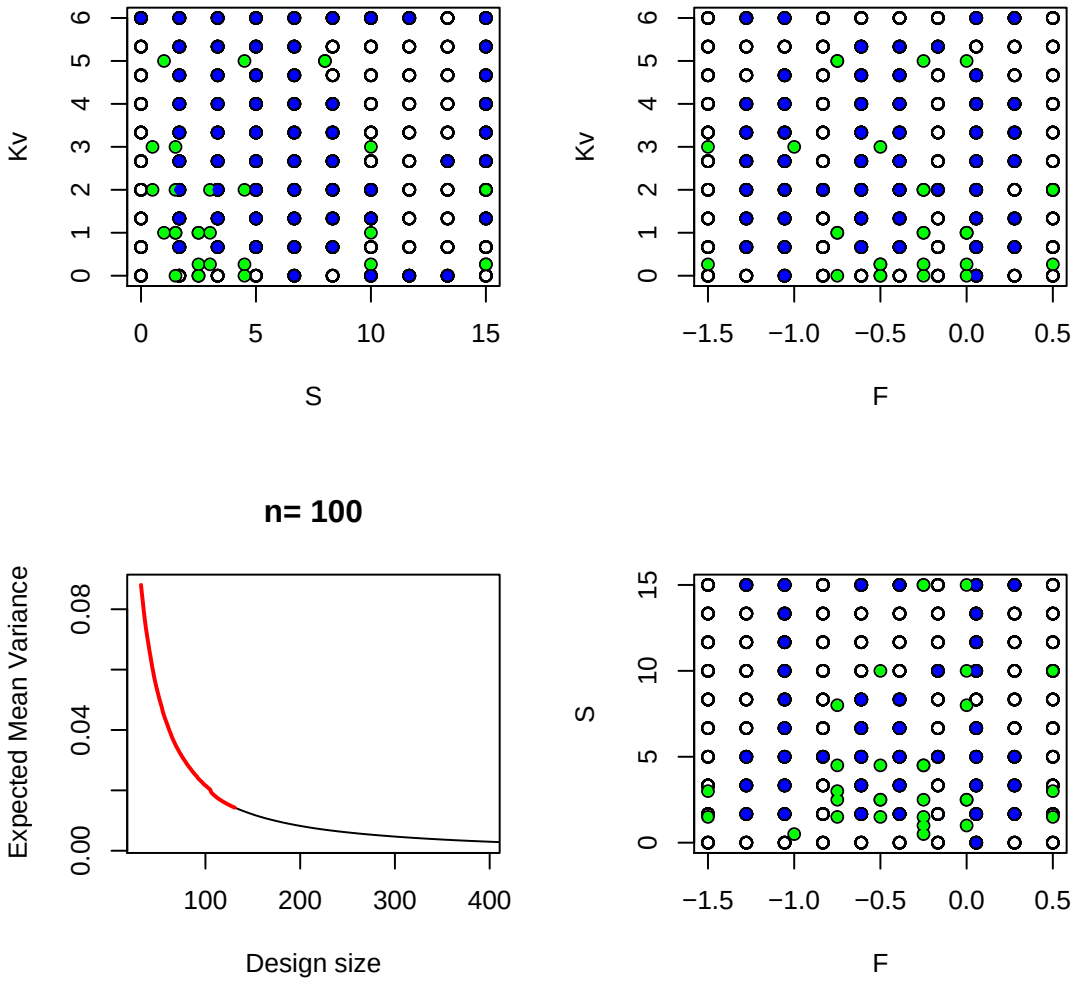


Figure 5: Two by two representation of our design, using a regular grid (500 points), with 30 random points (green) taken as initial trail and the 100 design points taken using our score weighted by the prior (blue). Note how with only 130 points most projections are covered. In the left bottom panel we present the expected average predictive variance for all design sizes. With this 130 point design the plot is highlighted (red); the expected variance is nearly the same as for the full 426 point sample. Similar results are obtained using the original grid weighting the score with the prior.

motivating formal argument (based on standard linear algebra results), our design score is also intuitively clear and may be generalized to other spacial processes. The only requirements are a current estimate of the covariance function and an initial grid of points to base the design. Specifically, we would like to experiment using our design scheme for a surrogate model based on non-stationary treed Gaussian Processes (Gramacy and Lee, 2007).

We have experimented with the use of our biased design strategy for maximization purposes. For the example presented in Section 4 we have obtained very god results. Design points are first scattered around the design region (very much as presented in Figure 1) but are progressively assigned around the maximum. The global maximum is soon found using very few evaluations of the objective function. These and other potentially favorable characteristics of our design scheme should be further investigated and are left for future research.

7 Acknowledgments

Part of this reaerch was completed while J. Andrés Christen was visiting the department of AMS at UCSC with a UC-MEXUS-CONACYT sabbatical grant. The second author was partially supported by the National Science Foundation grant NSF-Geomath 0417753.

References

[1] Bernardo, J.M and Smith, A.F.M. (1994), *Bayesian Theory*, Chichester, UK: Wiley.

[2] Bhatia, R. (1997), *Matrix analysis*, Springer: NY.

[3] Bursztyn, D. and Steinberg, D.M. (2006) "Comparison of designs for computer experiments", *Journal of Statistical Planning and Inference*, **136**(3), 1103–1119.

[4] Christen, J.A. and Fox, C. (2008), "A General Purpose Scale-Independent MCMC Algorithm", (submitted).

[5] Cioppa, T.M. and Lucas, T.W. (2007), "Efficient nearly orthogonal and space-filling Latin hypercubes", *Technometrics*, **49**(1), 45–55.

[6] Cohon, D.A. (1996), "Neural Network Exploration Using Optimal Experiment Design", *Neural Networks*, **9**(6), 1071–1083.

[7] Fang, K.T. and Li, R.Z. (2006), "Uniform design for computer experiments and its optimal properties", *International Journal of Materials and Product Technology*, **25**(1-3), 198–210.

[8] Feynman, R.P. (1985), *"Surely you are joking Mr Feynman!" Adventures of a curious character*, New York: W. W. Norton & Company.

[9] Forest, C.E. , Stone, P.H. and Sokolov, A.P. (2006), "Estimated PDFs of climate system properties including natural and anthropogenic forcings", *Geophys. Res. Let.*, **33**, ??.

[10] Gramacy, R (2005), *Bayesian treed Gaussian process models*, PhD thesis, University of California at Santa Cruz.

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

[12] Horn, R.A. and Johnson, C.R. (1985), *Matrix Analysis*, Cambrige, UK: Cambrige University Press.

[13] Kennedy, M.C. and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models", *Journal of the Royal Statistical Society, Series B*, **63**, 425–464.

[14] Lehman, J.S., Santner, T.J. and Notz, W.I. (2004) "Designing computer experiments to determine robust control variables", *Statistica Scinica*, **14**(2), 571–590.

[15] Mease, D. and Bingham, D. (2006), "Latin hyperrectangle sampling for computer experiments", *Technometrics*, **48**(4), 467–477.

[16] Sansó, B., Forest, C. and Zantedeschi, D. (2007), "Statistical Calibration of Climate System Properties", UCSC-AMS preprints ams2007-06, `http://www.ams.ucsc.edu/reports/trview.php?content=view&name=ams2007-06`.

[17] Santner, T.J., Willimas, B.J. and Notz, W.I. (2003), *The design and analysis of computer experiments*, New York : Springer Verlag.

[18] Sokolov, A. P. and Stone, P. H. (1998), "A flexible climate model for use in integrated assessments", *Climate Dynamics*, **14**, 291-303.

[19] Stinstra, E., Den Hertog, D., Stehouwer, P. and Vestjens, A. (2003), "Constrained maximin designs for computer experiments", *Technometrics*, **45**(4), 340–346.

[20] Williams, B.J., Santner, T.J. and Notz, W.I. (2000), "Sequential designs of computer experiments to minimize integrated response functions", *Statistica Scinica*, **10**, 1133–1152.

A Formal argument for (3)

In a GP as above, given the inverse of the correlation matrix $\boldsymbol{R}_{\boldsymbol{D}_N}^{-1}$, σ^2 and $\boldsymbol{\beta}$ we have that

$$V(\boldsymbol{y}|\boldsymbol{D}_N) = \sigma^2(1 - r(\boldsymbol{y})'\boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y})),$$

where $r(\boldsymbol{y})' = (K(\boldsymbol{y}, \boldsymbol{x}_1), K(\boldsymbol{y}, \boldsymbol{x}_2), \ldots, K(\boldsymbol{y}, \boldsymbol{x}_N))$ (we assume any correlation parameters $\boldsymbol{\lambda}$ to be fixed and write $K_{\boldsymbol{\lambda}}(\cdot, \cdot) = K(\cdot, \cdot)$).

Therefore

$$ALC(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{\sigma^2}{m} \sum_{j=1}^{m} r_1(\boldsymbol{y}_j)'\boldsymbol{R}_{\boldsymbol{D}_{N+1}}^{-1} r_1(\boldsymbol{y}_j) - r(\boldsymbol{y}_j)'\boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y}_j),$$

where $\boldsymbol{D}_{N+1} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1})$ and $r_1(\boldsymbol{y})' = (K(\boldsymbol{y}, \boldsymbol{x}_1), K(\boldsymbol{y}, \boldsymbol{x}_2), \ldots, K(\boldsymbol{y}, \boldsymbol{x}_N), K(\boldsymbol{y}, \boldsymbol{x}_{N+1}))$.

We will try to establish a lower bound for

$$D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = r_1(\boldsymbol{y})'\boldsymbol{R}_{\boldsymbol{D}_{N+1}}^{-1} r_1(\boldsymbol{y}) - r(\boldsymbol{y})'\boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y}) \tag{4}$$

in order to obtain a lower bound for $ALC(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$.

We write

$$\boldsymbol{R}_{\boldsymbol{D}_{N+1}} = \boldsymbol{R} + \boldsymbol{E} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{D}_N} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & r(\boldsymbol{x}_{N+1}) \\ r(\boldsymbol{x}_{N+1})' & 0 \end{bmatrix}.$$

A simple calculation leads to

$$D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + r_1(\boldsymbol{y})'(\boldsymbol{R}+\boldsymbol{E})^{-1} r_1(\boldsymbol{y}) - r_1(\boldsymbol{y})'\boldsymbol{R}^{-1} r_1(\boldsymbol{y}).$$

Let for $\boldsymbol{A} = (\boldsymbol{R}+\boldsymbol{E})$, $\boldsymbol{R}$ or $\boldsymbol{E}$, $\lambda_1(\boldsymbol{A}) \geq \lambda_2(\boldsymbol{A}) \geq \cdots \geq \lambda_{N+1}(\boldsymbol{A})$ be its eigenvalues and $e_1(\boldsymbol{A}), e_2(\boldsymbol{A}), \ldots, e_{N+1}(\boldsymbol{A})$ the corresponding normalized eigenvectors. We assume that $\lambda_{N+1}(\boldsymbol{R}+\boldsymbol{E}) > 0$ and $\lambda_{N+1}(\boldsymbol{R}) > 0$. We see that

$$D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R}+\boldsymbol{E})^{-1} (e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2 - \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1} (e_i(\boldsymbol{R})' r_1(\boldsymbol{y}))^2.$$

Assuming that the average change in eigenvectors is small we assume that

$$\sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1} (e_i(\boldsymbol{R})' r_1(\boldsymbol{y}))^2 \approx \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1} (e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2, \tag{5}$$

and obtain

$$D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq \sum_{i=1}^{N+1} (\lambda_i(\boldsymbol{R}+\boldsymbol{E})^{-1} - \lambda_i(\boldsymbol{R})^{-1}) (e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2,$$

simply by assuming equality in (5). Indeed, a proper lower bound will need further justification, but the argument as presented is only to show a justification for our score in (3), se below. Note that $(e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2 = r_1(\boldsymbol{y})' r_1(\boldsymbol{y}) (\cos\theta_i)^2$, where θ_i is the angle between vectors $e_i(\boldsymbol{R}+\boldsymbol{E})$ and $r_1(\boldsymbol{y})$. On the other hand,

$$\lambda_i(\boldsymbol{R}+\boldsymbol{E})^{-1} - \lambda_i(\boldsymbol{R})^{-1} = \frac{1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})}{\lambda_i(\boldsymbol{R}+\boldsymbol{E})} \geq \frac{1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})}{\lambda_1(\boldsymbol{R}) + \lambda_1(\boldsymbol{E})}.$$

The last inequality is part of Weyl's inequalities (Bhatia, 1997, Chap III) since $\lambda_i(\boldsymbol{R}+\boldsymbol{E}) \leq \lambda_i(\boldsymbol{R}) + \lambda_1(\boldsymbol{E}) \leq \lambda_1(\boldsymbol{R}) + \lambda_1(\boldsymbol{E})$. Moreover, using Geršgorin discs (see Horn and Johnson, 1985, p.346) we see that $\lambda_1(\boldsymbol{R}) \leq C(\boldsymbol{R}) = \max_j \sum_{i=1}^{N} |K(\boldsymbol{x}_i, \boldsymbol{x}_j)|$. By direct calculations we see that $\lambda_1(\boldsymbol{E}) = ||r(\boldsymbol{x}_{N+1})||$. Thus we obtain

$$D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq \frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{C(\boldsymbol{R}) + ||r(\boldsymbol{x}_{N+1})||} \sum_{i=1}^{N+1} (1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R}))(\cos\theta_i)^2.$$

Assuming the above lower bound is positive, $\frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{C(\boldsymbol{R})+||r(\boldsymbol{x}_{N+1})||}$ will be a proxy for the magnitude of such bound. We are able to calculate this last term, as opposed to the sum, right term, above that is just as difficult to calculate as $D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1})$. Note also that the sum does not depend on the absolute correlation values of $\boldsymbol{y}$ with $\{\boldsymbol{D}_N, \boldsymbol{x}_{N+1}\}$ only in its direction, no matter how small $||r_1(\boldsymbol{y})||$ is. Without further justification, we take $\frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{C(\boldsymbol{R})+||r(\boldsymbol{x}_{N+1})||}$ as a coarse representation proportional to a lower bound of $D(\boldsymbol{y}|\boldsymbol{D}_N, \boldsymbol{x}_{N+1})$. Adding over m we obtain the score

$$A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{\sigma^4}{\sigma^2} \frac{\frac{1}{m}\sum_{j=1}^{m} K(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2 + \frac{1}{m} C^1_{\boldsymbol{D}_N}}{C^2_{\boldsymbol{D}_N} + \sqrt{\sum_{i=1}^{N} K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2}}$$

with $C^1_{\boldsymbol{D}_N} = \sum_{j=1}^{m}\sum_{i=1}^{N} K(\boldsymbol{y}_j, \boldsymbol{x}_i)^2$ and $C^2_{\boldsymbol{D}_N} = \max_j \sum_{i=1}^{N} |K(\boldsymbol{x}_i, \boldsymbol{x}_j)|$. Changing correlations to covariances the score becomes

$$A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N) = \frac{\frac{1}{m}\sum_{j=1}^{m} c(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2 + \frac{1}{m} C^1_{\boldsymbol{D}_N}}{C^2_{\boldsymbol{D}_N} + \sqrt{\sum_{i=1}^{N} c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2}}.$$

Maximizing $ALC(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$ and $A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$ will not, in general, lead to the same results and indeed we expect the latter to be only an educated (suboptimal) guess of the former. However, suboptimal solutions might lead to more spread out designs, permitting better exploration of the design space, with the added benefit of the far lower computational burden of calculating $A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$. In fact, $A(\boldsymbol{x}_{N+1}|\boldsymbol{D}_N)$ may be calculated virtually in as many points as desired.