

# Parameter estimation for Gaussian process emulators

Y. Andrianakis and P. G. Challenor

July 16, 2011

## Abstract

Gaussian processes are often used as statistical representations (emulators) of computer models, due to their flexibility in capturing the shape of smooth functions. They do have however, a number of parameters, the estimation of which is a fundamental step towards building an emulator. The estimation of these parameters is the problem considered in this work. An analytical marginalisation of the parameters is advocated, when this is tractable, and when it is not, the effect of a numerical marginalisation is investigated and compared to the use of a single parameter estimate, a method also known as plug-in. A major contribution of this work is the proposal of a correlation length prior, which yields a proper posterior by imposing limits on the acceptable values of the correlation lengths. We argue that such limits can be defined a priori, and not only they do not deteriorate the inferences made with the emulator, but on the contrary, they reduce the risk of numerical instabilities, facilitate the convergence of MCMC algorithms and generally help the validation process. The considered parameter estimation methods and the proposed prior are tested on emulators built for two climate models.

## 1 Introduction

An ever increasing number of natural or man made systems are being studied with the help of computer models, in an endeavour to enhance our understanding of the underlying processes and our ability to predict their behaviour. The complexity of modern computer models can be so high, that even a single run can require a substantial amount of time to complete. Furthermore, the execution of tasks such as model calibration or input sensitivity analysis can require thousands of runs, which for some classes of models may be impractical or simply infeasible.

Computer model emulation is a technology that helps addressing some of the aforementioned problems. It can essentially be seen as the development of a meta-model (emulator) that acts as a computationally efficient surrogate to the original computer model (simulator). The emulators we examine in this work, were first introduced in [1], and are based on Gaussian processes (GP). This class of emulators uses a GP as a prior for the simulator's output, or in other words, it assumes that the simulator's output is a random sample drawn from a joint Gaussian distribution. The development of an emulator allows to predict the simulator's output at untested inputs almost instantaneously, giving at the same time confidence intervals for this prediction. Additionally, related research has given rise to methods that perform a probabilistic model calibration [2] or sensitivity analysis [3] in a very efficient manner.

Gaussian processes have a number of parameters that are generally unknown. Therefore, they either need to be estimated or marginalised, in the Bayesian sense. Marginalisation offers the advantage of formally accounting for the uncertainty about the parameters' true value, but also makes the estimates of any non-marginalised parameters more meaningful and stable [4]. However, marginalisation may not always be feasible or practical for one or more parameters of the model.

In this case, it is common to find a best - in some sense - parameter estimate and consider this as being the true value, a method that is known as ‘plug-in’ [2].

The Gaussian process we consider in this work has three parameters: the regression coefficients  $\beta$ , the scaling factor  $\sigma^2$  and a vector of parameters  $\delta$  that control the smoothness of the output, known as correlation lengths. Under the assumption of a linear mean function, an analytical marginalisation is possible for the regression coefficients and the scaling factor. The marginalisation of the correlation lengths however, is analytically intractable, even for the most elementary forms of the correlation function. We examine two approaches for their treatment: the first is the plug-in method of [2], and the second is a numerical marginalisation, which is based on drawing a number of samples of  $\delta$  from its posterior distribution. The latter approach can account for the uncertainty about the true value of  $\delta$  but comes at a computational cost that can be considerable. The merits of each approach are examined and guidelines are given as to when each one might be more suitable.

An important hurdle when drawing samples of  $\delta$  can be that the use of a uniform prior yields an improper posterior, which is known to hinder the convergence of MCMC algorithms [5]. A major contribution of this work is the proposal of a correlation length prior, which is continuous, differentiable and yields a proper posterior. The proposed prior is based on defining a range of interest for the correlation lengths, where the prior is flat, while the parameter space that is outside these limits is given a very low probability. The result of this effective limitation of the parameter space on the emulator is evaluated with a number of examples. The proposed prior is also compared with the reference prior, which is known to yield a proper posterior, but has a significantly higher computational cost. An approximate method that involves drawing samples from a multivariate normal distribution that is fitted on the posterior distribution of the log correlation lengths, is also considered.

The structure of the paper is as follows: section 2 details the GP model of the simulator and section 3 discusses the various approaches for treating its parameters. Section 4 gives the motivation behind the use of correlation length priors, and presents the proposed prior, together with a review of the reference prior. Methods for drawing samples from a posterior distribution of the correlation lengths are presented in section 5 and results from applying the parameter estimation methods to two different climate models, are given in section 6. Finally, the Appendix contains expressions for the derivatives of the posterior of the correlation lengths, and a section on the optimal numerical implementation of the Gaussian process emulators, using the Cholesky decomposition.

## 2 Gaussian Process model of the simulator

Let  $f(\mathbf{x})$  denote the simulator’s output for the input vector  $\mathbf{x}$ . We are assuming that the simulator has a  $p$  dimensional input  $\mathbf{x} = [x_1, x_2, \dots, x_p] \in \mathbb{R}^p$  and a 1 dimensional output  $f(\mathbf{x}) \in \mathbb{R}$ . The central idea behind Gaussian process emulators is the utilisation of a joint normal distribution as a prior for the simulator’s output. We express this modelling assumption as

$$p(f(\mathbf{x})|\beta, \sigma^2, \delta) \sim \mathcal{N}(h(\mathbf{x})^T\beta, \sigma^2c(\mathbf{x}, \mathbf{x})). \quad (1)$$

The mean response of this model is given by the inner product  $h(\mathbf{x})^T\beta$ , where  $h(\mathbf{x})$  is a  $(q \times 1)$  vector of predetermined functions of  $\mathbf{x}$ , known as regression (or basis) functions, and  $\beta$  is a  $(q \times 1)$  vector of regression coefficients. This form of the mean function is called linear, and can be fairly flexible because of the arbitrary choices that can be made for the basis functions; at the same time, its linearity with respect to the regression coefficients simplifies the subsequent analysis.

The covariance of the model is given by  $\sigma^2c(\mathbf{x}, \mathbf{x}')$ , where  $\sigma^2$  controls the overall scaling (variance) of the process. The correlation between the simulator’s outputs, as induced by the proximity of their

inputs in  $\mathbb{R}^p$ , is modelled with the correlation function  $c(\mathbf{x}, \mathbf{x})$ . The family of correlation functions we will consider are stationary and their value depends on the ratios  $\frac{|x_i - x'_i|}{\delta_i}$ ,  $i \in [1..p]$ . The strictly positive parameters  $\delta = \{\delta_i : i \in [1..p]\}$  are called correlation lengths and control the smoothness of the output. Correlation functions that belong in the above category are the Spherical, the Power exponential, the Rational quadratic and the Matérn [6].

Having specified the Gaussian process prior for the simulator, the next step is to observe the simulator's output for a number of different inputs. These inputs are called design points and we collectively denote them by  $D = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ . The simulator's output at these points is denoted by the  $(n \times 1)$  vector  $f(D)$ .

Using Bayes' theorem and standard identities for multivariate Gaussian distributions (e.g. [7], App. A2), the distribution of the simulator's output at a new input point  $\mathbf{x}$  conditional on the previous simulator runs and the GP parameters  $\beta, \sigma^2$  and  $\delta$  is

$$p(f(\mathbf{x})|f(D), \beta, \sigma^2, \delta) = \mathcal{N}(m_0(\mathbf{x}), \sigma^2 u_0(\mathbf{x}, \mathbf{x})) \quad (2)$$

where

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta + c(\mathbf{x}, D) A^{-1} (f(D) - H \beta) \\ u_0(\mathbf{x}, \mathbf{x}) &= c(\mathbf{x}, \mathbf{x}) - c(\mathbf{x}, D) A^{-1} c(D, \mathbf{x}). \end{aligned}$$

In the above equations, the argument  $D$  implies that the respective functions apply to all the elements of  $D$ , that is,  $c(\mathbf{x}, D) \equiv [c(\mathbf{x}, \mathbf{x}_1), c(\mathbf{x}, \mathbf{x}_2), \dots, c(\mathbf{x}, \mathbf{x}_n)]$  and  $c(D, \mathbf{x}) \equiv c(\mathbf{x}, D)^T$ . We also define the correlation matrix of the design points  $D$  as  $A \equiv c(D, D)$ , and  $H \equiv h(D)^T = [h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_n)]^T$ .

### 3 Treatment of the parameters

The predictive distribution given in eq. 2, is conditional on the three parameters, whose value is unknown. The simplest method of treating them is to maximise the likelihood  $p(f(D)|\beta, \sigma^2, \delta)$ , with respect to  $\beta, \sigma^2$  and  $\delta$ , and use those estimates in the predictive distribution of eq. 2. Incidentally, the values of  $\beta$  and  $\sigma^2$  that maximise the above expression for a given value of  $\delta$ , can be derived in closed form and are

$$\hat{\beta} = (H^T A^{-1} H)^{-1} H^T A^{-1} f(D) \quad (3)$$

$$\hat{\sigma}_0^2 = \frac{(f(D) - H \hat{\beta})^T A^{-1} (f(D) - H \hat{\beta})}{n}. \quad (4)$$

Note that 3 and 4, are functions of  $\delta$ , which enters the equations via the matrix  $A$ . A closed form expression for the maximum likelihood estimate of  $\delta$  does not exist, hence a solution has to be found numerically.

An alternative approach is to treat  $\beta, \sigma^2$  and  $\delta$  as nuisance parameters and marginalise them. This method takes formally into account the uncertainty about their true values and incorporates this in the predictions. A parameter can be marginalised by integrating it out from its joint distribution with  $f(\mathbf{x})$ , conditional on the other parameters and the observations. This integration however, may not always be analytically tractable, in which case a Monte Carlo or other numerical integration method can be employed. In the present work, we take the Monte Carlo approach, according to which, a number of samples are first drawn from the posterior distribution of the given parameter, and they are subsequently used for carrying out the numerical integration. For the problem at hand, an analytical marginalisation is possible for the regression coefficients  $\beta$  and the scaling coefficient  $\sigma^2$ , as will be shown in section 3.1. A similar approach is not possible for the correlation lengths  $\delta$  and a number of alternative methods for their treatment will be considered in section 3.2.

### 3.1 Marginalisation of $\beta$ and $\sigma^2$

Before marginalising the parameters, some assumptions need to be made about their prior distributions. We assume that the three parameters of the GP are a priori independent, and we use the standard non-informative prior for  $\beta$  and  $\sigma^2$ ,  $p(\beta, \sigma^2) \propto \sigma^{-2}$  [6]. The selection of a prior density for the correlation lengths is a more involved topic, and its discussion will be deferred to section 4.

Given the prior for  $\beta$  we can show that its posterior density is

$$p(\beta|f(D), \sigma^2, \delta) \propto \mathcal{N}\left(\hat{\beta}, \sigma^2(H^T A^{-1} H)^{-1}\right). \quad (5)$$

Multiplying with  $p(f(\mathbf{x})|f(D), \beta, \sigma^2, \delta)$  and integrating w.r.t.  $\beta$  we get

$$p(f(\mathbf{x})|f(D), \sigma^2, \delta) \propto \mathcal{N}(m(\mathbf{x}), \sigma^2 u_1(\mathbf{x}, \mathbf{x})), \quad (6)$$

where  $m(\mathbf{x})$  equals  $m_0(\mathbf{x})$  after substituting  $\beta$  with  $\hat{\beta}$  and

$$u_1(\mathbf{x}, \mathbf{x}) = u_0(\mathbf{x}, \mathbf{x}) + (h(\mathbf{x})^T - c(\mathbf{x}, D)A^{-1}H) \\ (H^T A^{-1}H)^{-1}(h(\mathbf{x})^T - c(\mathbf{x}, D)A^{-1}H)^T.$$

The likelihood of  $\sigma^2$  and  $\delta$  after  $\beta$  has been marginalised can be found by multiplying  $p(f(D)|\beta, \sigma^2, \delta)$  with  $p(\beta)$  and integrating  $\beta$  out. In our case this is

$$p(f(D)|\sigma^2, \delta) \propto \frac{|A|^{-1/2}|H^T A^{-1}H|^{-1/2}}{(2\pi\sigma^2)^{\frac{n-q}{2}}} \exp\left\{-\frac{\hat{\sigma}^2}{2\sigma^2}\right\}, \quad (7)$$

with

$$\hat{\sigma}^2 = (y - H\hat{\beta})^T A^{-1}(y - H\hat{\beta}). \quad (8)$$

It can also be shown that the value of  $\sigma^2$  that maximises the above expression is  $\hat{\sigma}_1^2 = \hat{\sigma}^2/(n - q)$ .

Marginalisation of the regression coefficients is equivalent to the Restricted Maximum Likelihood (REML) [8], and as pointed out by Harville [9], it is equivalent to maximising the likelihood function associated with  $n - q$  linearly independent error contrasts rather than maximising the full likelihood.

The scaling factor  $\sigma^2$  can be marginalised using the same procedure. We can first show that its posterior density is

$$p(\sigma^2|f(D), \delta) \sim \text{IG}\left(\frac{n - q}{2}, \frac{\hat{\sigma}^2}{2}\right), \quad (9)$$

with IG denoting the inverse Gamma distribution, and that multiplication with  $p(f(\mathbf{x})|f(D), \sigma^2, \delta)$  and integration yields

$$p(f(\mathbf{x})|f(D), \delta) \propto \left(1 + \frac{(f(\mathbf{x}) - m(\mathbf{x}))^2}{\hat{\sigma}^2 u_1(\mathbf{x}, \mathbf{x})}\right)^{-\frac{n-q+1}{2}}. \quad (10)$$

The above is a Student's t distribution with  $n - q$  degrees of freedom. The mean of  $f(\mathbf{x})$  is  $m(\mathbf{x})$  and its variance is  $V(\mathbf{x}, \mathbf{x}') = \frac{\hat{\sigma}^2}{n - q - 2} u_1(\mathbf{x}, \mathbf{x}')$ . Finally, the likelihood of  $\delta$  can be found by integrating the product of  $p(f(D)|\sigma^2, \delta)$  with the prior of  $\sigma^2$ , and is given by

$$p(f(D)|\delta) \propto |A|^{-1/2}|H^T A^{-1}H|^{-1/2} (\hat{\sigma}^2)^{-\frac{n-q}{2}}. \quad (11)$$

It is interesting to note that  $p(f(D)|\hat{\sigma}_1^2, \delta) \propto p(f(D)|\delta)$ , or that the marginal likelihood of  $\delta$  is proportional to the joint likelihood of  $\delta$  and  $\sigma^2$ , when  $\hat{\sigma}_1^2$  is used as an estimate for the latter.

This implies that the marginalisation of  $\sigma^2$  does not affect the estimation of  $\delta$ . Additionally, both  $p(f(\mathbf{x})|f(D), \hat{\sigma}_1^2, \delta)$  and  $p(f(\mathbf{x})|f(D), \delta)$  have the same mean, and the variance of the latter distribution is only larger by a factor of  $\frac{n-q}{n-q-2}$ . This shows that incorporating our uncertainty about the true value of  $\sigma^2$  does not affect the posterior mean, but increases the variance of the predictions. However, as the number of points gets significantly larger than the number of the regression coefficients, i.e.  $n \gg q$ , the predicted variances of the two approaches become nearly identical.

Regardless of the method applied for estimating the correlation lengths, the marginalisation of  $\beta$  and  $\sigma^2$  is recommended, because the estimates of  $\delta$  that come from the marginalised (with respect to  $\beta$  and  $\sigma^2$ ) likelihood, tend to be more meaningful and stable [4].

## 3.2 Correlation length estimation

Unlike  $\beta$  and  $\sigma^2$ , the analytic marginalisation of the correlation lengths is not tractable. At this point, we are faced with two options: the first is to find a value of  $\delta$  that maximises the likelihood  $p(f(D)|\delta)$  or a posterior  $p(\delta|f(D))$ , and use this estimate as if it were the true value of  $\delta$ . This method, proposed in [2], has the benefit of being the most simple computationally. The second method involves drawing samples from a posterior distribution of  $\delta$  and use these to carry out the marginalisation numerically. This method is more expensive computationally, not only because an algorithm for sampling the posterior distribution of  $\delta$  has to be used, but also because the mean and variance of the predictions have to be calculated for each of the samples drawn. However, this approach can account for the uncertainty about the true value of  $\delta$ , which can be significant when either the number of design points  $n$  is not sufficiently large, or when the simulator's output is not adequately captured by the Gaussian process model. These two approaches are presented in the next two sections.

### 3.2.1 Point estimates of $\delta$

The simplest approach for treating the parameter  $\delta$  is to obtain a point estimate that maximises the likelihood  $p(f(D)|\delta)$  or the resulting posterior, if we have any prior information about  $\delta$ . For example, if we consider  $p(\delta) \propto \text{const}$ , we seek a point estimate  $\hat{\delta}$ , such that

$$\hat{\delta} = \arg \max_{\delta} (p(f(D)|\delta)). \quad (12)$$

This estimate is then used for the evaluation of the predictive distribution  $p(f(\mathbf{x})|f(D), \hat{\delta})$  [2].

The likelihood function can have multiple maxima, especially in high dimensional problems. A potential multimodality of the likelihood can be verified by repeatedly applying the optimisation algorithm, using a different starting point each time. If the likelihood function is indeed found to be multimodal, the simpler and most common solution would be to use the higher (global) mode. In practice however, this may not necessarily be the optimal solution. For example, if we consider two modes in 10 dimensions, with the first being 10 times taller and the second having twice the standard deviation in all dimensions, the second mode will have approximately 99% of the probability mass, despite being 10 times shorter. Therefore, when faced with multiple modes, a more cautious approach would be to examine the width of the mode (e.g. by evaluating its Hessian matrix) in addition to its height. For more on this point, the reader may consult [10]. An alternative and slightly simpler approach would be to test the most prominent modes and use the one that results in a valid emulator.

Even though an optimisation algorithm can be used for solving eq. 12 directly, it is useful to first

apply a logarithmic transformation to the correlation lengths and optimise using the transformed variable. A transformation that can be applied is

$$\tau = \ln(\delta^2). \quad (13)$$

Subsequently, the optimisation problem is transformed to

$$\hat{\tau} = \arg \max_{\tau} (p(f(D)|\tau)). \quad (14)$$

Once  $\hat{\tau}$  is found, the maximum likelihood estimate of  $\delta$  is simply given by  $\hat{\delta} = \exp(\hat{\tau}/2)$ . This transformation offers the benefit of making the optimisation problem unconstrained, because  $\tau$  can take any value in  $(-\infty, \infty)$ , while  $\delta$  is constrained in  $(0, \infty)$ . Another advantage is that the logarithmic transformation is claimed to make the likelihood more Gaussian [11], which can also help the search for the mode. Note also that the Jacobian is not included in the transformation, because in that case the modes with respect to  $\tau$  and  $\delta$  would no longer be related via equation 13. Finally, the optimisation process can be aided by the existence of closed form derivatives for the log likelihood  $\ln(p(f(D)|\tau)$ , through the use of algorithms such as Newton’s method or Fisher’s scoring algorithm. The derivatives of the log likelihood function are given in appendix A.

### 3.2.2 Accounting for the uncertainty in $\delta$

Using a point estimate for  $\delta$  assumes that this represents the true value of the correlation lengths. This approach does not take into account the uncertainty about the true value of  $\delta$ , which is the case with the marginalisation of  $\beta$  and  $\sigma^2$ . When a large number of design points are available and the Gaussian process model is a good fit to the simulator’s output, it is possible to estimate the correlation lengths accurately enough, so that accounting for their uncertainty may not offer any substantial benefits. However, when the number of design points is limited and/or when the simulator’s output deviates from the Gaussian process model, either by not being smooth or stationary enough, then accounting for the uncertainty in  $\delta$  can increase the predictive variance and yield a valid emulator.

A method for accounting for the uncertainty in  $\delta$  is via Monte Carlo integration. This can be achieved by first drawing a number of samples from the posterior distribution  $p(\delta|f(D))$ , which we denote as  $\delta^i$ , with  $i \in [1..M]$ ,  $M$  being their total number. Once the samples have been drawn, we can do inferences using the predictive distribution  $p(f(\mathbf{x})|f(D))$ . In particular, the mean and the variance of the above distribution are

$$\hat{m}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M m^i(\mathbf{x}) \quad (15)$$

$$\hat{V}(\mathbf{x}, \mathbf{x}') = \frac{1}{M} \sum_{i=1}^M V^i(\mathbf{x}, \mathbf{x}') + \frac{1}{M} \sum_{i=1}^M [m^i(\mathbf{x}) - \hat{m}(\mathbf{x})] [m^i(\mathbf{x}') - \hat{m}(\mathbf{x}')] \quad (16)$$

where  $m^i(\mathbf{x})$ ,  $V^i(\mathbf{x}, \mathbf{x}')$  are the posterior mean and variance  $m(\mathbf{x})$  and  $V(\mathbf{x}, \mathbf{x}')$  calculated using  $\delta^i$ . Methods for drawing samples from the posterior of  $\delta$  are considered in section 5.

## 4 Prior densities for the correlation lengths

### 4.1 Motivation

A characteristic of the likelihood function  $p(f(D)|\delta)$  is that it results in an improper posterior when a uniform prior is used for  $\delta$  [2,6]. This is a consequence of the likelihood being bounded away from

zero for  $\delta \rightarrow 0$  and  $\delta \rightarrow \infty$ . Palmer and Pettit [5] argue that sampling from improper posterior distributions can hinder the convergence of MCMC algorithms. A remedy to this condition is the application of a proper prior, which will result in a proper posterior distribution.

Another condition that often appears in high dimensional problems is that the likelihood can be very flat with respect to one or more variables as  $\delta \rightarrow \infty$ . A common cause is that the respective inputs are either inactive, or appear to be so due to a small number of design points. The result is that the optimisation algorithm can converge towards solutions that have unrealistically high correlation length values in some dimensions. This can create numerical problems, for example with the inversion of  $A$ , while at the same time, restricting the maximum value that a correlation length takes does not have a significant impact on the resulting emulator, as we will show in section 6.3. A prior distribution on the correlation lengths can also be used to prevent the optimisation algorithm from converging to extremely large correlation length values.

One method for ensuring the propriety of the posterior, is to use a vague proper prior, such as the inverse Gamma [6] or the exponential [4]. The main drawback of this approach is that common proper priors can affect the shape of the likelihood near its mode(s) and can introduce a probability mass in regions of low likelihood. This results in a poor frequentist coverage of the Bayesian credible intervals, which is considered a useful method for the evaluation of non-informative priors [12].

A prior that yields a proper posterior distribution and results in good coverage of the Bayesian credible intervals, is the reference prior, [4,6,13]. Its derivation is based on maximising the Kullback Leibler distance between the prior and the posterior distribution, in an effort to ensure that the prior carries the least amount of information possible about the posterior. The reference prior is relatively flat in the areas where the likelihood has the majority of its mass, and decays to zero for extreme values of the correlation lengths, thus yielding a proper posterior. It is however a computationally expensive prior, as it requires knowledge of the design matrix, and its cost increases substantially with the dimensionality of the input.

In this work we propose a prior density function that has the following properties:

- is proper and therefore yields a proper posterior
- does not affect the shape of the likelihood at the parameter range of interest, nor does it introduce a spurious probability mass elsewhere
- is computationally cheap
- has simple derivatives and can be used with a derivative-based optimisation algorithm

Unlike the reference prior, the proposed prior does not assume an a priori knowledge of the design points correlation matrix  $A$ . Therefore, the range of interest for  $\delta$  has to be specified by the user. However, we do not view this as a drawback but more as an advantage of being able to incorporate in the model any prior knowledge on the range of interest, as well as having the choice on how specific or vague this might be, according to the Bayesian paradigm. A range of interest for correlation lengths can only be specified when the range of the design points is known. For this reason, in the remainder, all the design points are assumed to lie in the unit hypercube. The proposed correlation length prior is presented in section 4.2, while section 4.3 gives a brief review of the reference prior.

## 4.2 A prior density for correlation lengths

Gaussian process emulators are based on the assumption that the computer model outputs are correlated when their respective inputs are close in the Euclidean space they are defined. In other

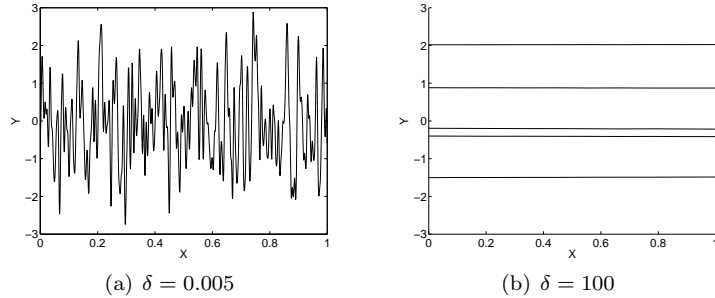


Figure 1: (a) a sample of a Gaussian process with  $\delta = 0.005$ , (b) 5 samples of a Gaussian process with  $\delta = 100$ . In both cases  $\sigma^2$  was equal to 1 and  $\beta$  was equal to 0.

words, we expect that the model output has some degree of smoothness, which in turn implies that it is unlikely to observe correlation lengths that are arbitrarily close to zero. On the other hand, as the values of the correlation lengths increase, there comes a point at which the response of the Gaussian process becomes entirely flat, and further increases in the correlation lengths have little or no effect. The above considerations indicate that there might be some boundaries in the correlation length values, that define a range of interest, within which the emulator’s correlation lengths are likely to fall. The definition of such a range is the main concept behind the development of the proposed prior.

Determining such correlation length boundaries is possible only after having specified the type of the correlation function that will be employed. In this work, we tune the prior to the Squared Exponential (SE) correlation function, but the same procedure could be applied for any of the other stationary correlation functions mentioned in section 2. The SE correlation function is given by

$$c(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p \exp \left\{ -\frac{(x_i - x'_i)^2}{\delta_i^2} \right\}. \quad (17)$$

Based on the above principles and on observations of Gaussian processes with inputs from the unit interval, we choose a correlation length interval with limits  $\delta_{lo} = 0.005$  and  $\delta_{hi} = 100$ . Samples of a Gaussian process with the above correlation lengths are shown in figure 1, which we believe represent adequate upper and lower limits for the variability of a computer model output.

Having defined the correlation length range of interest, we formulate a prior that is flat in the above range and decays rapidly outside this. The functional form of the proposed prior is

$$\ln(p(\delta)) \propto -2 \sum_{k=1}^p \left[ \left( \frac{\delta_k}{\delta_{lo}} \right)^{-2\alpha_{lo}} + \left( \frac{\delta_k}{\delta_{hi}} \right)^{2\alpha_{hi}} \right]. \quad (18)$$

In the above equation, the  $\delta_k/\delta_{lo}$  ratio controls the behaviour of the function for small correlation lengths for input  $k$  and the  $\delta_k/\delta_{hi}$  ratio controls the behaviour for large  $\delta$ . The  $\delta_{lo}$ ,  $\delta_{hi}$  parameters determine the value of  $\delta_k$  at which  $\ln(p(\delta))$  drops by two units in the direction of  $k$  from its overall maximum, which is zero. These points are known to define a range that contains approximately 95% of the probability mass. The rate of the drop is determined by the  $\alpha_{lo}$  and  $\alpha_{hi}$  parameters. The selection of values for the hyperparameters  $\alpha_{lo}$  and  $\alpha_{hi}$  is less critical. From experimentation we have found that setting  $\alpha_{lo} = \alpha_{hi} = 2$  is sufficient to ensure a flat response in the range of interest and an adequate drop outside this. However, if the range defined by  $\delta_{lo}$ ,  $\delta_{hi}$  is too narrow, the values of the  $\alpha$ ’s might need to increase so as to ensure the flatness of the prior. Although in

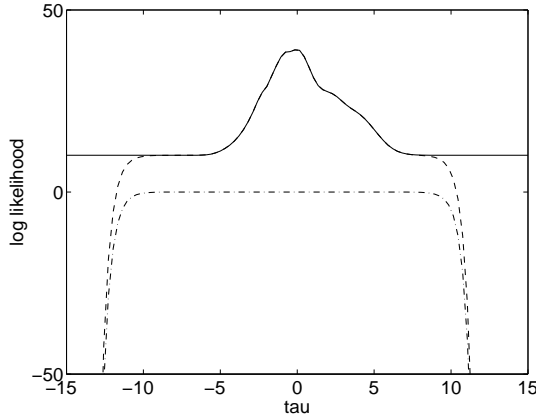


Figure 2: Plot of the log likelihood  $p(f(D)|\tau)$  (continuous line), the log prior  $\ln(g(\tau))$  (dash-dotted line) and their sum (dashed line) for  $n = 10$  points drawn from a GP with  $\delta = 1$ ,  $\sigma^2 = 1$  and  $\beta = 0$ .

the above formulation of the prior its hyperparameters are the same for all the  $p$  inputs, they can be chosen to vary with  $k$ , if we have different prior beliefs for the correlation lengths of different inputs.

The proposed prior can also be used with the optimisation algorithm used for finding modes of the posterior distribution, especially if we are interested in exploring a particular area of the parameter space. The prior can be written as a function of  $\tau$  as

$$\ln(g(\tau)) = -2 \sum_{k=1}^p \left[ e^{-a_{l_o}(\tau - \tau_{l_o})} + e^{a_{h_i}(\tau - \tau_{h_i})} \right] \quad (19)$$

where  $\tau = 2 \ln(\delta)$  and  $\tau_{l_o}$ ,  $\tau_{h_i}$  are similarly defined from  $\delta_{l_o}$ ,  $\delta_{h_i}$ . The above expression can be added to  $\ln(p(f(D)|\tau))$  to form the expression that is to be optimised. Note that the Jacobian of the transformation is not used, and the reason is that we want the modes with respect to  $\tau$  to coincide with those with respect to  $\delta$ .

Figure 2 shows the log likelihood  $\ln(p(f(D)|\tau))$ , the log of  $g(\tau)$  and their sum for  $n = 10$  design points drawn from a one dimensional Gaussian process, with  $\tau = 0$ . The parameters of the prior were set to  $\delta_{l_o} = 0.005$ ,  $\delta_{h_i} = 100$  and  $a_{h_i} = a_{l_o} = 2$ . Note that the posterior is identical to the likelihood for the range of interest, but it rapidly decays to minus infinity outside this.

The above hyperparameter values reflect our personal beliefs on what a likely range of correlation lengths for computer model emulators is. However, the proposed prior can be easily adjusted to reflect a different set of beliefs. For example, if there is evidence that the likely correlation lengths are outside the above range, this can be readily incorporated in the prior, by adjusting the values of  $\delta_{l_o}$  and  $\delta_{h_i}$ . Alternatively, if one is interested in drawing samples or searching for maxima from a narrower range of  $\delta$ , this can also be encoded in the above hyperparameters. Finally, the proposed prior has closed form derivatives, which are given in appendix B.

### 4.3 Reference prior

Another prior that yields a proper posterior distribution by effectively truncating the parameter space, is the reference prior. The reference prior was proposed by Berger et al. [6] for 1 dimension, and was extended to higher dimensions by Paulo [4]. It can be considered as a Jeffrey's prior on

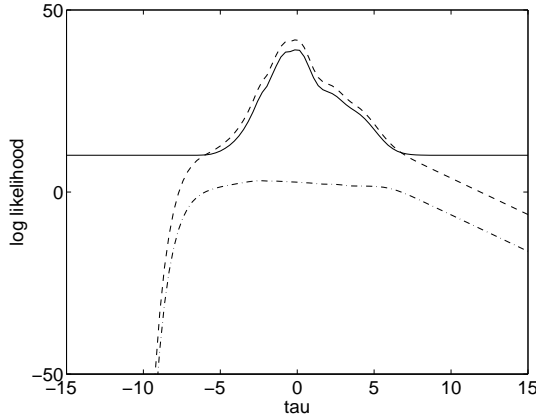


Figure 3: Plot of the log likelihood  $p(f(D)|\tau)$  (continuous line), the reference prior (dash-dotted line) and their sum (dashed line) for  $n = 10$  points drawn from a GP with  $\delta = 1$ ,  $\sigma^2 = 1$  and  $\beta = 0$ .

the marginalised likelihood  $p(f(D)|\sigma^2, \delta)$  and its derivation is based on the maximisation of the Kullback Leibler divergence between the prior and the posterior distribution. The reference prior is given by

$$p(\delta) \propto |\mathcal{I}(\delta)|^{1/2} \quad (20)$$

where

$$\mathcal{I} = \begin{bmatrix} n - q & \text{tr}W_1 & \text{tr}W_1 & \cdots & \text{tr}W_p \\ & \text{tr}W_1^2 & \text{tr}W_1W_2 & \cdots & \text{tr}W_1W_p \\ & & \ddots & \cdots & \vdots \\ & & & & \text{tr}W_p^2 \end{bmatrix}$$

and

$$W_k = \frac{\partial A}{\partial \delta_k} A^{-1} (I - H(H^T A^{-1} H)^{-1} H^T A^{-1}).$$

Given that the reference prior assumes knowledge of the design points correlation matrix  $A$ , it is capable of identifying regions of the parameter space in which most of the likelihood's mass is concentrated, truncating at the same time the remainder of the space, and therefore resulting in a proper posterior. However, it is computationally expensive and its cost increases rapidly with the input dimension  $p$ . Finally, it is not known to have closed form derivatives, which implies that it cannot be easily incorporated in a derivatives-based optimisation scheme, while the Hessian matrix  $H_{\delta}$ , which is often needed in setting up MCMC sampling algorithms, has to be calculated numerically.

Figure 3 shows the logarithms of the likelihood, the reference prior and the resulting posterior distribution, for the same 10 points used in figure 2. The reference prior is relatively flat in the range of interest, while it drops off as  $\delta$  takes very large or very small values, hence making the posterior distribution proper.

## 5 Drawing correlation length samples

Accounting for the uncertainty about the true values of the correlation lengths, according to the method presented in section 3.2.2, requires the generation of a number of correlation length samples.

A very common method for achieving this is to sample their posterior distribution using a Markov chain Monte Carlo (MCMC) algorithm. An alternative approximate, but very efficient method is based on a Gaussian approximation of the posterior distribution. These two methods are discussed in the following.

## 5.1 The Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) is a relatively straightforward algorithm for drawing samples from posterior distributions, which can also be high dimensional. The posterior of  $\delta$  can be formed by multiplying the likelihood (eq. 11) with one of the prior distributions of section 4 as given in equations 18 and 20. Using a uniform prior, which is equivalent to sampling directly from 11, is not recommended, as the implied posterior is not proper and can affect the convergence of the MH algorithm.

The first step of the MH algorithm is to find a starting point for the Markov chain. A good starting point is the value of  $\delta$  that maximises  $p(\delta|f(D))$ , which we call  $\hat{\delta}$ . The second step is to define a proposal distribution, that will determine the step size of the chain's random walk. A sensible choice is to make the proposal distribution proportional to the width of the main mode, as suggested in [4]. This can be achieved by first finding the Hessian matrix  $H_{\hat{\delta}}$  at  $\hat{\delta}$ . This matrix can be calculated with eq. (22), substituting the derivatives of  $A$  w.r.t.  $\tau$  with those w.r.t.  $\delta$ , as given in eqs. (27 - 29). If the prior of section 4.2 is used, its second derivatives should also be added, as given by eq. 31. If the reference prior is used on the other hand, the Hessian must be calculated numerically, as an analytical form is not available. Once the Hessian matrix has been calculated, an estimate of the main mode's variance is given by  $-H_{\hat{\delta}}^{-1}$ . We define the variance of the proposal distribution as  $V_{\hat{\delta}} = -c^2 H_{\hat{\delta}}^{-1}$ , where the scalar  $c$  is a factor that controls the convergence rate of the algorithm. Following the advice in [14], we set  $c = 2.4/\sqrt{p}$ , but this can be modified if the mixing is not deemed satisfactory. The Metropolis-Hastings algorithm can then be described in the following steps:

1. Set  $\delta^{(1)}$  equal to  $\hat{\delta}$
2. Add to  $\delta^{(i)}$  a normal variate drawn from  $\mathcal{N}(0, V_{\hat{\delta}})$ , call the result  $\delta^*$
3. Calculate the ratio  $\alpha = \frac{p(\delta^*|f(D))}{p(\delta^{(i)}|f(D))}$
4. Set  $\delta^{(i+1)} = \delta^*$  with probability  $\alpha$  and  $\delta^{(i+1)} = \delta^{(i)}$  with probability  $1 - \alpha$
5. Repeat steps 2-4 until  $M$  samples have been drawn

Despite being straightforward in its implementation, the Metropolis Hastings algorithm can have poor mixing properties, especially if the posterior distribution is wide in one dimension but narrow in another. If the performance of Metropolis Hastings is not deemed adequate, more efficient alternatives can be used, such as the Hamiltonian Monte Carlo [14], where the gradient of the posterior is used to improve convergence.

## 5.2 Gaussian approximation of the posterior

An alternative method to the Metropolis-Hastings algorithm is given by the approximation of the posterior distribution  $p(\tau|f(D))$  using a multivariate Gaussian distribution [11]. The posterior distribution of  $\tau$  is used instead of the posterior of  $\delta$ , because the former is reported to be closer to a Gaussian distribution than the latter [11]. The posterior of  $\tau$  can be obtained from the likelihood

$p(f(D)|\tau)$  using a uniform prior on  $\tau$ , or any other preferred prior. For simplicity, in the following we assume that  $p(\tau) \propto \text{const}$ , therefore  $p(\tau|f(D)) \propto (f(D)|\tau)$ .

The approximating Gaussian distribution is completely specified by its mean and variance matrix. Its mean is the vector  $\hat{\tau}$ , which was defined in eq. 14 as the vector that maximises the likelihood  $p(f(D)|\tau)$ . The variance is calculated from the Hessian of the log likelihood of  $\tau$  as this is given in equation 22, and evaluated at  $\hat{\tau}$ . If we denote this matrix as  $H_{\hat{\tau}}$ , the variance of the approximating equation is given by  $V_{\hat{\tau}} = -H_{\hat{\tau}}^{-1}$ . The Gaussian approximation method (GA) consists then of two steps:

1. Independently draw  $M$  samples  $\tau^i \sim \mathcal{N}(\hat{\tau}, V_{\hat{\tau}})$ , for  $i = [1..M]$
2. Obtain the correlation length samples by the transformation  $\delta^i = \exp(\tau^i/2)$

The main benefit of this method, is that the samples are independently drawn from a multivariate normal distribution, which implies that the sampling is very efficient. On the other hand, there is no guarantee that the normal distribution is an accurate representation of the posterior distribution  $p(\tau|f(D))$ , and the results may be suboptimal. In the majority of the cases considered however, the approximation was sufficiently accurate.

There is also a more subtle point that needs consideration before applying this method. If the likelihood is very flat with respect to one input, then the respective element on the diagonal of the Hessian  $H_{\hat{\tau}}$  will be close to zero, and the variance of the drawn samples will be unrealistically large. The exponential transformation of the step 2 above, can subsequently yield samples for a single input that can have an unrealistically large range (e.g.  $10^{-6} - 10^{12}$ ). This does not imply that the posterior is flat over the entire range, but it is an indication of a poor approximation of the mode by its Hessian. It is therefore strongly recommended, especially in high dimensional problems, that the variance of the samples drawn with the Gaussian approximation method is checked before these are used for making inferences. A potential remedy for this condition could be fixing the value of the particular correlation lengths to the value of the mode.

## 6 Results

The evaluation of the various parameter estimation approaches is presented in this section. We start by examining the coverage probabilities of the Bayesian credible intervals for each of the three posterior sampling methods, and then test the parameter estimation methods by building emulators for two different climate models.

### 6.1 Frequentist coverage

A method for comparing different correlation length prior distributions is to examine the coverage probabilities of credible intervals of the posterior distribution of  $\delta$ . Following the method described in [12], we repeatedly draw samples from a Gaussian process, with known parameters, and calculate the number of times that the true values of the correlation lengths fall in the 95% equi-tailed interval of the posterior distribution of  $\delta$ . The closer this number is to 0.95, the better the prior is assumed to be.

In the simulations that follow,  $n$  points were drawn from a Gaussian process with fixed parameters, which were then used to form the posterior of  $\delta$ , with the reference and the proposed priors. 10500

samples were drawn from the resulting posteriors, the 500 first of which were discarded as burn in. The posterior samples were then used to evaluate the frequentist coverage as described in the previous paragraph. The Gaussian approximation method was assessed in the same way, even though its samples are not drawn from the exact posterior. The whole procedure was repeated 3000 times. The parameters of the Gaussian process were set to  $\sigma^2 = 1$  and  $h(\mathbf{x}) = [1, \mathbf{x}]^T$ . The values of the two other parameters ( $\beta, \delta$ ) are given in table 1, together with the results of the simulation experiment.

The results suggest that the proposed prior consistently provides coverage that is at least as good as that of the reference prior. The only exception is found for  $p = 5$  and small  $\delta$ , a case that is rather problematic, as it is indicated by the small coverage of all the three methods. The most likely reason, is that the sample size  $n$  is likely to be small for this combination of small correlation lengths ( $\delta < 0.5$ ) and moderate input dimension ( $p = 5$ ) and the drawn samples did not represent accurately the posterior distribution. An interesting observation is that the coverage of the Gaussian approximation method seems to improve with the increase in dimension, a finding that was also observed in [11]. This could be due to the fact that the posterior distribution is better approximated by a Gaussian in high dimensional spaces, and/or because the employed posterior sampling methods were not capable of exploring more intricate patterns in the posterior as the dimensionality of the input increased.

		$\delta$	GA	RF	PR	$n$
$p = 2$	$\beta = 0$	{0.1, 0.3}	88.7	91.7	93.5	30
		{0.1, 1}	92.0	93.3	93.6	30
		{1, 1.5}	93.0	95.1	94.5	30
	$\beta = 1$	{0.1, 0.3}	89.2	91.9	93.3	30
		{0.1, 1}	93.4	93.7	94.3	30
		{1, 1.5}	93.5	94.8	94.7	30
$p = 5$	$\beta = 0$	{0.2, 0.3, 0.4, 0.3, 0.5}	87.6	86.9	85.9	100
		{1, 0.5, 2, 0.3, 1.5}	94.3	93.1	93.6	100
		{1, 1.2, 1.5, 1.8, 2}	95.2	94.2	94.4	100
	$\beta = 1$	{0.2, 0.3, 0.4, 0.3, 0.5}	86.9	86.3	84.5	100
		{1, 0.5, 2, 0.3, 1.5}	94.0	92.6	93.2	100
		{1, 1.2, 1.5, 1.8, 2}	94.4	93.7	93.7	100

Table 1: Coverage probability of the 95% intervals of the posterior distribution of  $\delta$ , for the Gaussian approximation method (GA), the reference prior (RF) and the proposed prior (PR).

## 6.2 Surfefbm model emulator

In this section we use the parameter estimation methods described previously to build an emulator for the *surfefbm* model. This is a very simple planetary climate model, which nevertheless has distinct end states and nonlinear behaviour. We vary 2 out of its 8 inputs, the solar constant and the albedo, which represents the fraction of the solar energy reflected from the surface of the planet. The output we measure is the mean surface temperature. The emulator is built using 30 training points and a further 10 points are used for testing.

The training and testing points were initially drawn from a Latin Hypercube (LHC) in the unit interval. The points were then linearly mapped to the ranges [1370-1420] for the solar constant and

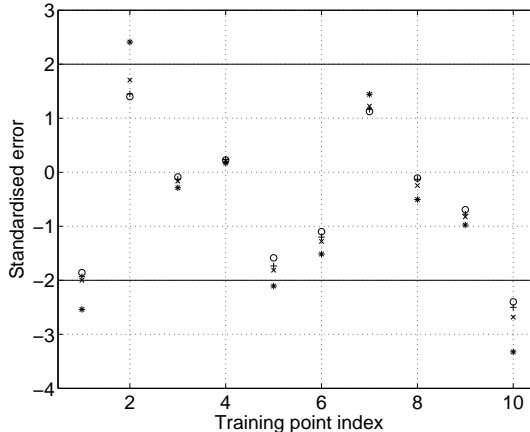


Figure 4: Standardised errors for the testing points, using the 4 sampling methods: (\*) ML, (x) GA, (+) RF, (o) PR.

[0.2-0.4] for albedo in order to run the simulator. In building and running the emulator however, the input points were mapped in the original unit interval, to ensure that the resulting correlation lengths have values in the range that the proposed prior was tuned for.

The likelihood mode was located at  $\hat{\delta} = [0.31, 0.16]$ . The posterior obtained using the proposed prior had its mode in the same location, because in this value range the prior is very flat. The mode of the posterior that resulted using the reference prior was virtually identical to that of the likelihood, with its mode being less than 0.05 away from  $\hat{\delta}$ .

In the next step, we drew 10000 samples from the posteriors obtained using the reference and the proposed prior, as well as 10000 samples using the Gaussian approximation method. The value of the output at the testing points was then predicted using the four different methods, ML (maximum likelihood), GA (Gaussian approximation), RF (reference prior) and PR (proposed prior). We evaluate the predictions using the Mahalanobis distance and the standardised errors [15]. The mean of the Mahalanobis distance is equal to the number of validation points, i.e. 10, and the expected standard deviation<sup>1</sup> is 5.5.

ML	GA	RF	PR
36.5	22.2	16.1	12.4

Table 2: Mahalanobis distances for the different parameter estimation methods and the surfeb model runs

Table 2 shows the Mahalanobis distances obtained with the 4 different methods. We see that using a single estimate of  $\delta$  the Mahalanobis distance is approximately 5 standard deviations away from its theoretical mean. This implies that the emulator is overconfident about its predictions, which can be seen as the result of not taking into account the uncertainty about the true value of the correlation lengths. On the other hand, the three other methods that account for this uncertainty, yield lower Mahalanobis distances, which reflect the increased uncertainty bounds of the emulator. Figure 4 shows the standardised errors for the four methods. It is clear that the ML method results in the higher standardised errors, because its uncertainty bounds are the smaller.

<sup>1</sup>The formula for the expected variance is  $\frac{2n'(n'+n-q-2)}{n-q-4}$ , with  $n'$  being the number of training points.

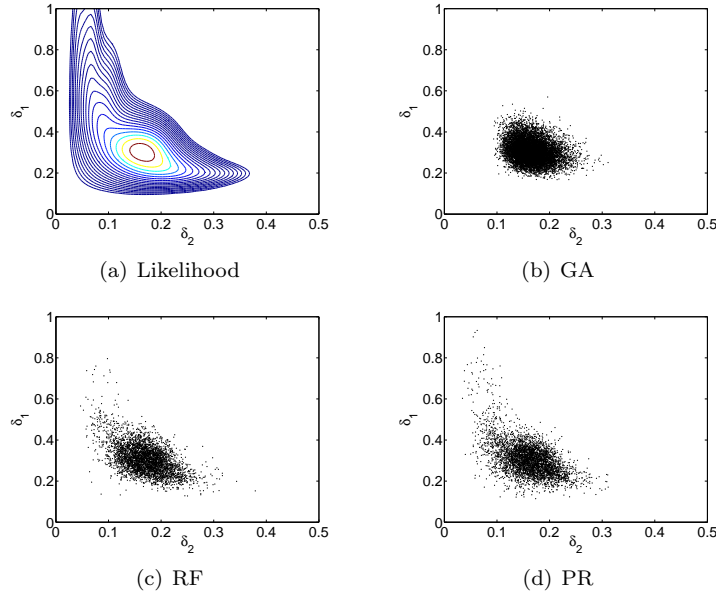


Figure 5: A contour plot of the likelihood and a scatter plot of the samples drawn using the different methods.

If we compare the three methods that account for the uncertainty in  $\delta$ , we see that the Mahalanobis distance for PR is within 1 standard deviation from the mean, and for the RF is just out. For the GA method however, it is almost 2.5 standard deviations larger than the mean. The reason behind this is that the Gaussian approximation method does not capture the shape of the likelihood as accurately as the other two methods. Figure 5 shows a contour plot of the likelihood surface and scatter plots of the samples drawn with the three methods. The PR and RF methods reproduce more precisely the wedged shape of the likelihood, whereas the GA method captures mainly its central peak.

### 6.3 C-Goldstein model emulator

In this section, we build an emulator for runs taken from a more complex climate model, the C-Goldstein [16]. This is an intermediate complexity model, which features a fully 3 dimensional ocean and the ability to model global trends, while being at the same time significantly faster than its high resolution counterparts. We vary 18 of the model’s inputs and observe one output, the mean atmospheric temperature. The training points are drawn from a 255 point Sobol sequence and the 97 testing points are drawn from a Latin Hypercube. Both the training and testing points lie in the  $[0,1]$  interval, and these values are used for building and testing the emulator. In order to run the model and obtain the output values, the training and testing points are linearly mapped on the input parameter ranges suggested in [17].

Table 3 shows the correlation lengths that maximise the likelihood (ML) and a number of different posteriors. Inspection of columns titled ‘ML’ in table 3, reveals 5 small correlation lengths ( $\delta < 4$ ), 5 medium ( $4 < \delta < 20$ ) and 8 that take extremely large values, considering that the input points lie in the unit hypercube. The columns titled ‘PR<sub>100</sub>’ show the mode of the posterior obtained with the proposed prior, using  $\delta_{hi} = 100$ . The small and medium correlation lengths are virtually unaffected by the application of the prior. The very large correlation lengths on the other hand,

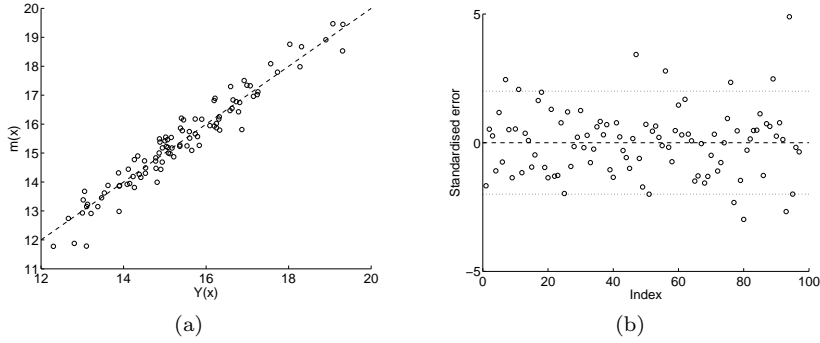


Figure 6: (a) plot of validation points against the ML emulators posterior mean (b) standardised errors for the ML emulator.

are limited to values smaller than 100. Decreasing the prior’s upper limit to  $\delta_{hi} = 30$ , reduces the value of the large correlation lengths further, and changes slightly the medium ones. The impact on the small  $\delta$ ’s is again minimal. The reference prior (‘RF’ columns), imposes an upper limit equal to that of  $\text{PR}_{30}$ , but at the same time has a larger effect on the small correlation lengths.

input	ML	$\text{PR}_{100}$	$\text{PR}_{30}$	RF	input	ML	$\text{PR}_{100}$	$\text{PR}_{30}$	RF
6	0.19	0.19	0.19	0.23	2	3954	32.25	15.82	20.67
1	0.20	0.20	0.20	0.21	15	6780	32.47	14.56	12.20
14	0.91	0.91	0.89	1.09	16	$9.1 \cdot 10^4$	37.79	16.01	14.42
8	3.61	3.66	3.94	5.49	3	$9.6 \cdot 10^4$	38.14	16.87	17.96
5	3.79	3.77	3.66	4.46	18	$1.7 \cdot 10^5$	39.37	17.13	14.99
11	6.19	6.12	6.07	7.69	9	$1.8 \cdot 10^5$	39.22	16.40	18.21
17	8.73	9.40	11.30	14.55	7	$2.6 \cdot 10^5$	40.01	17.06	17.92
10	12.35	12.46	11.50	10.11	4	$1.2 \cdot 10^6$	42.58	18.07	20.34
12	13.91	12.35	9.10	6.85	Mahalanobis Distance				
13	18.62	18.28	14.46	14.07		187	164	122	137

Table 3: Correlation lengths that maximised the likelihood (ML), the posteriors obtained using the proposed prior with  $\delta_{hi} = 100$ , ( $\text{PR}_{100}$ ), with  $\delta_{hi} = 30$ , ( $\text{PR}_{30}$ ) and the reference prior (RF). The last row shows the Mahalanobis distances for the resulting emulators.

Next, we investigate the effect of the different priors on the inferences made by the emulator. We start by considering the ML solution as a reference, and compare the solutions provided by the posterior modes against it. Figure 6(a) shows the validation points plotted against the predictions of the emulator built using the ML mode. Figure 6(b) shows the standardised errors. These figures indicate that the emulator does a reasonable job, although it appears to be somewhat overconfident, as judged by the number standardised errors outside the  $[-2, 2]$  interval. The estimated Mahalanobis distance of 187 (table 3), with a theoretical mean and standard deviation of 97 and 16.6 respectively, also supports this observation.

Figures 7(a,b) show the differences in the posterior mean and variance between the emulators obtained using the different posterior modes (henceforth called  $\text{PR}_{100}$ ,  $\text{PR}_{30}$  and RF) and the maximum likelihood emulator (ML). The proposed posterior with  $\delta_{hi} = 100$  has a rather negligible effect on the emulator: the root mean square (rms) difference between the posterior means is approximately 0.02 and the variance is slightly inflated, having an rms difference of 0.006. As a

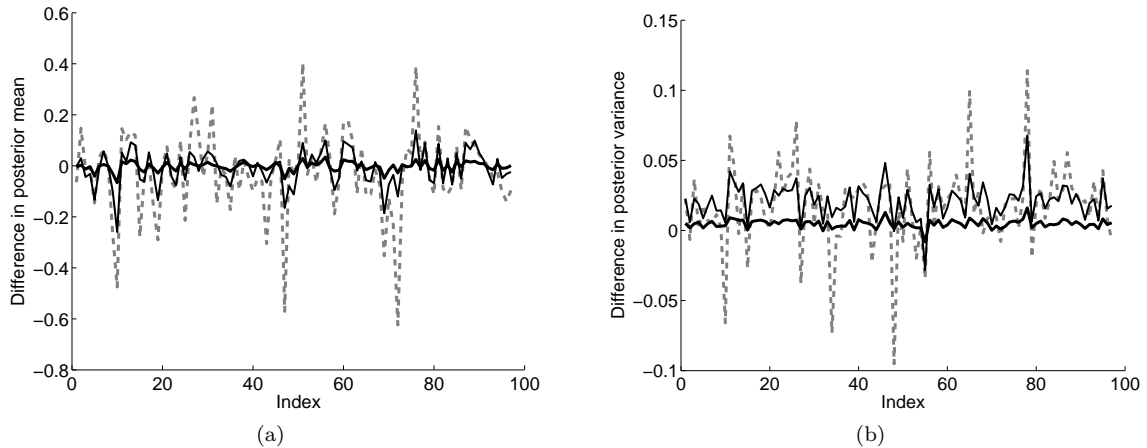


Figure 7: Differences in the posterior mean and variance between  $\text{PR}_{100}$  and ML (thick line),  $\text{PR}_{30}$  and ML (thin line) and RF and ML (dashed line).

result the Mahalanobis distance drops to 164. Making the prior ‘tighter’, by decreasing  $\delta_{hi}$  to 30, alters slightly more the posterior mean, increasing its rms difference to 0.07 and inflates further the variance, yielding an rms difference of 0.02. Note that the posterior variance of  $\text{PR}_{30}$  is almost constantly larger than the posterior variance of ML, as indicated by the positive values of the thin line in figure 7(b). This causes the Mahalanobis distance to drop further to 122, making the emulator only marginally invalid. The reference prior on the other hand, alters more substantially the resulting emulator. The rms difference of the posterior means is approximately 0.16, and the rms difference of the variance is around 0.03. What is more important perhaps, is that the posterior variance is not consistently larger than the variance of the ML emulator, as it can be seen by the positive and negative values of the dashed line in figure 7(b).

We now examine the effect the priors have on drawing samples from the posterior, and how this ultimately affects the emulator. A first observation is that the restrictions imposed by the proposed prior simplify the exploration of the parameter space, and as a result ease the convergence of the MCMC algorithm. Figures 8, 9 show the samples drawn with the reference and the proposed priors. The paths in figure 9, look significantly more stationary, mainly because the proposed prior is more effective in limiting the parameter space, hence, making its exploration less cumbersome.

Apart from convergence issues, one might also argue that exploring parameter space regions with large correlation length values is not an effective way of using the available computational power. For example, if the likelihood is very flat in one dimension, as  $\delta \rightarrow \infty$  drawing very large samples will have little or no effect on the emulator. On the other hand, smaller values of  $\delta$  can have a greater impact, therefore, exploration of such areas can be much more informative. This intuition manifested itself in our example, where after marginalising  $\delta$  using the posterior samples, the emulator that used the reference prior had a Mahalanobis distance of 135.2, while the emulator that used the proposed prior had a Mahalanobis distance of 96.6, almost identical to the theoretical mean of 97.

Figure 10, shows the samples drawn with the Gaussian approximation of the posterior mode obtained with  $\text{PR}_{30}$ . The distribution of the samples for inputs with small correlation lengths (e.g. 1,5,6,8,14), is similar to that of the samples drawn with the  $\text{PR}_{30}$  prior. The range of the remaining

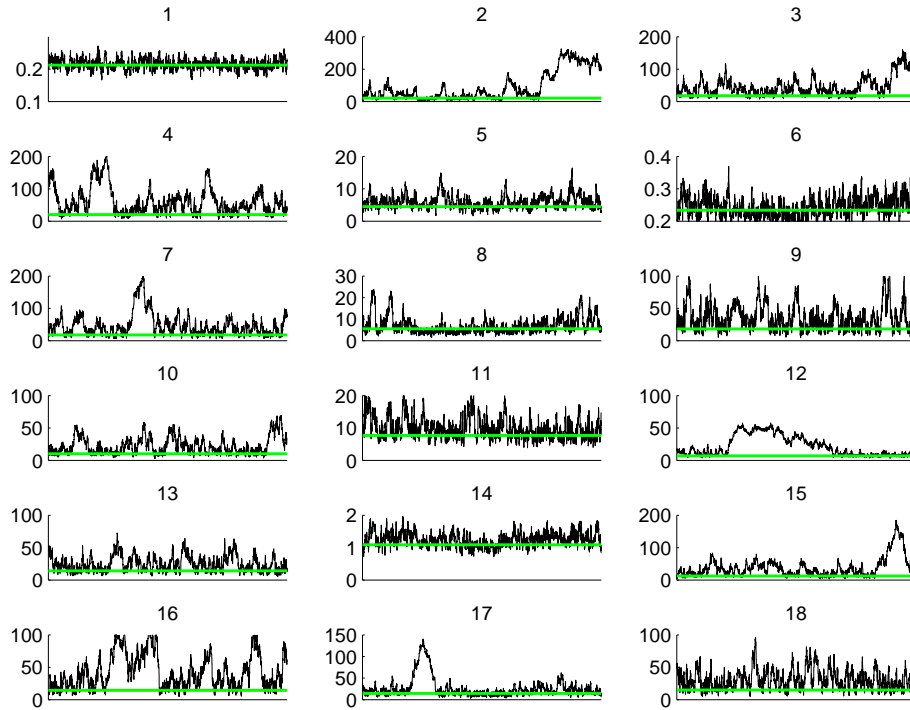


Figure 8: Posterior samples of  $\delta$  drawn using the Reference prior. Green line shows the value at the mode.

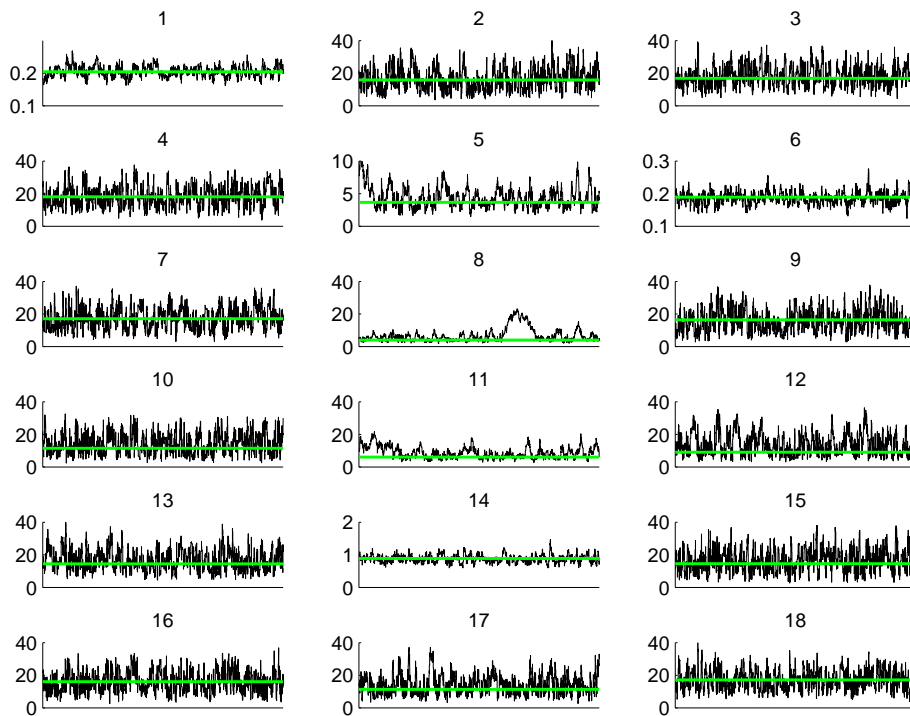


Figure 9: Posterior samples of  $\delta$  drawn using the Proposed prior with  $\delta_{hi} = 30$ . Green line shows the value at the mode.

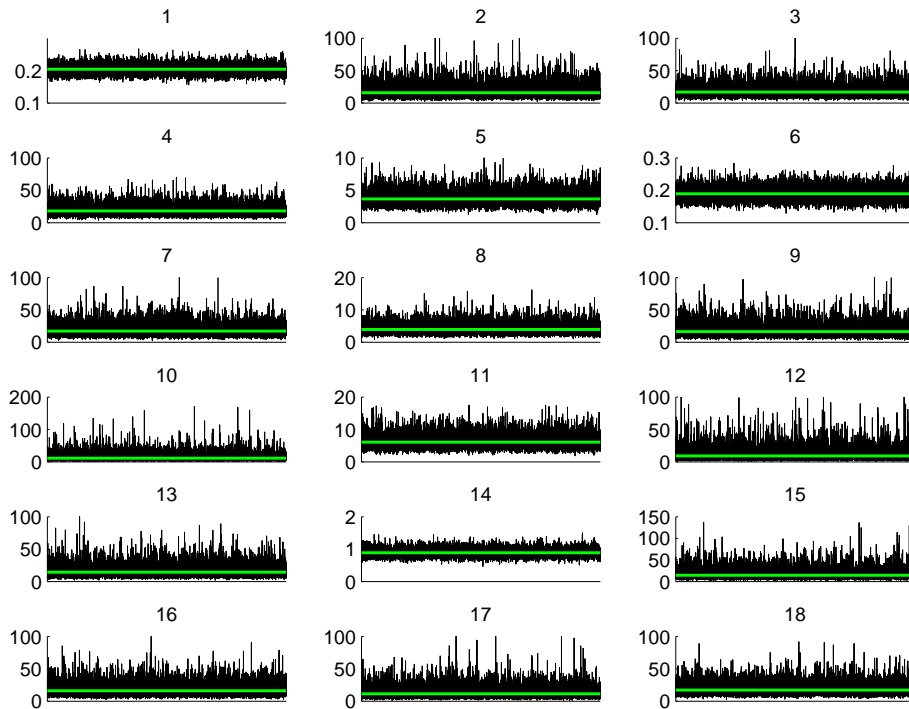
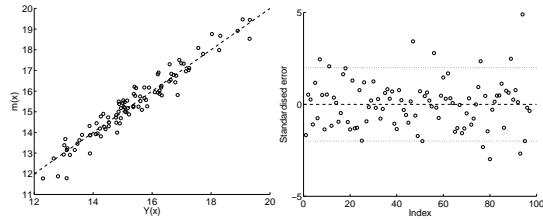


Figure 10: Posterior samples of  $\delta$  drawn using the Gaussian approximation method. Green line shows the value at the mode.

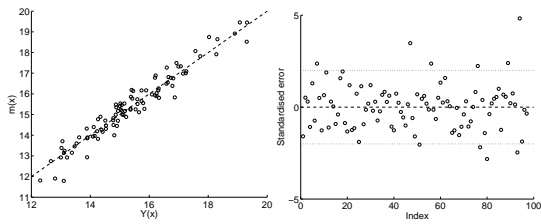
samples is generally larger than the corresponding ranges of the samples drawn with  $\text{PR}_{30}$ . Once again, the larger sampled values do not affect the emulator much, but the smaller ones, inflate the posterior variance further, and yield a Mahalanobis distance of 90.1.

We should mention that without the application of the proposed prior, the Gaussian approximation method was either not feasible or did not result in a valid emulator. A direct approximation of the likelihood mode resulted in a non invertible Hessian matrix  $H_{\bar{\tau}}^{-1}$  for some inputs, while for some others the curvature at the mode was so flat, that the drawn samples were spanning the almost the entire range of double precision numbers (due to the logarithmic transformation on  $\tau$ ). If one chose to fix the problematic inputs, then the remaining samples could not increase the posterior variance enough so as to result in a valid emulator.

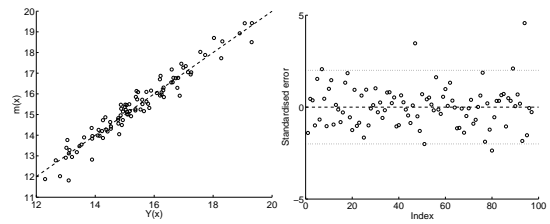
Figure 11 shows diagnostics for all the emulators built in this section and summarises the main findings. First, the application of the  $\text{PR}_{100}$  prior, has a negligible effect on the emulator apart from a slight increase in the posterior variance, as it can be seen by the similarity of panels (a) and (b). Decreasing the prior's upper limit mainly inflates the posterior variance, therefore resulting in smaller standardised errors and a drop in the Mahalanobis distance (panel (c)). Marginalisation of the correlation lengths using the samples drawn with the  $\text{PR}_{30}$  prior, increases the posterior variance enough so as to yield a valid emulator (panel (d)). The success of the prior is that it avoids sampling very large correlation length values, and focusses the MCMC exploration to more informative areas of the parameter space. The Gaussian approximation method (panel (e)) overestimates somewhat the ranges of  $\delta$  for some inputs, and increases the posterior variance even further, although it still yields a valid emulator in this case. Panel (f) shows that the using the RF mode results in an emulator that is less similar to the ML emulator, than any of the  $\text{PR}_{100}$  or  $\text{PR}_{30}$  are. Finally, marginalisation using the samples drawn with the RF prior (panel (g)), decreased somewhat the standardised errors, but not sufficiently so as to make the emulator valid.



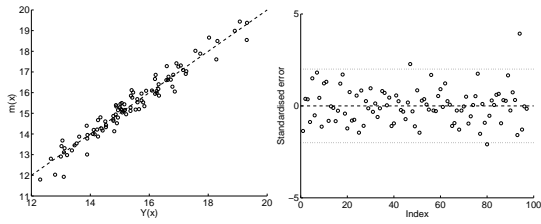
(a) Likelihood mode (M.D.=187)



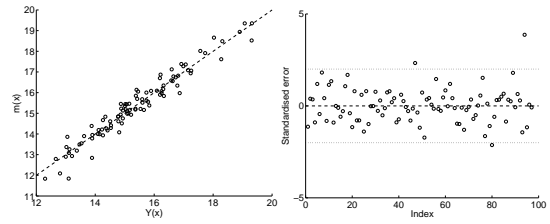
(b) PR<sub>100</sub> mode (M.D.=164)



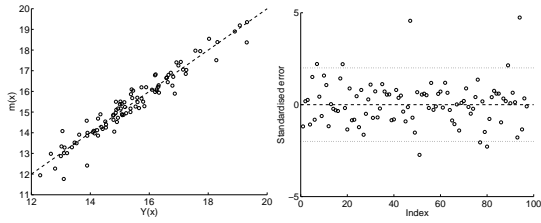
(c) PR<sub>30</sub> mode (M.D.=122)



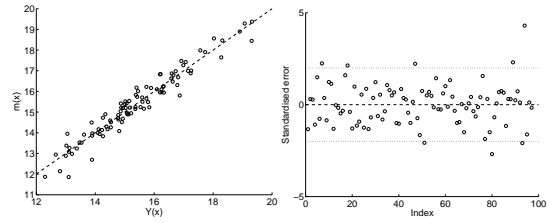
(d) PR<sub>30</sub> mcmc (M.D.=96.6)



(e) Gaussian Approximation (M.D.=90.1)



(f) RF mode (M.D.=137)



(g) RF mcmc (M.D.=135)

Figure 11: Diagnostics for different emulators

## 7 Conclusion

This work addressed the problem of parameter estimation for computer model emulators based on Gaussian processes. Three parameters were used in the formulation of the Gaussian process, the regression coefficients  $\beta$ , the scaling parameter  $\sigma^2$  and the correlation lengths  $\delta$ . Two approaches were considered for treating the above parameters. The first was to obtain an estimate and treat this as if it were the true value, and the second was to marginalise the parameters so as to account for the uncertainty about their true value.

With the assumption of the linear form for the mean function, both  $\beta$  and  $\sigma^2$  could be marginalised analytically. We advocate this approach because apart from formally accounting for the parameter uncertainty, the resulting marginal likelihood is reported to yield more stable estimates for the correlation lengths [4]. Given that the analytical marginalisation of the correlation lengths is not tractable, we considered both the plug in and the numerical marginalisation approaches. The latter approach results in a larger predictive variance, which can yield a valid emulator when fewer model runs are available, or when the simulator is not adequately modelled by the Gaussian process. On the other hand, if the available runs can yield a valid emulator using plug in estimates, this approach might be preferred because of its reduced computational load.

One of the main contributions of this work was the proposal of a correlation length prior that a) results in a proper posterior, b) facilitates the application of MCMC algorithms and c) offers the possibility of focussing the search for posterior modes in specified regions of the parameter space. The proposed prior was compared with the reference prior in terms of the frequentist coverage of Bayesian intervals and was found to provide at least as good coverage as its more computationally expensive counterpart.

As the proposed prior is based on the definition of a range of interest for the correlation lengths, we also investigated the effect of imposing such constraints on the parameter space. We saw that it is possible to impose an upper limit to the values of the correlation lengths without essentially altering the emulator's predictions. Lowering this limit inflates the posterior variance, which can be useful in case of overconfident emulators. We also saw that restricting the parameter space by means of the proposed prior, aided the convergence of the MCMC algorithm, and perhaps more importantly, discouraged it from spending its computational effort in less informative regions of the parameter space. Finally, the proposed prior resulted in posterior modes that could be easily approximated by the Gaussian approximation method, which proved to be an efficient way of drawing posterior samples, especially in high dimensions.

## A Derivatives of the log likelihood

The following equations describe the two first derivatives of  $\ln(p(f(D)|\tau))$ . Obtaining the derivatives with respect to  $\delta$  is a straightforward extension that will be discussed right after. The first derivative with respect to  $\tau$  is a  $p \times 1$  vector, whose  $k^{\text{th}}$  element is

$$\frac{\partial(\ln(p(f(D)|\tau))}{\partial\tau_k} = -\frac{1-n+q}{2}\text{tr}\left[P\frac{\partial A}{\partial\tau_k}\right] - \frac{n-q}{2}\text{tr}\left[R\frac{\partial A}{\partial\tau_k}\right]. \quad (21)$$

The second derivative is a  $p \times p$  matrix with elements

$$\frac{\partial^2(\ln(p(f(D)|\tau))}{\partial\tau_l\partial\tau_k} = -\frac{1-n+q}{2}\text{tr}\left[P\frac{\partial^2 A}{\partial\tau_l\partial\tau_k} - P\frac{\partial A}{\partial\tau_l}P\frac{\partial A}{\partial\tau_k}\right] - \frac{n-q}{2}\text{tr}\left[R\frac{\partial^2 A}{\partial\tau_l\partial\tau_k} - R\frac{\partial A}{\partial\tau_l}R\frac{\partial A}{\partial\tau_k}\right]. \quad (22)$$

In the above equations the matrices  $P$  and  $R$  are defined as

$$P \equiv A^{-1} - A^{-1}H(H^T A^{-1}H)^{-1}H^T A^{-1}$$

and

$$R \equiv P - Pf(D)(f(D)^T Pf(D))^{-1}f(D)^T P.$$

Finding the partial derivatives of the correlation matrix  $A$  with respect to  $\tau$ , requires the specification of a correlation function. We provide here the derivatives for the squared exponential correlation function given in eq. 17. We first denote the  $(\mu, \nu)^{\text{th}}$  element of  $A$  by

$$(A)_{(\mu, \nu)} = c(\mathbf{x}_\mu, \mathbf{x}_\nu) = \prod_{i \in p} \exp \left\{ -\frac{(x_{i, \mu} - x_{i, \nu})^2}{e^{\tau_i}} \right\} \quad (23)$$

where the first of the two subscripts of  $x_{i, \mu}$  indexes the input, and the second indexes the design point. The two first derivatives of  $A$  are  $(n \times n)$  matrices, whose  $(\mu, \nu)^{\text{th}}$  element is given by

$$\begin{aligned} \left( \frac{\partial A}{\partial \tau_k} \right)_{(\mu, \nu)} &= \prod_{i \in p} \exp \{ -(x_{i, \mu} - x_{i, \nu})^2 e^{-\tau_i} \} [(x_{k, \mu} - x_{k, \nu})^2 e^{-\tau_k}] \\ &= (A)_{(\mu, \nu)} \frac{(x_{k, \mu} - x_{k, \nu})^2}{e^{\tau_k}} \end{aligned} \quad (24)$$

and

$$\begin{aligned} \left( \frac{\partial^2 A}{\partial \tau_k^2} \right)_{(\mu, \nu)} &= \prod_{i \in p} \exp \{ -(x_{i, \mu} - x_{i, \nu})^2 e^{-\tau_i} \} [(x_{k, \mu} - x_{k, \nu})^2 e^{-\tau_k}] [(x_{k, \mu} - x_{k, \nu})^2 e^{-\tau_k} - 1] \\ &= (A)_{(\mu, \nu)} \left[ \left( \frac{(x_{k, \mu} - x_{k, \nu})^2}{e^{\tau_k}} \right)^2 - \frac{(x_{k, \mu} - x_{k, \nu})^2}{e^{\tau_k}} \right]. \end{aligned} \quad (25)$$

The second cross partial derivative is

$$\begin{aligned} \left( \frac{\partial^2 A}{\partial \tau_l \partial \tau_k} \right)_{(\mu, \nu)} &= \prod_{i \in p} \exp \{ -(x_{i, \mu} - x_{i, \nu})^2 e^{-\tau_i} \} [(x_{l, \mu} - x_{l, \nu})^2 e^{-\tau_l}] [(x_{k, \mu} - x_{k, \nu})^2 e^{-\tau_k}] \\ &= (A)_{(\mu, \nu)} \frac{(x_{l, \mu} - x_{l, \nu})^2}{e^{\tau_l}} \frac{(x_{k, \mu} - x_{k, \nu})^2}{e^{\tau_k}}. \end{aligned} \quad (26)$$

The derivatives of the likelihood function with respect to  $\delta$  can be found by substituting the partial derivatives of  $A$  w.r.t.  $\tau$  in eqs. 21, 22 with the following partial derivatives:

$$\left( \frac{\partial A}{\partial \delta_k} \right)_{(\mu, \nu)} = (A)_{(\mu, \nu)} \left[ \frac{2(x_{k, \mu} - x_{k, \nu})^2}{\delta_k^3} \right] \quad (27)$$

$$\left( \frac{\partial^2 A}{\partial \delta_k^2} \right)_{(\mu, \nu)} = (A)_{(\mu, \nu)} \frac{(x_{k, \mu} - x_{k, \nu})^2}{\delta_k^4} \left[ \frac{4(x_{k, \mu} - x_{k, \nu})^2}{\delta_k^2} - 6 \right] \quad (28)$$

and

$$\left( \frac{\partial^2 A}{\partial \delta_l \partial \delta_k} \right)_{(\mu, \nu)} = (A)_{(\mu, \nu)} \left[ \frac{2(x_{l, \mu} - x_{l, \nu})^2}{\delta_l^3} \right] \left[ \frac{2(x_{k, \mu} - x_{k, \nu})^2}{\delta_k^3} \right]. \quad (29)$$

## B Derivatives of the proposed prior

In this section we provide the derivatives of the prior proposed in section 4.2 with respect to  $\delta$  and  $\tau$ . The first derivative w.r.t  $\delta$  is a  $p \times 1$  vector, whose  $k^{\text{th}}$  element is

$$\frac{\partial \ln p(\delta)}{\partial \delta_k} = \frac{4\alpha_{lo}}{\delta_{lo}} \left( \frac{\delta_k}{\delta_{lo}} \right)^{-2\alpha_{lo}-1} - \frac{4\alpha_{hi}}{\delta_{hi}} \left( \frac{\delta_k}{\delta_{hi}} \right)^{2\alpha_{hi}-1}. \quad (30)$$

The second derivative is a diagonal  $k \times k$  matrix, whose  $k^{\text{th}}$  diagonal element is

$$\frac{\partial^2 \ln p(\delta)}{\partial \delta_k^2} = -\frac{4\alpha_{lo}(2\alpha_{lo} + 1)}{\delta_{lo}^2} \left( \frac{\delta_k}{\delta_{lo}} \right)^{-2\alpha_{lo}-2} - \frac{4\alpha_{hi}(2\alpha_{hi} - 1)}{\delta_{hi}^2} \left( \frac{\delta_k}{\delta_{hi}} \right)^{2\alpha_{hi}-2}. \quad (31)$$

The derivatives w.r.t.  $\tau$  are

$$\frac{d \ln g(\tau)}{d\tau_k} = 2a_{lo} \exp[-a_{lo}(\tau_k - \tau_{lo})] - 2a_{hi} \exp[a_{hi}(\tau_k - \tau_{hi})] \quad (32)$$

and

$$\frac{d^2 \ln g(\tau)}{d\tau_k^2} = -2a_{lo}^2 \exp[-a_{lo}(\tau_k - \tau_{lo})] - 2a_{hi}^2 \exp[a_{hi}(\tau_k - \tau_{hi})]. \quad (33)$$

## C Computational issues

The implementation of the estimation and prediction formulae described in this work is prone to numerical problems that arise from rounding errors. The main reason is that these computations involve the inversion of matrices that are often badly conditioned. In this section we describe some methods for sidestepping such issues.

A common problem is that the design points correlation matrix is not positive definite, and therefore non invertible. This can occur when the correlation lengths are very large compared with the distance between some of the design points, and as a result, a number of off-diagonal elements of  $A$  take a value very close to 1, rendering the matrix singular. The physical meaning of this condition is that the respective design points are so close in the input space for the given correlation lengths that they offer no new information.

A method for identifying such points is provided by the pivoted Cholesky decomposition [18]. The Cholesky decomposition of a positive definite matrix involves decomposing a matrix to a lower triangular matrix and its conjugate transpose, i.e.

$$LL^T = A. \quad (34)$$

The pivoted Cholesky decomposition in addition returns a permutation matrix  $P$ , such that  $P^T A P = LL^T$ . The pivoting is such that the points that have negative or zero eigenvalues are found at the bottom of the matrix. These eigenvalues correspond to design points that are too close to others, and in such a way, the pivoted Cholesky decomposition provides a way of identifying these points. After identification, these points can be excluded from building the emulator, as they do not offer any new information, and the correlation matrix  $A$  can become positive definite.

Numerical problems can crop up even when the matrix  $A$  is positive definite. These are often due to rounding errors at the higher decimal places, which can nevertheless interfere with the estimation or prediction. A typical example can be found in the calculation of  $u_0(\mathbf{x}, \mathbf{x})$ , where

the quadratic term  $c(\mathbf{x}, D)A^{-1}c(D, \mathbf{x})$  can be slightly greater than one, due to a rounding off error, forcing  $u_0(\mathbf{x}, \mathbf{x})$  to be negative. The robustness of the calculations to numerical errors of this type can be enhanced by using the Cholesky decomposition in the calculation of the involved matrix inverses. Furthermore, matrix inversion via the Cholesky decomposition is more efficient computationally, because it exploits in essence the symmetric form of the correlation matrix  $A$ .

Matrix inverses are not typically found isolated in the Gaussian process equations. Instead they are either found multiplying a matrix/vector from the left, e.g.  $A^{-1}y$  or in quadratic expressions of the form  $H^T A^{-1}H$ . A numerically robust implementation for the first type is

$$A^{-1}y \equiv L^T \backslash (L \backslash y). \quad (35)$$

The backslash operator  $\backslash$  denotes left matrix division, which is the solution to the linear system of equations  $Ax = y \leftrightarrow x = A \backslash y$ . The fact that the matrix  $L$  is triangular, allows the expressions  $L \backslash y$  to be efficiently calculated by backsubstitution. The quadratic expressions of the type  $H^T A^{-1}H$  can be calculated by defining an intermediate vector  $v$  as in the following

$$H^T A^{-1}H \equiv v^T v, \quad v = L \backslash H. \quad (36)$$

Although the above could have been calculated in one step as  $H^T(L^T \backslash (L \backslash H))$ , the above method requires one backsubstitution less, at the expense of some extra storage space for vector  $v$ , which is usually negligible.

Finally, the Cholesky decomposition can be used in the calculation of the logarithm of the determinants, taking advantage of the fact that the determinant of a positive definite matrix is the product of the squared terms on the diagonal of the triangular matrix. We can therefore calculate

$$\ln |A| \equiv 2 \sum \ln(L_{ii}) \quad (37)$$

noting at the same time that the summation of the logarithms is numerically more stable than the logarithm of the product.

## References

- [1] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–435, 1989.
- [2] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models (with discussion)," *Journal of the royal statistical society, ser. B*, vol. 63, pp. 425–464, 2001.
- [3] J. E. Oakley and A. O'Hagan, "Probabilistic sensitivity analysis of complex models: a Bayesian approach," *J. R. Statist. Soc. B*, vol. 66, no. 3, pp. 751–769, 2004.
- [4] R. Paulo, "Default priors for Gaussian processes," *Annals of Statistics*, vol. 33, pp. 556–582, 2005.
- [5] J. L. Palmer and L. I. Pettit, "Risks of using improper priors with Gibbs sampling and autocorrelated errors," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 245–249, 1996.
- [6] J. O. Berger, V. D. Oliveira, and B. Sansó, "Objective Bayesian analysis of spatially correlated data," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1361–1374, 2001.
- [7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

- [8] H. Patterson and R. Thompson, “Recovery of inter-block information when block sizes are unequal,” *Biometrika*, vol. 58, no. 3, pp. 545–554, 1971.
- [9] D. Harville, “Bayesian inference for variance components using only error contrasts,” *Biometrika*, vol. 61, pp. 383–385, 1974.
- [10] A. O’Hagan and J. Forster, *Kendall’s advanced theory of statistics*. Wiley, 2004.
- [11] B. Nagy, J. Loeppky, and W. Welch, “Correlation parameterization in random function models to improve normal approximation of the likelihood or posterior,” Dept. of Statistics, The University of British Columbia, URL <http://stat.ubc.ca/Research/TechReports/techreports/229.pdf>, Tech. Rep. 229, 2007.
- [12] J. O. Berger and J. M. Bernardo, “Estimating a product of means: Bayesian analysis with reference priors,” *Journal of the American Statistical Association*, vol. 84, pp. 200–207, 1989.
- [13] J. M. Bernardo, “Reference posterior distributions for Bayesian inference (with discussion),” *Journal of the royal statistical society, ser. B*, vol. 41, pp. 113–147, 1979.
- [14] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. London: Chapman and Hall, 1995.
- [15] L. Bastos and A. O’Hagan, “Diagnostics for Gaussian process emulators,” *Technometrics*, vol. 51, pp. 425–438, 2009.
- [16] N. R. Edwards and R. Marsh, “Uncertainties due to transport-parameter sensitivity in an efficient 3-d ocean-climate model,” *Climate dynamics*, vol. 24, pp. 415–433, 2005.
- [17] D. J. McNeall, “Dimension reduction in the Bayesian analysis of a numerical climate model,” Ph.D. dissertation, University of Southampton, Sep. 2008.
- [18] G. H. Golub and C. F. Van Loan, *Matrix computations*. Baltimore: Johns Hopkins University Press, third edition, 1996.