J. R. Statist. Soc. B (2001)
63, *Part* 3, *pp.* 425–464

Bayesian calibration of computer models

Marc C. Kennedy and Anthony O'Hagan

University of Sheffield, UK

[*Read before* The Royal Statistical Society *at a meeting organized by the* Research Section *on Wednesday, December 13th, 2000*, Professor P. J. Diggle *in the Chair*]

Summary. We consider prediction and uncertainty analysis for systems which are approximated using complex mathematical models. Such models, implemented as computer codes, are often generic in the sense that by a suitable choice of some of the model's input parameters the code can be used to predict the behaviour of the system in a variety of specific applications. However, in any specific application the values of necessary parameters may be unknown. In this case, physical observations of the system in the specific context are used to learn about the unknown parameters. The process of fitting the model to the observed data by adjusting the parameters is known as calibration. Calibration is typically effected by *ad hoc* fitting, and after calibration the model is used, with the fitted input values, to predict the future behaviour of the system. We present a Bayesian calibration technique which improves on this traditional approach in two respects. First, the predictions allow for all sources of uncertainty, including the remaining uncertainty over the fitted parameters. Second, they attempt to correct for any inadequacy of the model which is revealed by a discrepancy between the observed data and the model predictions from even the best-fitting parameter values. The method is illustrated by using data from a nuclear radiation release at Tomsk, and from a more complex simulated nuclear accident exercise.

Keywords: Calibration; Computer experiments; Deterministic models; Gaussian process; Interpolation; Model inadequacy; Sensitivity analysis; Uncertainty analysis

1. Overview

1.1. Computer models and calibration

Various sciences use mathematical models to describe processes that would otherwise be very difficult to analyse, and these models are typically implemented in computer codes. Often, the mathematical model is highly complex, and the resulting computer code is large and may be expensive in terms of the computer time required for a single run. Nevertheless, running the computer model will be much cheaper than making direct observations of the process. Sacks, Welch, Mitchell and Wynn (1989) have given several examples. The codes that we consider are deterministic, i.e. running the code with the same inputs always produces the same output.

Computer models are generally designed to be applicable to a wide range of particular contexts. However, to use a model to make predictions in a specific context it may be necessary first to *calibrate* the model by using some observed data. To illustrate this process we introduce a simple example. Two more examples are described in detail in Section 2.2.

To decide on a dose regime (e.g. size, frequency and release rates of tablets) for a new drug, a pharmacokinetic model is used. This models the movement of the drug through various 'compartments' of the patient's body and its eventual elimination (e.g. by chemical reactions

Address for correspondence: Anthony O'Hagan, Department of Probability and Statistics, University of Sheffield, Sheffield, S3 7RH, UK.
E-mail: a.ohagan@sheffield.ac.uk

 1369–7412/01/63425

or excretion). Such a model allows the consequences of any given dose regime to be explored. However, to use the model for a particular drug it is necessary to specify rates with which it moves between different body compartments, such as the rate of transfer from the stomach to blood, or of elimination from the liver. Some or all of these rates will be specific to the drug in question. and it is through requiring the user to specify them as inputs to the code that the model achieves applicability to a wide range of drugs. These rates will, of course, not be known for a given drug, and so experiments are performed to obtain observational data. It is not possible to obtain the rates themselves. Instead, the data are observations of certain *outputs* of the pharmacokinetic model, e.g. the concentration in blood or urine at certain time points. Crudely put, calibration is the activity of adjusting the unknown rate parameters until the outputs of the model *fit* the observed data.

More generally, a computer model will have a number of *context-specific* inputs that define a particular situation in which the model is to be used. When, as is often the case, the values of one or more of the context-specific inputs are unknown, observations are used to learn about them. This is calibration.

In current practice, calibration invariably consists of searching for a set of values of the unknown inputs such that the observed data fit as closely as possible, in some sense, to the corresponding outputs of the model. These values are considered as estimates of the context-specific inputs, and the model is then used to predict the behaviour of the process in this context by setting these inputs to their estimates.

Clearly, this 'plug-in' prediction treats the context-specific inputs as if they were known. The reality is that they are only estimated, and residual uncertainty about these inputs should be recognized in subsequent predictions from the model.

We present in this paper a Bayesian approach to the calibration of computer models. We represent the unknown inputs as a parameter vector $\boldsymbol{\theta}$. Using the observed data we derive the posterior distribution of $\boldsymbol{\theta}$, which in particular quantifies the 'residual uncertainty' about $\boldsymbol{\theta}$. This uncertainty is fully accounted for when using the computer model subsequently for prediction, by obtaining a predictive distribution. The principles of Bayesian predictive inference are set out in Aitchison and Dunsmore (1975), but the problem of computer code calibration has some complicating features.

Our approach treats the computer code as simply a 'black box'. We make no use of information about the mathematical model implemented by the code, except in so far as this may be represented through the prior information about the relationship between inputs and outputs. Clearly, methods that open up the black box and exploit its structure might prove to be more powerful than our approach, but they would be correspondingly more complex to apply. This is a potentially important topic for future research.

1.2. Outline of this paper

Section 2 provides a detailed analysis of issues associated with computer codes. In particular, there are several other sources of uncertainty in the use of computer codes besides uncertainty about context-specific inputs, and calibration for the purpose of prediction is just one of several topics in the statistical analysis of computer code outputs. Section 2 begins with a careful study of uncertainties in computer models, illustrating the possible sources with some detailed examples. It ends with a review of previous work on calibration and related problems.

Our Bayesian method is built on a general Bayesian approach to inference about unknown functions. A review of the relevant theory and published work is given in Section 3. Our basic

model and analysis is presented in Section 4, and practical application issues are addressed in Section 5. Section 6 presents a case-study based on real data. Section 7 offers some conclusions and directions for further work.

2. Statistical analysis of computer code outputs

2.1. Uncertainties in computer models

The widespread application of computer models is accompanied by a widespread concern about quantifying the uncertainties prevailing in their use. The following is one way of classifying the various sources of uncertainty.

2.1.1. Parameter uncertainty

We have already discussed the matter of uncertainty about the values of some of the computer code inputs. We can think of those inputs as unknown parameters of the model. Generally, they specify features of a particular application context, but they may also be more global parameters, assumed to have a common value over a range of contexts or even in all contexts.

2.1.2. Model inadequacy

No model is perfect. Even if there is no parameter uncertainty, so that we know the true values of all the inputs required to make a particular prediction of the process being modelled, the predicted value will not equal the true value of the process. The discrepancy is model inadequacy. Since the real process may itself exhibit random variability, we define model inadequacy to be the difference between the true *mean* value of the real world process and the code output at the true values of the inputs. Note that this definition is not precise until we understand what are meant by true values of inputs and the true value of the process. Given that there is model inadequacy, we cannot think of the true input values as those which lead to perfect predictions being outputted, so how *can* we define true values for the uncertain input parameters? This issue is discussed in Section 4.3.

2.1.3. Residual variability

The model is supposed to predict the value of some real process under conditions specified by the inputs. In practice, the process may not always take the same value if those conditions are repeated. We call this variation of the process even when the conditions are fully specified residual variability. There are really two sources of uncertainty combined in one here. The process itself may be inherently unpredictable and stochastic, but it may also be that this variation would be eliminated (or at least reduced) if only we could recognize and specify within the model some more conditions. The latter is effectively another form of model inadequacy, where the model lacks detail to discriminate between conditions which actually lead to different process values. However, we define the true process value to be a mean value averaged over these unrecognized conditions, as well as with respect to intrinsic random variation; model inadequacy has been defined relative to this true mean value of the process. The variability due to unrecognized conditions is deemed to be part of residual variability.

2.1.4. Parametric variability

It is often desired to use the model to predict the process when some of the conditions specified in the inputs are uncontrolled and unspecified. In a sense, this is the opposite of the

problem of the model inputs being insufficiently detailed, which contributes to residual variability. Here, the inputs require more detail than we desire (or are able) to use. By leaving some of the input parameters unspecified, and allowing them to vary according to an appropriate joint distribution, the predicted process value acquires an extra uncertainty that we shall call parametric variability.

2.1.5. Observation error

In tackling the calibration problem, we will be making use of actual observations of the process. We should allow for the possibility of observation errors. This adds further uncertainty in addition to residual variation, although in practice it may not be feasible to separate them (see Section 4.2).

2.1.6. Code uncertainty

The output of the computer code given any particular configuration of inputs is in practice not known until we actually run it with those inputs. *In principle*, we could say that it is not really unknown because the output is a known function of the inputs. After all, there is a mathematical model which, at least implicitly, defines that function. Nevertheless, *in practice* the relationship is so complex that it needed to be implemented in a computer code (which may even take hours to run), and it is not realistic to say that the output is known for given inputs before we actually run the code and see that output. It may not be practical to run the code to observe the output for every input configuration of interest, in which case uncertainty about code output needs to be acknowledged. Examples are given in Section 2.2.

2.2. Examples

It is helpful to be able to relate the various concepts, definitions and notation in this paper to some concrete examples of real computer codes. The following are just two of many examples that could be given.

2.2.1. Gaussian plume model

In the field of radiological protection, a simple Gaussian plume model (Clarke, 1979) is used to predict the dispersion and subsequent deposition of radioactive material following an accidental release. Under these circumstances the detailed input information that is required to run more complex models is not available.

The code inputs can be divided into those defining the atmospheric conditions at the time of the accident (wind direction, wind speed and atmospheric stability) and those defining the nature of the release (source term, source location, release height, release duration and deposition velocity).

The dispersion of radionuclides is a highly complex process involving various chemical and environmental processes which are not directly observable. Many simplifying assumptions are made in the Gaussian plume model, typically resulting in a high degree of model inadequacy. For example, the speed and direction of the wind are assumed to remain constant during the travel time of the particles released.

Even for this simplified model many of the inputs have parameter uncertainty associated with them. The source term, which represents the amount of material released, and the deposition velocity, which is the rate at which material in the air at ground level is deposited on the ground, are examples. The appropriate value for the deposition velocity is very

difficult to determine (see Jones (1981)). Default values are often used based on the size of the particles and the type of terrain over which the material passes. The height and duration of the release, and the wind speed and direction may also have associated parameter uncertainty.

The Gaussian plume model is cheap. We can make many thousands of runs within a very short space of time, so for practical purposes code uncertainty is not an issue and we can treat the function as known. This is not true of more complex atmospheric dispersion models.

During the course of an actual release, measurements of deposition are made by different organizations using a variety of devices. The process of making these measurements is not straightforward. Typically a sample of grass is taken from a site and analysed in a laboratory. Measurement error is introduced at each stage of the measurement process.

This discussion relates to the use of the plume model in a specific accident scenario where parameter uncertainty represents beliefs about true values of the input parameters for an actual release. We may also want to consider a risk assessment context, where we want to predict possible contamination in the future. An accident could occur at any time, and therefore the source term, wind speed and wind direction are random, and the inputs are subject to parametric variability.

2.2.2. Hydrological model

Hydrological models are used to predict ground-water flow, e.g. to predict the movement of contaminants in the soil or the level of discharge from a stream after rainfall. We consider the model described in Romanowicz *et al.* (1994), which predicts stream discharge. Inputs to the model include a time series of measured rainfall data, the rate of evapotranspiration, the average effective transmissivity of the soil T_0 when the profile is just saturated and a constant m which is related to the level of subsurface drainage.

Parameter uncertainty about T_0 and m follows from the fact that the model is used to predict water flows through complex geological structures which are not directly observable. The measured rainfall data will include random measurement error, which is an example of parametric variability. Model inadequacy arises from the simplifications that are introduced by the modelling process, as with the Gaussian plume model.

Romanowicz *et al.* (1994) used measurements of actual stream discharge to learn about the values of m and T_0. However, as they pointed out, it is not possible to measure the true value of the flow process, since we can only make point measurements of the heterogeneous pattern of flow. This relates to the idea of residual variability described in Section 2.1, and we would define the 'true flow' to be an averaged value.

2.3. Statistical methods and previous work

Some of the early work in the field of statistical analysis of computer code outputs was primarily concerned with *interpolation*, i.e., given data comprising outputs at a sample of input configurations, the problem is to estimate the output at some other input configuration for which the code has not yet been run. This is relevant when the code is particularly large and expensive to run. An important review of this work is Sacks, Welch, Mitchell and Wynn (1989), and some more recent references are Currin *et al.* (1991), Morris *et al.* (1993), Bates *et al.* (1995) and Kennedy and O'Hagan (2000a).

The only form of uncertainty accounted for in this work is code uncertainty. Model inadequacy, residual variation and observation errors are not relevant because there is no attempt to predict the real process as distinct from the computer model output, and observations of that process are not used. All input parameters are supposed known, and not

subject to parameter uncertainty or parametric variation. The statistical approach that is used in this work is based on representing the computer code output as an unknown function of its inputs, and modelling that function as a stochastic process.

Another problem that has been tackled by statistical methods for a considerable time is that of *uncertainty analysis*. The objective of uncertainty analysis is to study the distribution of the code output that is induced by probability distributions on inputs. The input parameter distributions may be formulations of parameter uncertainty, i.e. parameters whose values are unknown, or of parametric variability, i.e. parameters whose values are left unspecified. The simplest approach to uncertainty analysis is a Monte Carlo solution in which configurations of inputs are drawn at random from their distribution. The code is then run for each sample input configuration and the resulting set of outputs is a random sample from the output distribution to be studied. See Helton (1993) for a review.

The Monte Carlo method for uncertainty analysis is simple but becomes impractical when the code is costly to run, because of the large number of runs required. More efficiency is claimed for Latin hypercube sampling (McKay *et al*., 1979; Stein, 1987; Owen, 1992) compared with simple Monte Carlo sampling; see for example Crick *et al*. (1988) and Helton *et al*. (1991). Aslett *et al*. (1998) combined a Monte Carlo approach to uncertainty analysis with statistical interpolation of the code, effectively using the interpolator as a cheap surrogate for the code.

Haylock and O'Hagan (1996) presented quite a different Bayesian approach to uncertainty analysis based on a Gaussian process prior model. They derived the posterior mean and variance of the output distribution, and this is extended to posterior estimation of the distribution function and the density function of the output distribution by Oakley and O'Hagan (1998).

These methods of uncertainty analysis take account of parameter uncertainty and parametric variation in addition to code uncertainty. However, the objective is still focused on the code output, in this case in the form of the output distribution, rather than on the process itself. There is therefore no treatment of model inadequacy, residual variation or observation error.

Another problem that is associated with computer codes is *sensitivity analysis*, whose goal is to characterize how the code output responds to changes in the inputs, with particular reference to identifying inputs to which the output is relatively sensitive or insensitive. A good source for the large literature on this subject is Saltelli *et al*. (2000). Although much of this makes no use of statistical methods there are some notable exceptions. See for example Helton (1993), Morris (1991), Iman and Conover (1980), Welch *et al*. (1992), Morris (1991), Homma and Saltelli (1996) and O'Hagan *et al*. (1999). In Draper *et al*. (1999), sensitivity analysis is applied across a range of 'scenarios', using the model averaging ideas of Draper (1995). As with interpolation, these statistical approaches to sensitivity analysis only take account of the source of uncertainty that is common to all statistical analyses of computer code outputs, namely code uncertainty.

The final topic in our survey of this field is *calibration*. As discussed in Section 1.1, the traditional way of estimating unknown parameters is by an *ad hoc* search for the best-fitting values. Some account is thereby taken of observation errors, residual variation and model inadequacy, but only implicitly through the measure of the discrepancy in fits. This measure is not generally developed by modelling these error terms in any explicit way and is usually entirely heuristic. Also, since the estimated values are then treated as if they were known, the subsequent predictions take no account of the (remaining) parameter uncertainty.

In contrast, the generalized likelihood uncertainty estimation method of Romanowicz *et al*. (1994) does allow fully for parameter uncertainty. The approach is effectively Bayesian. An initial Monte Carlo sample is drawn from what amounts to the prior distribution of the

unknown inputs and is then weighted by a likelihood term. Predictions are made using all the sampled input configurations, and the result is a weighted sample from the posterior predictive distribution. For instance, the weighted sample mean is an estimate of the mean of this predictive distribution.

It would be possible within the generalized likelihood uncertainty estimation method to allow for the code uncertainty arising from having only a sample of runs, and also to allow for parametric variation by extending the Monte Carlo method to allow for draws from the unspecified inputs at the prediction stage, but these are not done in the published literature to date. As in the more traditional calibration approach, the likelihood is rather heuristic and only very loosely based on modelling the discrepancy between code outputs and the real process. Model inadequacy, residual variation and observation errors are not distinguished or modelled explicitly. No account is taken of them in prediction, the objective still being to estimate the code output rather than reality.

Another Bayesian approach to calibration is given by Craig *et al.* (1996, 2001). They employed modelling for the relationship between code inputs and output that is akin to the Gaussian process model mentioned in connection with much other work in this field. In Craig *et al.* (1996), the primary objective is to make the search for a best-fit calibration more efficient and systematic. Their modelling reflects the iterative nature of the search, and their methods adopt the Bayes linear philosophy of Goldstein (1986, 1988) and Wooff (1992), as opposed to a fully specified Bayesian analysis. The approach is extended in Craig *et al.* (2001) to treat the question of prediction following calibration.

Cox *et al.* (1992) described a calibration method that is similar to the traditional search for best-fitting parameters but which replaces an expensive code with the much cheaper interpolator obtained using the Gaussian process model. Their method does not account for remaining parameter uncertainty at the prediction stage. See also Cox *et al.* (1996).

An attempt to combine prior expert opinion on both the calibration parameters and the model output is given by Raftery *et al.* (1995) using an approach that they called Bayesian synthesis. This was criticized by Wolpert (1995) and Schweder and Hjort (1996), and in a follow-up paper Poole and Raftery (1998) proposed an alternative called Bayesian melding. Neither method explicitly recognizes model inadequacy and the underlying computer code is supposed to be sufficiently simple for code uncertainty to be ignored.

Our analysis in this paper is the first attempt to model, and to take account of explicitly, *all* the sources of uncertainty arising in the calibration and subsequent use of computer models. We freely acknowledge that our method as currently implemented is not fully Bayesian, because we estimate hyperparameters by (approximate) posterior modes. To take full account of uncertainty in hyperparameters is another topic for further research, but we believe that our present methodology is the fullest treatment to date of computer code uncertainties, and we argue that it may be adequate in practice.

3. Bayesian inference for functions

3.1. Gaussian processes

The use of Gaussian processes has been mentioned several times in Section 2.3. They are being increasingly used in current statistical research and will be employed in this paper to model both the computer code output and model inadequacy. It is therefore appropriate to review the key theoretical and practical issues, together with some discussion of alternative models.

Let $f(\cdot)$ be a function mapping an input $\mathbf{x} \in \mathcal{X}$ into an output $y = f(\mathbf{x})$ in $\mathbb{R}$. The input space $\mathcal{X}$ can be arbitrary but is typically a subset of $\mathbb{R}^q$ for some q, so we can write $\mathbf{x}$ as a

vector $\mathbf{x} = (x_1, x_2, \ldots, x_q)$. We constrain y to be a scalar, $y \in \mathbb{R}$, for simplicity here and throughout this paper, but it is equally possible for y to be a vector in $\mathbb{R}^{q'}$. We regard $f(\cdot)$ as an unknown function, and in a Bayesian framework it therefore becomes a random function. The Gaussian process is a flexible and convenient class of distributions to represent prior knowledge about $f(\cdot)$. In a non-Bayesian framework, we might treat $f(\cdot)$ as if it were drawn randomly from some population of functions and postulate a Gaussian process model for the distribution of functions in that population. This is indeed the implied interpretation of the non-Bayesian work in Sacks, Welch, Mitchell and Wynn (1989), for instance, although they were clearly also aware of the Bayesian interpretation. In the present authors' opinion, the Bayesian interpretation is far more natural and will be used throughout this paper.

Formally, $f(\cdot)$ has a Gaussian process distribution if for every $n = 1, 2, 3, \ldots$ the joint distribution of $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ is multivariate normal for all $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$. In particular, $f(\mathbf{x})$ is normally distributed for all $\mathbf{x} \in \mathcal{X}$.

The distribution is characterized by its mean function $m(\cdot)$, where $m(\mathbf{x}) = E\{f(\mathbf{x})\}$, and its covariance function $c(\cdot, \cdot)$, where $c(\mathbf{x}, \mathbf{x}') = \mathrm{cov}\{f(\mathbf{x}), f(\mathbf{x}')\}$. We use the notation $f(\cdot) \sim N\{m(\cdot), c(\cdot, \cdot)\}$ to denote the assertion that $f(\cdot)$ has a Gaussian process distribution with mean function $m(\cdot)$ and covariance function $c(\cdot, \cdot)$.

The use of Gaussian processes to represent prior distributions (or frequentist models) for unknown functions dates back at least to Kimeldorf and Wahba (1970) and O'Hagan (1978). Both used the Gaussian process effectively to model a regression function in a nonparametric way. Although O'Hagan (1978) gave a very general treatment of a Gaussian process regression function, the full potential for Gaussian process models did not begin to be exploited until much later.

As we have seen in Section 2.3, their use to represent deterministic computer codes was described by Sacks, Welch, Mitchell and Wynn (1989), who reviewed work of this kind over a period of several years. Independently, Diaconis (1988) and O'Hagan (1991, 1992) described their application for arbitrary deterministic functions and for the traditional problems of numerical analysis — interpolation, integration and optimization. It is worth noting that the numerical analysis literature also includes work on methods that are optimal for randomly generated functions, including Gaussian processes. See Novak (1988), for example.

The Gaussian process also underlies, implicitly or explicitly, the methods of geostatistics, also known as kriging. See Matheron (1963) for the classical non-Bayesian theory and Omre (1987) and Handcock and Stein (1993) for Bayesian versions in which the Gaussian process appears explicitly as a prior distribution. In geostatistics, the $\mathcal{X}$-space is geographic, usually two dimensional but occasionally three dimensional. The function $f(\cdot)$ represents some characteristic that might be measured at any point in some geographic region. A major concern in geostatistics is the estimation of the covariance function or, equivalently, the semivariogram. See in particular Stein (1999). Other examples of the use of Gaussian processes to model unknown functions are reviewed in Neal (1999).

3.2. Modelling issues

The Gaussian process is used in practice for much the same reasons that normal distributions are used so ubiquitously in statistical theory and modelling. They are convenient, flexible and often quite realistic. It is, of course, important that normality, and specifically joint normality, is a reasonable representation of prior knowledge or beliefs about $f(\cdot)$. Transformations may be useful in this context, just as they are in other applications of normal theory models.

Given that a Gaussian process is a reasonable modelling choice, the mean and covariance functions should then be specified to reflect detailed prior knowledge about $f(\cdot)$. For instance, if stationarity was a feature of prior beliefs, so that the prior distribution of $f(\mathbf{x})$ is the same as that of $f(\mathbf{x}+\mathbf{d})$ for any $\mathbf{d} \in \mathcal{X}$ (and where the operation of addition on $\mathcal{X}$ is defined), $m(\cdot)$ will be a constant and $c(\mathbf{x}, \mathbf{x}')$ a function of $\mathbf{x}-\mathbf{x}'$ only.

In general, $m(\cdot)$ may be any function on $\mathcal{X}$ but $c(\cdot, \cdot)$ must have the property that for every $n = 1, 2, \ldots$ the variance–covariance matrix of $f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)$ (comprising elements $c(\mathbf{x}_i, \mathbf{x}_j)$) is non-negative definite for all $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$. Some conditions for a covariance function to be non-negative in this sense are given in Cressie (1991).

A useful device is to model $m(\cdot)$ and $c(\cdot, \cdot)$ hierarchically. In the case of $m(\cdot)$ we can use the linear model structure

$$m(\cdot) = \mathbf{h}(\cdot)^{\mathrm{T}}\boldsymbol{\beta}, \tag{1}$$

where $\mathbf{h}(\cdot) = (h_1(\cdot), h_2(\cdot), \ldots, h_p(\cdot))^{\mathrm{T}}$ is a vector of p known functions over $\mathcal{X}$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^{\mathrm{T}}$ is a vector of p unknown coefficients which are given a prior distribution at the next stage in the hierarchy. Thus $\mathbf{h}(\cdot)$ describes a class of shapes and model (1) expresses a belief that $f(\cdot)$ may be approximated by a function in this class. For instance $\mathbf{h}(x) = (1, x, \ldots, x^{p-1})^{\mathrm{T}}$ defines $m(\cdot)$ to be a polynomial of degree $p-1$ when x is a scalar.

As a prior distribution for $\boldsymbol{\beta}$, the multivariate normal distribution is a convenient choice. For instance, it has the property that $f(\cdot)$ remains a Gaussian process marginally after integrating out $\boldsymbol{\beta}$. Of course, the prior distribution should be specified to reflect genuine belief rather than convenience, but in practice prior information about hyperparameters such as $\boldsymbol{\beta}$ will typically be weak. The conventional representation of weak prior information through the improper uniform density $p(\boldsymbol{\beta}) \propto 1$ is therefore often used.

Note that we can separate the mean and covariance structures by writing

$$f(\mathbf{x}) = m(\mathbf{x}) + e(\mathbf{x}) = \mathbf{h}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta} + e(\mathbf{x}) \tag{2}$$

using model (1), where $e(\mathbf{x})$ is a zero-mean Gaussian process, with covariance function $c(\cdot, \cdot)$. The modelling of $c(\cdot, \cdot)$ is very important because it is through correlation between $f(\mathbf{x})$ and $f(\mathbf{x}')$ that we express a view that $f(\mathbf{x})$ and $f(\mathbf{x}')$ should be similar if $\mathbf{x}$ and $\mathbf{x}'$ are sufficiently close in $\mathcal{X}$, and thereby express a belief in smoothness of $f(\cdot)$. In the following sections we shall generally adopt a hierarchical model for $c(\cdot, \cdot)$ with first stage

$$c(\cdot, \cdot) = \sigma^2 r(\mathbf{x}-\mathbf{x}'), \tag{3}$$

where $r(\cdot)$ is a correlation function having the property $r(0) = 1$. This formulation expresses stationarity in prior information about $e(\cdot)$; we have a common (unknown) variance σ^2 and correlation that only depends on $\mathbf{x}-\mathbf{x}'$. The correlation function is further expressed in terms of other unknown hyperparameters; for instance if $\mathbf{x} = (x_1, \ldots, x_q)$ is a vector,

$$r(\mathbf{x}-\mathbf{x}') = \exp\left\{-\sum_{j=1}^{q}\omega_j(x_j - x_j')^2\right\}. \tag{4}$$

Then at the next stage we would express prior distributions for the variance σ^2 and the roughness parameters $\omega_1, \ldots, \omega_q$.

Of course equation (4) is just one possible formulation. A more general expression that has been widely used replaces $(x_j - x_j')^2$ by $|x_j - x_j'|^{\alpha}$, where α may have a specified value or be another hyperparameter. Even more generally, we can allow a different α_j in each dimension. Another generalization of equation (4) would be to set $r(\mathbf{d}) = \exp(-\mathbf{d}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{d})$, where

$\mathbf{\Omega}$ is an unknown symmetric positive definite matrix, which in equation (4) has the form $\mathbf{\Omega} = \text{diag}(\omega_1, \ldots, \omega_q)$.

In geostatistics it is normal to invest considerable effort in estimating $\sigma^2 r(\mathbf{x} - \mathbf{x}')$, or equivalently $\sigma^2\{r(0) - r(\mathbf{x} - \mathbf{x}')\}$, which is there known as the semivariogram. The geostatistics literature contains proposals for a wide range of semivariogram forms. See for instance Cressie (1991), Handcock and Wallis (1994), Stein (1999) and Chilès and Delfiner (1999).

3.3. Other Bayesian nonparametric models

The Gaussian process is a flexible and popular form for prior information about functions. It is of course possible to model an unknown function parametrically, asserting that it is definitely a member of some parametric family, so that prior information is expressed as a prior distribution for the parameters. An obvious example is representing a regression function parametrically in a linear model. In contrast, Gaussian processes are nonparametric because they do not constrain the function to be in a specific parametric family. The hierarchical form may, however, be viewed via equation (2) as *semiparametric*, combining a parametric underlying mean ($\mathbf{h}(\cdot)^{\mathrm{T}}\boldsymbol{\beta}$) with a Gaussian process residual term ($e(\cdot)$).

The literature of Bayesian nonparametric and semiparametric methods is growing rapidly. In addition to work using Gaussian processes, already referred to, various other ways have been proposed to express prior beliefs about functions. One alternative to Gaussian processes for modelling general functions is to represent the function as a linear combination of components of a set of basis functions. Smith and Kohn (1998) discussed this generally, contrasting splines and other bases. In the same volume of Dey *et al.* (1998), Vidakovic (1998) considered wavelet bases and Rios Insua and Müller (1998) dealt with neural networks, which employ sigmoidal basis functions. We can note here two connections between Gaussian processes and the basis function approach. First, Neal (1996) has shown that Gaussian processes can be viewed as neural networks with an infinite number of hidden nodes. Second, the posterior mean from a Gaussian process is a linear combination of the set of basis functions $c(\mathbf{x}, \mathbf{x}_i)$ formed by the correlation functions centred at the observations.

Another approach to modelling general functions is given by Liu and Arjas (1998), who modelled a growth curve by a piecewise linear process. All these approaches may be applied to other kinds of functions such as computer code outputs.

There is also a wide literature on more specialized models for unknown distributions; see other papers in Dey *et al.* (1998) and the review of Walker *et al.* (1999). However, these are not directly relevant to computer code outputs.

4. Bayesian calibration

4.1. Calibration and variable inputs

In the calibration problem we must distinguish between two groups of inputs to the computer model. One group comprises the unknown context-specific inputs that we wish to learn about; we refer to these as the calibration inputs. The calibration inputs are supposed to take fixed but unknown values $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{q_2})$ for all the observations that will be used for calibration, and for all the instances of the true process that we wish to use the calibrated model to predict. The other group comprises all the other model inputs whose values might change when we use the calibrated model. These are referred to as the variable inputs. The variable inputs are assumed to have known values for each of the observations that will be

used for calibration. In any subsequent use of the model their values will either be known or subject to parametric variability. For example in Section 2.2, variable inputs include the (x, y) co-ordinates of the Gaussian plume model or rainfall levels in the hydrological model.

We denote the output of the computer model when the variable inputs are given values $\mathbf{x} = (x_1, \ldots, x_{q_1})$ and when the calibration inputs are given values $\mathbf{t} = (t_1, \ldots, t_{q_2})$ by $\eta(\mathbf{x}, \mathbf{t})$. Note that we distinguish between the unknown value $\boldsymbol{\theta}$ of the unknown calibration inputs, that corresponds to the particular real process for which we wish to calibrate the model, from the (known) value $\mathbf{t}$ that we set as inputs when running the model. We never observe output from the model without knowing all the inputs specifying that run. We refer to $\boldsymbol{\theta}$ as the (vector of) *calibration parameters*.

We denote the true value of the real process when the variable inputs take values $\mathbf{x}$ by $\zeta(\mathbf{x})$. Only the variable inputs are needed here. For the computer code we can vary the calibration inputs but they are fixed for the real process.

The calibration data comprise the n observations $\mathbf{z} = (z_1, \ldots, z_n)^{\mathrm{T}}$, where z_i is an observation of $\zeta(\mathbf{x}_i)$ for known variable inputs $\mathbf{x}_i$, but subject to error. In addition, we have the outputs $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ from N runs of the computer code, where

$$y_j = \eta(\mathbf{x}_j^*, \mathbf{t}_j)$$

and both the variable inputs $\mathbf{x}_j^*$ and the calibration inputs $\mathbf{t}_j$ are known for each run. The full set of data that is available for the analysis is $\mathbf{d}^{\mathrm{T}} = (\mathbf{y}^{\mathrm{T}}, \mathbf{z}^{\mathrm{T}})$. Note that generally N will be much larger than n, since even if the computer code is expensive or time consuming to run it will still be much cheaper than obtaining observations of the real process.

A further comment in connection with variable inputs is that we are treating the computer code output for any given set of inputs as a scalar. In practice, computer codes typically produce many outputs for a given run. It is not generally necessary, however, to regard the output as being multivariate. The reason is that we can define one or more variable inputs to index the outputs. For instance, a Gaussian plume model will typically be implemented in a computer code that, for a given run, outputs concentrations at every point in a two-dimensional grid. Rather than thinking of this as a highly multivariate output, we can define two variable inputs to index a given point on the grid. We can think of the variable inputs as being augmented by these two new indexing inputs, and a run as producing a single output at the specified point.

4.2. Model

We represent the relationship between the observations z_i, the true process $\zeta(\cdot)$ and the computer model output $\eta(\cdot, \cdot)$ in the equation

$$z_i = \zeta(\mathbf{x}_i) + e_i = \rho\, \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + e_i, \tag{5}$$

where e_i is the observation error for the ith observation, ρ is an unknown regression parameter and $\delta(\cdot)$ is a model inadequacy function that is *independent* of the code output $\eta(\cdot, \cdot)$.

Consider the observation error e_i first. Strictly, this also includes any residual variation as well as observation error (see Section 2.1). We do not imagine having replication of observations in circumstances where not only the variable inputs but also all the unrecognized conditions were the same, so it is not possible to separate the two sources of uncertainty. We shall suppose that the e_is are independently distributed as $N(0, \lambda)$. (The assumption of normality may require a transformation of the raw data, as in our use of log-deposition in Section 6.)

Now consider the implication of equation (5) that

$$\zeta(\mathbf{x}) = \rho\,\eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}), \tag{6}$$

with $\eta(\cdot, \cdot)$ and $\delta(\cdot)$ independent. This is, of course, merely one way of modelling the relationship between the code output and reality. As a partial justification, it can be given a formal derivation from a kind of Markov property, as follows. Assume that we know $\boldsymbol{\theta}$ and can make as many runs of the code as we wish, to observe $\eta(\mathbf{x}, \boldsymbol{\theta})$ for various $\mathbf{x}$. Suppose first that to predict $\zeta(\mathbf{x}')$ at some specific point $\mathbf{x}'$ we would regard it as sufficient to observe the output $\eta(\mathbf{x}', \boldsymbol{\theta})$ of a single run at the same value $\mathbf{x}'$ of the variable inputs. This is the Markov assumption and can be shown to imply equation (6) (see O'Hagan (1998)), except that ρ may formally depend on $\mathbf{x}$. The further assumption that ρ is constant seems natural and follows if we have *stationary* processes $\eta(\cdot, \cdot)$, $\delta(\cdot)$ and $\zeta(\cdot)$. Despite this argument, we repeat that equation (6) is just one possible formulation; equally cogent arguments could probably be evinced in favour of other models. In the examples that we have tried, equation (6) seems reasonable, but more experience is needed to explore this aspect of modelling.

The meaning of the *true* values $\boldsymbol{\theta}$ of the calibration parameters is addressed in Section 4.3 below.

We represent prior information about both the unknown functions $\eta(\cdot, \cdot)$ and $\delta(\cdot)$ by Gaussian processes: $\eta(\cdot, \cdot) \sim N[m_1(\cdot, \cdot), c_1\{(\cdot, \cdot), (\cdot, \cdot)\}]$ and $\delta(\cdot) \sim N\{m_2(\cdot), c_2(\cdot, \cdot)\}$. In each case, we model the mean and variance functions hierarchically as in Section 3.2. Adopting the linear model form (1) with weak prior distributions, we have $m_1(\mathbf{x}, \mathbf{t}) = \mathbf{h}_1(\mathbf{x}, \mathbf{t})^{\mathrm{T}}\boldsymbol{\beta}_1$, $m_2(\mathbf{x}) = \mathbf{h}_2(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}_2$ and

$$p(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) \propto 1. \tag{7}$$

We write the combined vector as $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}})^{\mathrm{T}}$. For the covariance functions we shall not specify any particular forms at present, but we shall suppose that they are expressed in terms of some further hyperparameters that we denote by $\boldsymbol{\psi}$. We also denote $(\rho, \lambda, \boldsymbol{\psi})$ collectively by $\boldsymbol{\phi}$. The complete set of parameters therefore comprises the calibration parameters $\boldsymbol{\theta}$, the location parameters $\boldsymbol{\beta}$ and the hyperparameters $\boldsymbol{\phi}$. It is reasonable to suppose that prior information about $\boldsymbol{\theta}$ is independent of the others, and with expression (7) we suppose that the prior distribution takes the form

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\phi}) = p(\boldsymbol{\theta})\,p(\boldsymbol{\phi}). \tag{8}$$

4.3. True parameter values

We raised the question of the meaning of true parameter values in Section 2.1. We now discuss this important topic in the context of our model (6) for the true process $\zeta(\cdot)$, although the main points are relevant to any discussion of the calibration of computer models.

Our statistical model is formulated through equation (5), which can be viewed as defining a non-linear regression model. (The analogy is not perfect but provides some useful insight.) The computer code itself defines the regression function through the term $\rho\,\eta(\mathbf{x}_i, \boldsymbol{\theta})$, with parameters ρ and $\boldsymbol{\theta}$. The other two terms can be viewed as together representing (non-independent) residuals.

In this framework, we can see that the notion of a true value for $\boldsymbol{\theta}$ has the same meaning and validity as the true values of regression parameters. The true $\boldsymbol{\theta}$ is a 'best-fitting' $\boldsymbol{\theta}$, in the sense of representing the data $z_1, \ldots, z_n$ faithfully according to the error structure that is specified for the residuals.

Now the developers of the computer model will generally have given concrete physical meanings to the calibration inputs, but the true values of these physical quantities do not necessarily equate to $\boldsymbol{\theta}$. This is inevitable in calibration when we do not believe that the model can ever be a perfect fit. It may be that the physically true value of a calibration parameter gives a worse fit, and less accurate future prediction, than another value. It is dangerous to interpret the estimates of $\boldsymbol{\theta}$ that are obtained by calibration as estimates of the true *physical* values of those parameters.

In regression modelling, assuming that one of the parameters is known corresponds to fitting a simpler, more limited class of regression functions. Fixing a parameter constrains the form of the regression function, whereas adding more unknown parameters increases the flexibility of the class of regression functions and allows a better fit to the data. By analogy, we see that if we claim to know the value of one of the calibration parameters in $\boldsymbol{\theta}$, even if this is genuinely the true physical value of that parameter, we restrict the form of the code output and may have a worse fit to the data. The discrepancy will of course be taken up by the model inadequacy function $\delta(\cdot)$ but may also lead to the calibrated estimates of other components of $\boldsymbol{\theta}$ being further from their true physical interpretations. We shall generally have far fewer real observations with which to estimate $\delta(\cdot)$ than code outputs with which to estimate $\eta(\cdot, \cdot)$, so it makes sense to make the code fit as well as possible.

So it may be reasonable to treat an input as unknown, and therefore part of the calibration parameter $\boldsymbol{\theta}$, even if we believe that its true physical value is known. Particularly if this is an influential parameter (as might be revealed by a sensitivity analysis), allowing it to deviate from the true physical value may produce an empirically better computer model of reality. Its prior distribution would be centred at the true physical value, reflecting an expectation of the model's accuracy, but with a non-zero variance.

All models are wrong, and to suppose that inputs should always be set to their 'true' values when these are 'known' is to invest the model with too much credibility in practice. Treating a model more pragmatically, as having inputs that we can 'tweak' empirically, can increase its value and predictive power.

4.4. Posterior distribution

In the remaining subsections of this section, we present the posterior analysis of the calibration problem and subsequent prediction of the true phenomenon using the calibrated code. For brevity, we give only an outline of the development here. For fuller mathematical details the reader is referred to Kennedy and O'Hagan (2000b).

The first step is to derive the posterior distribution of the parameters $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and ϕ. The full data vector $\mathbf{d}$ is normally distributed given $(\boldsymbol{\theta}, \boldsymbol{\beta}, \phi)$, and this will yield the likelihood function. To express its mean vector and variance matrix we require some more notation.

We denote the set of points at which the code outputs $\mathbf{y}$ are available by $D_1 = \{(\mathbf{x}_1^*, \mathbf{t}_1), \ldots, (\mathbf{x}_N^*, \mathbf{t}_N)\}$. Similarly, we denote the set of points for the observations $\mathbf{z}$ of the real process by $D_2 = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Augmenting each of these points by the calibration parameters $\boldsymbol{\theta}$, we define $D_2(\boldsymbol{\theta}) = \{(\mathbf{x}_1, \boldsymbol{\theta}), \ldots, (\mathbf{x}_n, \boldsymbol{\theta})\}$. If we now let $\mathbf{H}_1(D_1)$ denote the matrix with rows $\mathbf{h}_1(\mathbf{x}_1^*, \mathbf{t}_1)^{\mathrm{T}}, \ldots, \mathbf{h}_1(\mathbf{x}_N^*, \mathbf{t}_N)^{\mathrm{T}}$, the expectation of $\mathbf{y}$ is $\mathbf{H}_1(D_1)\boldsymbol{\beta}_1$. In analogous notation, the expectation of $\mathbf{z}$ is

$$\rho\, \mathbf{H}_1\{D_2(\boldsymbol{\theta})\}\boldsymbol{\beta}_1 + \mathbf{H}_2(D_2)\boldsymbol{\beta}_2.$$

Hence

$$E(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\beta}, \phi) = \mathbf{m}_d(\boldsymbol{\theta}) = \mathbf{H}(\boldsymbol{\theta})\boldsymbol{\beta},$$

where

$$\mathbf{H}(\boldsymbol{\theta}) = \begin{pmatrix} \mathbf{H}_1(D_1) & \mathbf{0} \\ \rho\,\mathbf{H}_1\{D_2(\boldsymbol{\theta})\} & \mathbf{H}_2(D_2) \end{pmatrix}.$$

To express the variance matrix of $\mathbf{d}$, define $\mathbf{V}_1(D_1)$ to be the matrix with (j, j') element $c_1\{(\mathbf{x}_j^*, \mathbf{t}_j), (\mathbf{x}_{j'}^*, \mathbf{t}_{j'})\}$, so that this is the variance matrix of $\mathbf{y}$. Define $\mathbf{V}_1\{D_2(\boldsymbol{\theta})\}$ and $\mathbf{V}_2(D_2)$ similarly, and let $\mathbf{C}_1\{D_1, D_2(\boldsymbol{\theta})\}$ be the matrix with (j, i) element $c_1\{(\mathbf{x}_j^*, \mathbf{t}_j), (\mathbf{x}_i, \boldsymbol{\theta})\}$. Then

$$\mathrm{var}(\mathbf{d}|\boldsymbol{\theta}, \boldsymbol{\beta}, \phi) = \mathbf{V}_d(\boldsymbol{\theta}) = \begin{pmatrix} \mathbf{V}_1(D_1) & \rho\,\mathbf{C}_1\{D_1, D_2(\boldsymbol{\theta})\}^{\mathrm{T}} \\ \rho\,\mathbf{C}_1\{D_1, D_2(\boldsymbol{\theta})\} & \lambda\mathbf{I}_n + \rho^2\,\mathbf{V}_1\{D_2(\boldsymbol{\theta})\} + \mathbf{V}_2(D_2) \end{pmatrix}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix.

With prior distribution (8) we now obtain the full joint posterior distribution

$$p(\boldsymbol{\theta}, \boldsymbol{\beta}, \phi|\mathbf{d}) \propto p(\boldsymbol{\theta})\,p(\phi)\,f\{\mathbf{d};\, \mathbf{m}_d(\boldsymbol{\theta}),\, \mathbf{V}_d(\boldsymbol{\theta})\}, \tag{9}$$

where $f\{\cdot;\, \mathbf{m}_d(\boldsymbol{\theta}),\, \mathbf{V}_d(\boldsymbol{\theta})\}$ is the $N\{\mathbf{m}_d(\boldsymbol{\theta}),\, \mathbf{V}_d(\boldsymbol{\theta})\}$ density function. Note that we have explicitly shown dependence on $\boldsymbol{\theta}$ but $\mathbf{m}_d(\boldsymbol{\theta})$ also depends on $\boldsymbol{\beta}$ and ρ, whereas $\mathbf{V}_d(\boldsymbol{\theta})$ depends on all of ϕ.

4.5. Estimating hyperparameters

Since the exponent of expression (9) is quadratic in $\boldsymbol{\beta}$, we can integrate $\boldsymbol{\beta}$ out analytically to give $p(\boldsymbol{\theta}, \phi|\mathbf{d})$. This, however, is an even more complex function of $\boldsymbol{\theta}$ and ϕ than expression (9) is. A fully Bayesian analysis would now integrate out the hyperparameters ϕ as well to leave the posterior distribution $p(\boldsymbol{\theta}|\mathbf{d})$ of the calibration parameters. However, $p(\boldsymbol{\theta}, \phi|\mathbf{d})$ is a highly intractable function of ϕ. Even with the most parsimonious parameterization of $c_1\{(\cdot, \cdot), (\cdot, \cdot)\}$ and $c_2(\cdot, \cdot)$, to integrate over ϕ numerically would entail at least a six-dimensional quadrature. Since much of the methodology that we develop herein may be rather computationally intensive even conditional on fixed values of ϕ, the full Bayesian analysis will not typically be practical. It is important to note also that $p(\boldsymbol{\theta}, \phi|\mathbf{d})$ will generally be improper with respect to ϕ if $p(\phi)$ is improper. To adopt a fully Bayesian analysis will therefore demand a full and careful consideration of prior information regarding the hyperparameters.

We propose instead to derive plausible estimates of the components of ϕ and then to act as if these were fixed. Thus, for inference about $\boldsymbol{\theta}$ we shall use its conditional posterior given the estimated values of ϕ. We propose to estimate the hyperparameters in two stages. In the first stage we use just the code output data $\mathbf{y}$ to estimate the hyperparameters ψ_1 of $c_1\{(\cdot, \cdot), (\cdot, \cdot)\}$. There is some information about ψ_1 in the observational data $\mathbf{z}$, but

(a) $\mathbf{z}$ depends also on the other hyperparameters and
(b) the number n of observations in $\mathbf{z}$ will typically be very much smaller than the number N of output values in $\mathbf{y}$.

Therefore very little is lost by this simplification. In the second stage we use $\mathbf{z}$ to estimate ρ, λ and the hyperparameters ψ_2 of $c_2(\cdot, \cdot)$, having now fixed ψ_1.

Now we originally set out to model, and to account for explicitly, all the sources of uncertainty identified in Section 2.1. The compromise proposed here means that we do not account *fully* for all these sources.

(a) By fixing λ at an estimated value, we do not account *fully* for observation error and residual uncertainty.
(b) By fixing ρ and the hyperparameters ψ_2 of $c_2(\cdot, \cdot)$ at estimated values, we do not account *fully* for model inadequacy.
(c) By fixing the hyperparameters ψ_1 of $c_1\{(\cdot, \cdot), (\cdot, \cdot)\}$ at estimated values, we do not account *fully* for code uncertainty.

Nevertheless, we should stress that in each case it is only the 'second-order' effect of uncertainty about hyperparameters that is neglected, and we believe that our analysis captures the major part of all these sources of uncertainty. We therefore claim that our analysis *does* recognize *all* sources of uncertainty, and that it is more important to ensure that all sources are covered to this extent than to account for any missing hyperparameter uncertainty, at the cost of very much increased computation.

In geostatistics, it is generally recognized that kriging estimates are reasonably robust to the form of the covariance function, and even to roughness parameters in that function, but that the prediction variance will typically be very sensitive to roughness parameter values. Furthermore, such parameters are notoriously difficult to estimate. We are conscious that these considerations may give cause for concern about our treatment of hyperparameters and the choice of covariance structure. In our defence, we show in the example in Section 6 that our predictive variances calibrate well with held-back data, and this has been our experience with other examples also. In Section 6.3 we present an investigation which shows for those data that the effect of acknowledging uncertainty in roughness parameters is small.

4.6. Calibration, prediction and uncertainty analysis

Having estimated the hyperparameters ϕ we now condition on the estimates $\hat{\phi}$, so that we regard the posterior distribution of the calibration parameters to be $p(\boldsymbol{\theta}|\phi = \hat{\phi}, \mathbf{d}) \propto p(\boldsymbol{\theta}, \hat{\phi}|\mathbf{d})$. We can use this to make inference about $\boldsymbol{\theta}$, although its intractability means that numerical methods must be used. We discuss appropriate techniques in Section 5.

In practice, we shall not generally be interested in inference about $\boldsymbol{\theta}$ as such. The purpose of calibration is to use the calibrated model for predicting the real process. We can think of calibration as a preliminary to addressing the other statistical problems of interpolation, sensitivity analysis and uncertainty analysis, described in Section 2.3. Thus, the problem of predicting the true process $\zeta(\mathbf{x})$ at some specified variable inputs $\mathbf{x}$ can be seen as interpolating the function $\zeta(\cdot)$.

The posterior distribution of $\zeta(\cdot)$ conditional on the estimated hyperparameters ϕ and the calibration parameters $\boldsymbol{\theta}$ is a Gaussian process. Its mean function is given by

$$E\{z(\mathbf{x})|\boldsymbol{\theta}, \phi, \mathbf{d}\} = \mathbf{h}(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) + \mathbf{t}(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{V}_d(\boldsymbol{\theta})^{-1}\{\mathbf{d} - \mathbf{H}(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})\}, \tag{10}$$

where

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) = \begin{pmatrix} \rho\,\mathbf{h}_1(\mathbf{x}, \boldsymbol{\theta}) \\ \mathbf{h}_2(\mathbf{x}) \end{pmatrix}$$

and

$$\mathbf{t}(\mathbf{x}, \boldsymbol{\theta}) = \begin{pmatrix} \rho\,\mathbf{V}_1\{(\mathbf{x}, \boldsymbol{\theta}), D_1\} \\ \rho^2\,\mathbf{V}_1\{(\mathbf{x}, \boldsymbol{\theta}), D_2(\boldsymbol{\theta})\} + \mathbf{V}_2(\mathbf{x}, D_2) \end{pmatrix}.$$

Its covariance function is given by

$$\begin{aligned}\mathrm{cov}\{\zeta(\mathbf{x}), \zeta(\mathbf{x}')|\boldsymbol{\theta}, \phi, \mathbf{d}\} = \rho^2 c_1\{(\mathbf{x}, \boldsymbol{\theta}), (\mathbf{x}', \boldsymbol{\theta})\} + c_2(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{V}_d(\boldsymbol{\theta})^{-1}\mathbf{t}(\mathbf{x}', \boldsymbol{\theta}) \\ + (\mathbf{h}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{H}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{V}_d(\boldsymbol{\theta})^{-1}\mathbf{t}(\mathbf{x}, \boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}(\boldsymbol{\theta})\{\mathbf{h}(\mathbf{x}', \boldsymbol{\theta}) \\ - \mathbf{H}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{V}_d(\boldsymbol{\theta})^{-1}\mathbf{t}(\mathbf{x}', \boldsymbol{\theta})\},\end{aligned}$$

where $\mathbf{W}(\boldsymbol{\theta}) = \{\mathbf{H}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{V}_d(\boldsymbol{\theta})^{-1}\mathbf{H}(\boldsymbol{\theta})\}^{-1}$. By combining this distribution with $p(\boldsymbol{\theta}|\phi, \mathbf{d})$, we can make inferences about $\zeta(\mathbf{x})$, again using numerical computation methods. For instance to estimate $\zeta(\mathbf{x})$ we might use its posterior mean $E\{\zeta(\mathbf{x})|\phi, \mathbf{d}\}$ (for the estimated values of ϕ), obtained by integrating $E\{\zeta(\mathbf{x})|\boldsymbol{\theta}, \hat{\phi}, \mathbf{d}\}$ with respect to $p(\boldsymbol{\theta}|\hat{\phi}, \mathbf{d})$.

Now suppose that we wish to predict the real process in the context where one or more of the variable inputs is subject to parametric variability, as discussed in Section 2.1. Section 2.2 gives examples of computer codes for which inputs may be unspecified in this way. The problem of uncertainty analysis is to study the (extra) uncertainty in model outputs induced by this parametric variability. Although uncertainty analysis for computer codes is typically formulated in this way, i.e. with concern for uncertainty in the code outputs, in the present context the larger challenge is to study uncertainty in the real process $z(\cdot)$.

We therefore consider the random variable $\zeta(\mathbf{X})$, where the variable inputs $\mathbf{X}$ are now random, having a distribution $G_{\mathbf{X}}(\mathbf{x})$. (In practice, only a subset of the variable inputs will be subject to parametric variability, so $G_{\mathbf{X}}(\cdot)$ will be degenerate in the other dimensions.) The task of uncertainty analysis is now to make inference about the *distribution* of $\zeta(\mathbf{X})$. In particular, we wish to make inference about properties of this distribution such as the mean

$$K = E_{\mathbf{X}}\{\zeta(\mathbf{X})\} = \int_{\mathcal{X}} \zeta(\mathbf{x})\, \mathrm{d}G_{\mathbf{X}}(\mathbf{x}),$$

the variance

$$L = \mathrm{var}_{\mathbf{X}}\{\zeta(\mathbf{X})\} = K_2 - K^2,$$

where $K_2 = \int_{\mathcal{X}} \zeta(\mathbf{x})^2\, \mathrm{d}G_{\mathbf{X}}(\mathbf{x})$, or the value at some point g of the distribution function

$$F(g) = P_{\mathbf{X}}\{\zeta(\mathbf{X}) \leqslant g\} = \int_{z(\mathbf{x}) \leqslant g} \mathrm{d}G_{\mathbf{X}}(\mathbf{x}).$$

Inference about these or other summaries of the distribution of $\zeta(\mathbf{X})$ may be derived from the posterior distribution of $\zeta(\cdot)$. Details are given in Kennedy and O'Hagan (2000b).

It is equally straightforward to work in terms of an uncertainty analysis of the code output $\eta(\mathbf{x}, \boldsymbol{\theta})$, with respect to either or both of parametric variability in $\mathbf{x}$ and parametric uncertainty (after calibration) in $\boldsymbol{\theta}$.

We do not explicitly deal with sensitivity analysis in this paper: appropriate techniques are outlined in O'Hagan *et al.* (1999).

5. Implementation details

5.1. Design issues

We now consider some practical issues arising in the implementation of the theory in Section 4, beginning with the question of the choice of the sets of points at which the code is run and at which observational data are observed. The set D_2 of values $\mathbf{x}_i$ of the variable inputs for the calibration data will often not be a matter of choice. In our example in Section 6, for

example, the available data are given to us. In contrast, we shall generally be able to choose the code design D_1 of points $(\mathbf{x}_j^*, \mathbf{t}_j)$. There is a considerable literature on the design of computer experiments — see for example Sacks, Schiller and Welch (1989), Morris *et al.* (1993), Morris and Mitchell (1995) and Bates *et al.* (1996). All of this relates to the simpler problem of designing sets of code input values for interpolating the code itself, or for uncertainty analysis of the code (Haylock, 1997). The problem of design for calibration is more complex and a topic for future research. We have so far chosen designs more heuristically.

First, existing work suggests that it is important for the code design to give good coverage of the region of $(\mathbf{x}, \mathbf{t})$ space of greatest interest. The variable input co-ordinates $\mathbf{x}_j^*$ should cover both the range of points $\mathbf{x}_i$ in the calibration data and the range of values over which we may wish to predict the process in future. The calibration input co-ordinates $\mathbf{t}_j$ should cover the range that is plausible for the true value $\boldsymbol{\theta}$ of the calibration parameters. The latter suggests a sequential design approach, beginning with values spanning the prior distribution of $\boldsymbol{\theta}$ then adding more points over the range covered by its posterior distribution. For examples of this kind of sequential design approach see Bernardo *et al.* (1992), Craig *et al.* (1996) and Aslett *et al.* (1998).

A second intuitive consideration is that there should be control values $\mathbf{x}_j^*$ in D_1 that are close to the values $\mathbf{x}_i$ in D_2 to learn about the relationship between the code and reality.

For our example in Section 6, we have no choice over the calibration design D_2. We set the code design D_1 to be the Cartesian product of D_1 with a Latin hypercube design for the calibration inputs. For the latter, we have used a maximin Latin hypercube as described in Morris and Mitchell (1995). These designs give good coverage of the space and are evenly distributed in each one-dimensional projection. The use of a Cartesian product has some computational advantages which are briefly described in Kennedy and O'Hagan (2000b).

5.2. Modelling choices

In applications of the theory, we need to specify $\mathbf{h}_1(\mathbf{x}, \mathbf{t})$, $\mathbf{h}_2(\mathbf{x})$, $c_1\{(\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')\}$ and $c_2(\mathbf{x}, \mathbf{x}')$. $\mathbf{h}_1(\mathbf{x}, \mathbf{t})$ should be chosen to reflect beliefs about the general shape of the function $\eta(\mathbf{x}, \mathbf{t})$, and $\mathbf{h}_2(\mathbf{x})$ should be chosen to reflect beliefs about the shape of $\delta(\mathbf{x})$. In the latter case, particularly, we may not have any specific expectations to model through $\mathbf{h}_2(\cdot)$. Generally, it does not help to put components into these functions that are not motivated by actual prior knowledge. This is in contrast with parametric regression modelling, where adding extra regressor variables will in general produce an improved fit. The Gaussian process is nonparametric and will adapt to whatever shape of function is suggested by the data and will often do so better if spurious regressors are not included. In applications, therefore, unless there is prior information to suggest more complex modelling, we take $\mathbf{h}_1(\mathbf{x}, \mathbf{t}) = (1)$ and $\mathbf{h}_2(\mathbf{x}) = (1)$ as defaults. This means that $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are scalars and represent unknown constant means for $\eta(\cdot, \cdot)$ and $\delta(\cdot)$.

For the covariance functions we again model parsimoniously. We generally adopt the form (3)–(4), so that

$$c_1\{(\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')\} = \sigma_1^2 \exp\{-(\mathbf{x} - \mathbf{x}')^{\mathrm{T}} \boldsymbol{\Omega}_{\mathbf{x}} (\mathbf{x} - \mathbf{x}')\} \exp\{-(\mathbf{t} - \mathbf{t}')^{\mathrm{T}} \boldsymbol{\Omega}_{\mathbf{t}} (\mathbf{t} - \mathbf{t}')\}, \tag{11}$$

$$c_2(\mathbf{x}, \mathbf{x}') = \sigma_2^2 \exp\{-(\mathbf{x} - \mathbf{x}')^{\mathrm{T}} \boldsymbol{\Omega}_{\mathbf{x}}^* (\mathbf{x} - \mathbf{x}')\}, \tag{12}$$

with diagonal forms for $\boldsymbol{\Omega}_{\mathbf{t}}$, $\boldsymbol{\Omega}_{\mathbf{x}}$ and $\boldsymbol{\Omega}_{\mathbf{x}}^*$.

It is important to recognize that these are not trivial modelling choices. The Gaussian forms for the covariance function imply a belief in differentiability of both $\eta(\cdot, \cdot)$ and $\delta(\cdot)$, and indeed

imply a belief that these functions are analytic. This may be appropriate for the computer code $\eta(\cdot, \cdot)$ but these assumptions for both $\eta(\cdot, \cdot)$ and $\delta(\cdot)$ imply the same beliefs about the real world process $\zeta(\cdot)$, which will often be inappropriate. We return to this in Section 6.3.

Assuming diagonal forms for the roughness matrices implies that any elliptical anisotropy in the covariances is oriented along the individual parameter axes. A transformation may be relevant to make this assumption more realistic and is implemented in our examples as described in Section 6. Another assumption in equation (11) is that there is separability between the calibration and variable inputs in the covariance structure of the code $\eta(\cdot, \cdot)$. Separability is frequently assumed in various statistical applications, particularly for modelling space–time processes, e.g. by Haslett and Raftery (1989) and Oehlert (1993). A result in O'Hagan (1998) provides a characterization of separability that may give a justification in some contexts, but it is primarily assumed in practice for convenience and parsimony.

Finally, we may remark that even the underlying assumption of stationarity may be questioned, particularly in respect of the real process. A reasonably tractable non-stationary alternative might be the localized regression model of O'Hagan (1978). Sampson and Guttorp (1992) gave a very general nonparametric technique, but it would be much more difficult to fit this into our framework.

Our reasons for using these assumptions in our examples are briefly as follows. First, equations (11) and (12) facilitate the computation by allowing analytical results that would otherwise need to be evaluated numerically (see Kennedy and O'Hagan (2000b)), greatly increasing computation times. We employ equation (4) rather than the more general $r(\mathbf{d}) = \exp(-\mathbf{d}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{d})$ for parsimony. In general, hyperparameters are not well identified in these models. Quite different values for $\boldsymbol{\phi}$ may fit the data equally well and produce comparable predictions. The form (4) is sufficiently flexible to allow for some anisotropy in the correlation structure, and indeed we adopt the isotropic form $r(\mathbf{d}) = \exp(-\omega\mathbf{d}^{\mathrm{T}}\mathbf{d})$ wherever it seems acceptable.

We freely admit, therefore, that we employ these assumptions essentially for convenience and simplicity. However, we believe that to a large extent other assumptions could give very similar results. Some tentative support for this view is given by some findings reported in Section 6.3.

We also need to specify prior distributions $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\phi})$ for the calibration parameters and hyperparameters respectively. We adopt a normal prior distribution for $\boldsymbol{\theta}$, after transformation where appropriate, again for computational convenience. Prior information about hyperparameters will often be weak, with the possible exception of λ. It is useful, however, to try to formulate some prior knowledge about roughness parameters like ω. A uniform prior distribution should generally not be used for such parameters because our modal estimation method will often give unrealistically large estimates. Where prior information is weak, the form $p(\omega) \propto \omega^{-1}$ is preferable.

5.3. Computation

The main computational issues concern the need for numerical integration with respect to the posterior distribution of $\boldsymbol{\theta}$, and the fact that we need to invert the matrix $\mathbf{V}_d(\boldsymbol{\theta})$ for each $\boldsymbol{\theta}$-value in that numerical integration.

If the code $\eta(\cdot, \cdot)$ is complex and computer intensive, we shall expect the number N of code evaluations that are available to be relatively small (and we expect n to be smaller still). Then the inversion of the $(N + n) \times (N + n)$ matrix $\mathbf{V}_d(\boldsymbol{\theta})$ may not be a serious problem. However, for a simpler code we may expect to be able to make larger numbers of runs to obtain more

information about $\eta(\cdot, \cdot)$. Then N is potentially very large. In this case considerable computational savings are achieved by the code design D_1 having a Cartesian product form. See Kennedy and O'Hagan (2000b) for details.

Another device that might be considered for computation with large correlation matrices is the local computation approach; Vecchia (1988). However, it is not clear how that idea could be usefully applied in the more complex framework of calibration.

Turning now to the question of numerical integration with respect to $\boldsymbol{\theta}$, in our examples we use the iterative Gauss–Hermite quadrature method of Naylor and Smith (1982). This approach is realistic because the dimensionality of $\boldsymbol{\theta}$ is relatively low, so quadrature is feasible, and because the code is relatively simple, so we can afford to use Cartesian product rules and iteration. With more expensive codes or in somewhat higher dimensional $\boldsymbol{\theta}$-space it becomes important to use more efficient quadrature designs (for references see Evans and Schwartz (1995)). For high dimensional $\boldsymbol{\theta}$, it may become necessary to use simulation methods of integration: we have not explored this yet.

6. Example: Tomsk data

6.1. Data and model

We now present an analysis of data from an accident at the Tomsk-7 chemical plant in 1993. A detailed account of the accident is given in Shershakov *et al.* (1995). Measurements were made for three radionuclides. However, for this example we consider only the deposition of ruthenium 106 (^{106}Ru). A total of 695 measurements of ^{106}Ru deposition were made, at locations shown in Fig. 1. The contour lines represent an interpolation of the log-deposition

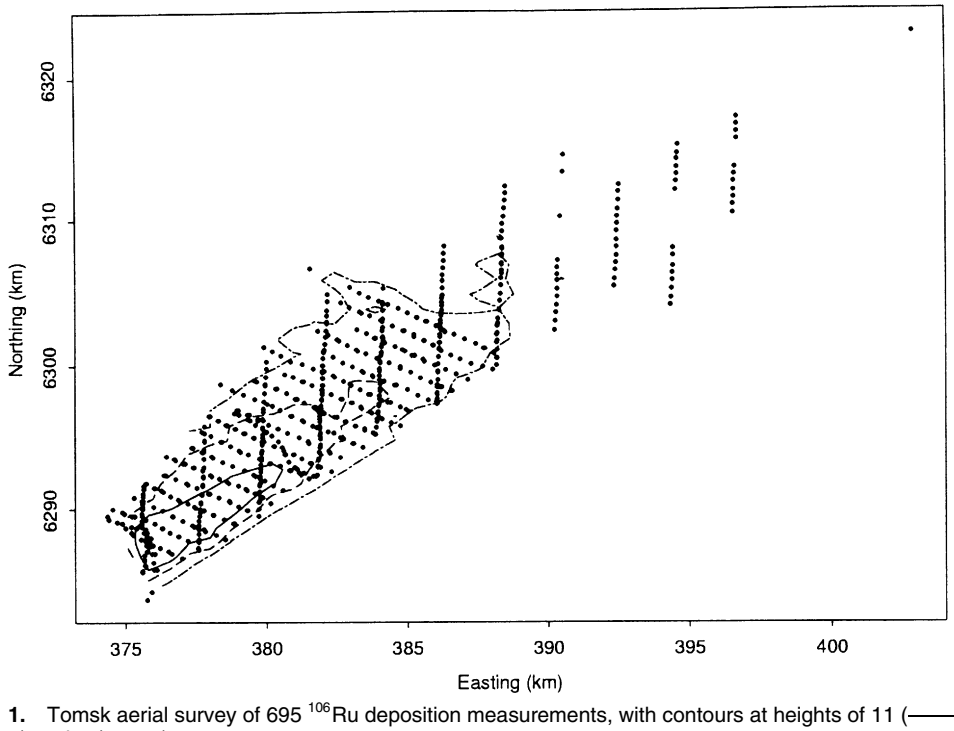

Fig. 1. Tomsk aerial survey of 695 ^{106}Ru deposition measurements, with contours at heights of 11 (———), 10 (– – –) and 9 (— · —)

at these points. These data were obtained from an aerial survey which started close to the source and continued to approximately 40 km downwind, in such a way that consecutive measurements are very close.

For the prior approximation $\eta(\cdot, \cdot)$ we use the logarithm of the Gaussian plume model described in Section 2.2, with a fixed level of background radiation added. The log-transformation is used to approximate better the assumptions of normality that are made in our model. The critical unknown inputs for this model are the source term and deposition velocity. We therefore treat the logarithms of these as the calibration parameters $\boldsymbol{\theta}$. $\zeta(\mathbf{x})$ represents the true log-deposition for variable inputs $\mathbf{x}$. The variable inputs in this case comprise two orthogonal linear functions of the northing and easting co-ordinates such that $\mathbf{x} = (0, 0)$ represents the source point and a point $\mathbf{x} = (x_1, x_2)$ is distance x_1 downwind from the source and distance x_2 from the plume centre line. The z_is are logarithms of observed depositions.

The pattern of deposition that is seen in Fig. 1 has a well-defined plume shape, and we would expect the Gaussian plume code to provide a reasonably good approximation.

A normal distribution is used to approximate prior beliefs about $\boldsymbol{\theta}$. The prior means were obtained from the National Radiological Protection Board. The variances were set to 5 to represent vague prior knowledge, and prior covariances were assumed to be 0. These are realistic values for the variances, since the National Radiological Protection Board think that the values could be a couple of orders of magnitude from the prior estimates.

In this example we have treated the plume code as a known function, for given values of the parameters, since it is practical to run the code for any input configuration of interest. Although it is perfectly possible to follow through the more complex analysis with code uncertainty, this simplification allowed us to perform some analyses to examine the sensitivity to the model's assumptions (see Section 6.3). Much of the theory of Section 4 simplifies as a result. For example, the only correlation function to specify is $c_2(\cdot, \cdot)$. We first assume the simple product form (4) with two roughness parameters (ω_1, ω_2), corresponding to the directions parallel and perpendicular to the plume axis, as the prevailing wind and the cross-wind are believed to affect the deposition differently.

From the original 695 measurements, a subset of size 10 was chosen to represent a small sample of observed data similar to that which might be collected from ground measurements shortly after an accident. The points were chosen reasonably close to the source, but to avoid clustering were constrained so that each point is at least five measurement points from every other selected point. Clustered points lead to redundant information, which would make the estimation of the model hyperparameters very difficult for such small data sets, and the constraint ensures that the data are relatively dispersed. Additional points were chosen similarly and added to this subset, giving subsets of size 10, 15, 20 and 25 points. The point furthest from the source was deliberately included in the 25-point data set.

6.2. Results

Conditionally on each of the data sets, posterior means and variances of $z(\mathbf{x})$ were calculated for all the 670 'unobserved' points, and the accuracy was assessed on the basis of the true values at these points. The following strategies were compared:

(a) strategy 1 — using a Gaussian process interpolation of the physical observations alone, taking into account measurement errors, but making no use of the Gaussian plume model;
(b) strategy 2 — using Bayesian calibration and model inadequacy correction, as described in Section 4.6;

(c) strategy 3 — using the Gaussian plume model with 'plug-in' input parameters. The physical data are not interpolated in any way. Instead, we select the input parameters by minimizing the sum of squared differences between the model and the data.

The data that were used for each strategy were the same, comprising from 10 to 25 of the original data points. Table 1 gives the root-mean-squared errors RMSE of prediction for each of these strategies. For comparison, RMSE = 0.84 is obtained using the code with input parameters fixed at their prior mean. Strategy 1 achieves little improvement over this value for the data samples used. Strategy 3 is the kind of 'best-fitting' calibration technique that is often used in practice. Given enough data points with which to calibrate the model, this method improves on the use of the code with prior mean for $\boldsymbol{\theta}$, but strategy 2 is even better, because it also takes account of model inadequacy. A comparison of strategies 1 and 3 shows that, when the number of observations is small, the use of the code (suitably calibrated) is more accurate than simply interpolating the data.

To assess the significance of the improvement here, we note that the observations are log-depositions. So an error of 0.82 (strategy 3, $n = 10$) corresponds to errors in predicting deposition by a factor of $\exp(0.82) = 2.3$, and reducing this to 0.42 (strategy 2, $n = 10$) cuts the error in predicting deposition to a factor of 1.5. This is a genuinely useful improvement in the context of radiological protection.

Table 1. Root-mean-squared errors based on *n* observations

Strategy	*RMSEs for the following values of n:*			
	n = 10	*n = 15*	*n = 20*	*n = 25*
1	0.75	0.76	0.86	0.79
2	0.42	0.41	0.37	0.36
3	0.82	0.79	0.76	0.66

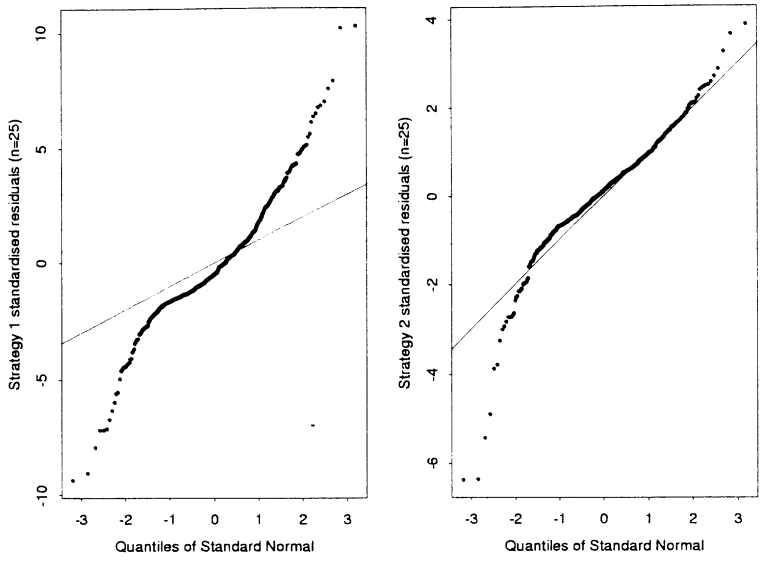

Fig. 2. Quantile–quantile plots for strategies 1 and 2 with $n = 25$

The quantile–quantile plots in Fig. 2 correspond to the standardized residuals based on strategies 1 and 2 in the case $n = 25$. The plot for strategy 2 clearly shows a better fit of the data to the predictive distribution than is achieved by strategy 1. However, both plots show heavy-tailed characteristics, and this feature is explored further in the following section.

For larger data sets we would expect strategies 1 and 2 to produce similar results. However, good predictions from interpolations of the data alone rely much more on an even distribution of design points over the prediction region. In most applications, optimal designs for physical observations are not practical. Our method makes a greater use of the code in regions where there are few physical data points and the uncertainty about model inadequacy in these regions is reflected in the posterior variance. Further evidence of predictive improvements using Bayesian calibration and model inadequacy correction may be found in a second example described in Kennedy and O'Hagan (2000b).

6.3. Sensitivity to modelling assumptions

So far, we have made various modelling choices, particularly in relation to the correlation function, which will not be appropriate for all applications. We now briefly examine how some alternative plausible modelling assumptions affect inferences in the case of the 25-point Tomsk data. Further details are given in Kennedy and O'Hagan (2000b).

The model described above (strategy 2) will be referred to as M1. Three alternative models are outlined below. In model M2 we relax the assumption that the hyperparameters are fixed, in model M3 we use an alternative functional form for the correlation function and in model M4 we consider the isotropic form of the Gaussian correlation function.

6.3.1. Model M2: integration with respect to the roughness parameters

It was suggested in Section 4.5 that fixing hyperparameters at the posterior modal values, rather than treating them as uncertain, is an acceptable simplification of the model. Often in models of this kind inferences (especially posterior variances) are sensitive to the choice of the roughness parameters in the correlation function. To take more account of the uncertainty about these parameters, we used a crude numerical method to integrate over ω_1, ω_2 in calculating the posterior predictive means and variances. These represent the roughness parameters in our non-isotropic product of one-dimensional correlation functions.

6.3.2. Model M3: isotropic Matérn correlation

The exponential form of the correlation function is appropriate if the inadequacy function is analytic and therefore may not be the best for modelling physical systems. We carried out an analysis which was identical with that described above but using the isotropic Matérn correlation function suggested by Handcock and Wallis (1994).

6.3.3. Model M4: isotropic Gaussian correlation function

The final model variation considered uses the isotropic Gaussian correlation function $c(\mathbf{x}, \mathbf{x}') = \exp(-b|\mathbf{x} - \mathbf{x}'|^2)$. This is equivalent to assuming that $\omega_1 = \omega_2$ in model M1. The estimated values of these parameters under M1 differ by a factor of 20. Under model M4 we might therefore expect to see inferences that are quite different from those which we obtained with M1 if there is sensitivity to these roughness parameters.

These analyses suggest that

(a) any improvement due to integrating with respect to the hyperparameters, as opposed to maximizing, is likely to be small and
(b) the effect of using alternative covariance structures is also small.

However, we do not propose that any of these models fits perfectly. Plots of prediction errors suggest that the true deposition surface exhibits local features that we are failing to predict. In some sense, all the models have predictive distributions with tails that are too thin. This is an area for on-going investigation but is likely to be application specific.

7. Conclusions and further work

We have presented a Bayesian approach to calibrating a computer code by using observations from the real process, and subsequent prediction and uncertainty analysis of the process which corrects for model inadequacy. The posterior summaries take account of all remaining sources of uncertainty.

We have treated the code as a black box, and the methods described in this paper are applicable to computer codes of arbitrary complexity.

As we have already mentioned in Section 5.1, important questions remain about the choice of design points. The physical observation sites will often be limited, as in the example presented here, but there will be situations in which these may be controlled. The designs that are already used to learn about code uncertainty might be used for D_2 to learn about model inadequacy. The problem of choosing designs D_1 and D_2 to give good calibration of the model is more difficult. The Cartesian product designs for D_1 described in Section 5 work well in the example of Section 6, in which the design D_2 is assumed to be fixed. These designs also facilitate the computations as outlined in Section 5.3.

We have carried out integration with respect to $\boldsymbol{\theta}$ by simple quadrature, which is feasible for low dimensional $\boldsymbol{\theta}$ but would become impractical with larger numbers of calibration parameters. The obvious approach then is Markov chain Monte Carlo sampling. However, the calibration distribution $p(\boldsymbol{\theta}|\hat{\phi}, \mathbf{d})$ is a complicated function of $\boldsymbol{\theta}$, and it would appear to be difficult to simulate from this distribution by using Markov chain Monte Carlo methods. Nevertheless, in our examples we have found it to be reasonably well approximated by a normal distribution (convergence of our iterative quadrature relies on a normal approximation). If this is true for high dimensional $\boldsymbol{\theta}$ it should be possible to use an approximate Markov chain Monte Carlo integration method.

In our examples, we have not considered the more important case in which the code output is multivariate. Effective calibration should ideally use all available code outputs and corresponding physical measurements. For example, in the nuclear accident application the National Radiological Protection Board would make a large number of air concentration measurements some time before the first ground deposition measurements are available. The air measurements can provide information about the unknown source terms and therefore should be used in the analysis. The use of multiple code outputs and measurements is a topic for future research. It was suggested in Section 4.1 that multivariate outputs might be handled simply by creating additional input parameters. In the nuclear application we could in principle treat 'measurement type' and 'radionuclide' as inputs. However, for these types of input it would not be reasonable to make the kind of assumptions that we make about the correlation structure.

We have discussed quite extensively the question of alternative covariance structures.

Experiments reported in Section 6.3 indicated some degree of robustness, but also a need to consider models that allow for more localized structure. We need to explore our methods with more and varied applications.

Within our Gaussian process framework, it would be relatively straightforward to accommodate observations of derivatives of the code, as in O'Hagan (1992). Derivatives are sometimes available, but we have not examined this so far. Other more speculative topics for future research include opening up the black box, discussed in Section 1.1, and applications with high dimensional $\boldsymbol{\theta}$. Our examples hitherto have not gone beyond a few dimensions, yet calibration problems may have many calibration parameters to 'fit'. The simulation-based approach of Romanowicz *et al*. (1994) can tackle high dimensional $\boldsymbol{\theta}$ but does not allow for model inadequacy. We suspect that a preliminary dimension reduction exercise, as in Craig *et al*. (1999), offers the most promising approach in such cases.

Acknowledgements

This research was supported by research grant GR/K54557 from the Engineering and Physical Sciences Research Council, UK, with additional financial contributions from the National Radiological Protection Board and the Environment Agency. We thank Neil Higgins, Tom Charnock and their colleagues at the National Radiological Protection Board for providing us with the data and for advice on the use of the Gaussian plume model. We are also genuinely grateful for many very helpful and pertinent comments from referees, which have led us to improve this paper in numerous ways.

References

Aitchison, J. and Dunsmore, I. R. (1975) *Statistical Prediction Analysis*. Cambridge: Cambridge University Press.

Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J. and Welch, W. J. (1998) Circuit optimization via sequential computer experiments: design of an output buffer. *Appl. Statist*., **47**, 31–48.

Bates, R. A., Buck, R. J., Riccomagno, E. and Wynn, H. P. (1996) Experimental design and observation for large systems. *J. R. Statist. Soc.* B, **58**, 77–94.

Bernardo, M. C., Buck, R. J., Liu, L., Nazaret, W. A., Sacks, J. and Welch, W. J. (1992) Integrated circuit design optimization using a sequential strategy. *IEEE Trans. Comput. Aid. Des*., **11**, 361–372.

Chilès, J. and Delfiner, P. (1999) *Geostatistics: Modeling Spatial Uncertainty*. New York: Wiley.

Clarke, R. H. (1979) *The First Report of a Working Group on Atmospheric Dispersion: a Model for Short and Medium Range Dispersion of Radionuclides Released to the Atmosphere*. London: Her Majesty's Stationery Office.

Cox, D. D., Park, J. S., Sacks, J. and Singer, C. E. (1992) Tuning complex computer codes to data. In *Proc. 23rd Symp. Interface of Computing Science and Statistics, April 21st–24th, 1991, Seattle*, pp. 266–271. Fairfax Station: Interface Foundation.

Cox, D. D., Park, J. S. and Singer, C. E. (1996) A statistical method for tuning a computer code to a data base. *Technical Report 96-3*. Department of Statistics, Rice University, Houston.

Craig, P. S., Goldstein, M., Rougier, J. C. and Seheult, A. H. (2001) Bayesian forecasting using large computer models. *J. Am. Statist. Ass*., **96**, in the press.

Craig, P. S., Goldstein, M., Seheult, A. H. and Smith, J. A. (1996) Bayes linear strategies for matching hydrocarbon reservoir history. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 69–95. Oxford: Oxford University Press.

Cressie, N. A. C. (1991) *Statistics for Spatial Data*. New York: Wiley.

Crick, M. J., Hofer, E., Jones, J. A. and Haywood, S. M. (1988) Uncertainty analysis of the foodchain and atmospheric dispersion modules of MARC. *Technical Report NRPB-R184*. National Radiological Protection Board, Didcot.

Currin, C., Mitchell, T., Morris, M. and Ylvisaker, D. (1991) Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Statist. Ass*., **86**, 953–963.

Dey, D., Müller, P. and Sinha, D. (eds) (1998) *Practical Nonparametric and Semiparametric Bayesian Statistics*. New York: Springer.

Diaconis, P. (1988) Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics IV* (eds S. S. Gupta and J. Berger), vol. 1, pp. 163–175. New York: Springer.

Draper, D. (1995) Assessment and propagation of model uncertainty (with discussion). *J. R. Statist. Soc.* B, **57**, 45–97.

Draper, D., Pereira, A., Prado, P., Saltelli, A., Cheal, R., Eguilior, S., Mendes, B. and Tarantola, S. (1999) Scenario and parametric uncertainty in GESAMAC: a methodological study in nuclear waste disposal risk assessment. *Comput. Phys. Communs*, **117**, 142–155.

Evans, M. and Schwartz, T. (1995) Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems (with discussion). *Statist. Sci.*, **10**, 254–272.

Goldstein, M. (1986) Separating beliefs. In *Bayesian Inference and Decision Techniques, Essays in Honour of Bruno de Finetti* (eds P. K. Goel and A. Zellner). Amsterdam: North-Holland.

———(1988) Adjusting belief structures. *J. R. Statist. Soc.* B, **50**, 133–154.

Handcock, M. S. and Stein, M. L. (1993) A Bayesian analysis of kriging. *Technometrics*, **35**, 403–410.

Handcock, M. S. and Wallis, J. R. (1994) An approach to statistical spatial-temporal modelling of meteorological fields (with discussion) *J. Am. Statist. Ass.*, **89**, 368–390.

Haslett, J. and Raftery, A. E. (1989) Space-time modelling with long-memory dependence: assessing Ireland's wind power resource (with discussion). *Appl. Statist.*, **38**, 1–50.

Haylock, R. (1997) Bayesian inference about outputs of computationally expensive algorithms with uncertainty on the inputs. *PhD Thesis*. University of Nottingham, Nottingham.

Haylock, R. and O'Hagan, A. (1996) On inference for outputs of computationally expensive algorithms with uncertainty on the inputs. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 629–637. Oxford: Oxford University Press.

Helton, J. C. (1993) Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliab. Engng Syst. Safty*, **42**, 327–367.

Helton, J. C., Garner, J. W., McCurley, R. D. and Rudeen, D. K. (1991) Sensitivity analysis techniques and results for performance assessment at the waste isolation pilot plant. *Technical Report SAND90-7103*. Sandia National Laboratories, Albuquerque.

Homma, T. and Saltelli, A. (1996) Importance measures in global sensitivity analysis of model output. *Reliab. Engng Syst. Safty*, **52**, 1–17.

Iman, R. L. and Conover, W. J. (1980) Small-sample sensitivity analysis techniques for computer models, with an application to risk assessment (with discussion). *Communs Statist. Theory Meth.*, **9**, 1749–1874.

Jones, J. A. (1981) *The Second Report of a Working Group on Atmospheric Dispersion: a Procedure to Include Deposition in the Model for Short and Medium Range Atmospheric Dispersion of Radionuclides*. London: Her Majesty's Stationery Office.

Kennedy, M. C. and O'Hagan, A. (2000a) Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, **87**, 1–13.

———(2000b) Supplementary details on Bayesian calibration of computer codes. University of Sheffield, Sheffield. (Available from `http://www.shef.ac.uk/~st1ao/ps/calsup.ps`.)

Kimeldorf, G. S. and Wahba, G. (1970) A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, **41**, 495–502.

Liu, L. and Arjas, E. (1998) A Bayesian model for fatigue crack growth. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (eds D. Dey, P. Müller and D. Sinha), pp. 339–353. New York: Springer.

Matheron, G. (1963) Principles of geostatistics. *Econ. Geol.*, **58**, 1246–1266.

McKay, M. D., Conover, W. J. and Beckman, R. J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**, 239–245.

Morris, M. D. (1991) Factorial sampling plans for preliminary computational experiments. *Technometrics*, **33**, 161–174.

Morris, M. D. and Mitchell, T. J. (1995) Exploratory designs for computational experiments. *J. Statist. Planng Inf.*, **43**, 381–402.

Morris, M. D., Mitchell, T. J. and Ylvisaker, D. (1993) Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, **35**, 243–255.

Naylor, J. C. and Smith, A. F. M. (1982) Applications of a method for the efficient computation of posterior distributions. *Appl. Statist.*, **31**, 214–225.

Neal, R. (1996) *Bayesian Learning for Neural Networks*. New York: Springer.

———(1999) Regression and classification using Gaussian process priors (with discussion). In *Bayesian Statistics 6* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 475–501. Oxford: Oxford University Press.

Novak, E. (1988) Deterministic and stochastic error bounds in numerical analysis. *Lect. Notes Math.*, **1349**.

Oakley, J. E. and O'Hagan, A. (1998) Bayesian inference for the uncertainty distribution. *Technical Report*. Statistics Section, University of Nottingham, Nottingham.

Oehlert, G. W. (1993) Regional trends in sulfate wet deposition. *J. Am. Statist. Ass.*, **88**, 390–399.

O'Hagan, A. (1978) Curve fitting and optimal design for prediction (with discussion). *J. R. Statist. Soc.* B, **40**, 1–42.

———(1991) Bayes-Hermite quadrature. *J. Statist. Planng Inf.*, **29**, 245–260.

———(1992) Some Bayesian numerical analysis (with discussion). In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 345–363. Oxford: Oxford University Press.

———(1998) A Markov property for covariance structures. *Technical Report 98-13*. Statistics Section, University of Nottingham, Nottingham. (Available from `http://www.shef.ac.uk/~st1ao/ps/kron.ps`.)
O'Hagan, A., Kennedy, M. C. and Oakley, J. E. (1999) Uncertainty analysis and other inference tools for complex computer codes (with discussion). In *Bayesian Statistics 6* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 503–524. Oxford: Oxford University Press.
Omre, H. (1987) Bayesian kriging—merging observations and qualified guesses in kriging. *Math. Geol.*, **19**, 25–39.
Owen, A. B. (1992) A central limit for Latin hypercube sampling. *J. R. Statist. Soc.* B, **54**, 541–551.
Poole, D. and Raftery, A. E. (1998) Inference for deterministic simulation models: the Bayesian melding approach. *Technical Report 346*. Department of Statistics, University of Washington, Seattle.
Raftery, A. E., Givens, G. H. and Zeh, J. E. (1995) Inference from a deterministic population dynamics model for bowhead whales (with discussion). *J. Am. Statist. Ass.*, **90**, 402–430.
Rios Insua, D. and Müller, P. (1998) Feedforward neural networks for nonparametric regression. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (eds D. Dey, P. Müller and D. Sinha), pp. 181–193. New York: Springer.
Romanowicz, R., Beven, K. and Tawn, J. A. (1994) Evaluation of predictive uncertainty in nonlinear hydrological models using a Bayesian approach. In *Statistics for the Environment 2: Water Related Issues* (eds V. Barnett and K. F. Turkman), pp. 297–319. New York: Wiley.
Sacks, J., Schiller, S. B. and Welch, W. J. (1989) Designs for computer experiments. *Technometrics*, **31**, 41–47.
Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989) Design and analysis of computer experiments. *Statist. Sci.*, **4**, 409–435.
Saltelli, A., Chan, K. and Scott, E. M. (2000) (eds) *Sensitivity Analysis*. New York: Wiley.
Saltelli, A. and Sobol', I. M. (1995) About the use of rank transformation in sensitivity analysis of model output. *Reliab. Engng Syst. Safty*, **50**, 225–239.
Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary covariance structure. *J. Am. Statist. Ass.*, **87**, 108–119.
Schweder, T. and Hjort, N. L. (1996) Bayesian synthesis or likelihood synthesis—what does Borel's paradox say? *Rep. Int. Whaling Commissn*, **46**, 475–480.
Shershakov, V. M., Vakulovski, S. M., Borodin, R. V., Vozzhennikov, O. I., Gaziev, Y. L., Kosykh, V. S., Makhonto, V. S., Chumiciev, V. B., Korsakov, A. T., Martynenko, V. P. and Godko, A. (1995) Analysis and prognosis of radiation exposure following the accident at the Siberian chemical combine Tomsk-7. *Radian Protectn Dosim.*, **59**, 93–126.
Smith, M. and Kohn, R. (1998) Nonparametric estimation of irregular functions with independent or autocorrelated errors. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (eds D. Dey, P. Müller and D. Sinha), pp. 157–179. New York: Springer.
Stein, M. L. (1987) Large sample properties of simulation using latin hypercube sampling. *Technometrics*, **29**, 143–151.
———(1999) *Interpolation of Spatial Data: Some Theory for Kriging*. New York: Springer.
Vecchia, A. V. (1988) Estimation and identification for continuous spatial processes. *J. R. Statist. Soc.* B, **50**, 297–312.
Vidakovic, B. (1998) Wavelet-based nonparametric Bayes methods. In *Practical Nonparametric and Semiparametric Bayesian Statistics* (eds D. Dey, P. Müller and D. Sinha), pp. 133–155. New York: Springer.
Walker, S. G., Damien, P., Laud, P. W. and Smith, A. F. M. (1999) Bayesian nonparametric inference for random distributions and related functions (with discussion). *J. R. Statist. Soc.* B, **61**, 485–527.
Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D. (1992) Screening, predicting, and computer experiments. *Technometrics*, **34**, 15–25.
Wolpert, R. L. (1995) Comment on the paper by Raftery, Givens and Zeh. *J. Am. Statist. Ass.*, **90**, 426–427.
Wooff, D. A. (1992) [B/D] works. In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 851–859. Oxford: Oxford University Press.

Discussion on the paper by Kennedy and O'Hagan

H. P. Wynn (*University of Warwick, Coventry*)

It is one of the traditions of the Royal Statistical Society to be critical. I shall refrain from this because I consider this a landmark paper. I shall confine my remarks to a few points which may show a slightly different emphasis from that of the authors.

Consider a simple weighing problem in which two people stand together on the same bathroom scale. The single reading may be modelled as

$$y = \theta_1 + \theta_2 + \epsilon$$

where, assuming no zero correction, θ_1 and θ_2 are the unknown weights and ϵ is a measurement error. Suppose that ϵ is 0; then it still holds that the model is not identifiable in the classical sense. The serious point is that the models proposed by the authors have complex hierarchical structures and we should

perhaps investigate the identifiability. A lack of (likelihood) identifiability, as in the weighing problem, persists independently of the prior assumptions and will typically lead to inconsistent estimation in the asymptotic sense.

There is a related but more subtle problem. There may be many different interpolators of the same data. This holds even when the model class is restricted, for example to polynomials, and there is no error. This means that non-statistical criteria such as smoothness must be used to select an interpolator. In the 'computer experiments' method (Gaussian kriging) this smoothness is held by the assumptions on the correlation function. But even in standard but complex linear methods the need to add such criteria is evident.

Again related is the status of unknown parameters. My comments here are more of a question. Is there an ontological difference between different parameters? For example do quantities which are potentially measurable, or future values of observables, have a different status from parameters which are buried deep down in some hierarchical model? Do the weights θ_1 and θ_2 have a different status from that of some hyperparameter of a superpopulation from which they are assumed to be sampled? This issue arises in several areas as evidenced by the different terminology for the same entity: hidden variable, latent variable, state variable, hyperparameter. In some cases, such as when a hidden variable is unearthed in a Bayes network and attributed to a real cause, the status seems higher. When the parameter is simply included to 'adjust' or 'shrink' the model in some way then the status seems lower.

In studying any class of computer models it is very important to work closely with the scientists or engineers. Large computer simulators are playing an increasing role in research and industry from models of the gulf stream to the control of chemical plants and automotive design. They are used for scientific study and for optimization. Very importantly they are used for supporting decisions in safety critical areas such as, if there is another Chernobyl accident, to determine the path of the radiation plume. In all cases, but particularly the latter, speed is of the essence in replacing or adapting computer models which have long run times.

Computer models are of different types: linear and non-linear ordinary or partial differential equations and so on. They are solved (integrated) by a variety of methods. It is useful to know what is going on inside the black box. For example if we can obtain the partial derivatives (sensitivities and Jacobians) of the outputs with respect to the parameters then we can construct the information matrix, as in non-linear regression, and hence approximations to the likelihood, approximate confidence intervals, tests and so on. We may even obtain higher derivatives and then use second-order corrections such as the saddlepoint. For dynamic models all these change with time and, for example, we typically learn more from the transient behaviour than from the steady state. If the black box is a solver which gives the output as the integral of the system then without too much additional analysis we can obtain these first- and second-order sensitivities. Three basic methods are available: numerical, symbolic or what is termed automatic differentiation. It is also possible to address the identifiability issues discussed earlier via these sensitivities by studying ranks of matrices.

As mentioned, in a real time safety critical area any emulator must run fast. It may also be the case that additional inputs, such as radiation ground readings, are fed to the emulator, or a model–emulator hybrid. In such cases the Kalman filter or the extended Kalman filter, in the non-linear case, is a leading candidate as the engine for the emulator. (Note that the extended Kalman filter requires the Jacobians referred to above.) Some of the Bayes updating in the paper has a Kalman filter flavour and to widen the comment these and other ideas from system theory would enhance understanding.

I have great pleasure in proposing the vote of thanks.

Philip J. Brown (*University of Kent at Canterbury*)

My interest in this paper was initially not through computer models but through calibration. The notion of calibration here is not quite the same as that employed in chemometrics, with which I am more familiar. One similarity with chemometric Bayesian calibration is the key idea of utilizing the uncertainty in the calibration input $\boldsymbol{\theta}$ through its posterior distribution, but there are several differences. It is an intriguing idea to try to model models. The essential feature of computer code models as used here seems to be the ability to learn about and to *approximate* the computer code through computer runs. Thus the code acts as a stepping-stone to reality.

In chemometrics over the past 10 years perhaps the greatest gains in accuracy of estimation of calibration inputs have arisen from instruments becoming highly multivariate perhaps with several hundreds of responses in the case of spectroscopic instruments; see for example Brown (1993). Typically the same calibration input $\boldsymbol{\theta}$ influences each response so that the multivariate response acts to replicate

information. Even when their $\boldsymbol{\theta}$-value cannot be assumed to be the same they are often related. This might be an embarrassment if the authors take as they have the stance of regarding the calibration input $\boldsymbol{\theta}$ as a convenient parameter which offers model flexibility and does not necessarily relate to 'truth'. I wonder how the current approach should generalize to the multivariate case. A simple generalization in the notation of Section 3.1 would be for $q' \times 1$ vector $\mathbf{f}$ to be generalized to the

$$\mathbf{f}(\cdot) \sim N\{\mathbf{M}(\cdot),\, r(\cdot)\boldsymbol{\Sigma}\}$$

generalizing the hierarchical form (3) in the covariance structure and where the $q' \times q'$ matrix $\boldsymbol{\Sigma}$ would be taken as inverted Wishart at the next stage; here equal prior means $\mathbf{M}(\cdot) = m(\cdot)\mathbf{1}$ may be too restrictive.

The flexibility of the Gaussian process has much to commend it. There is a rich choice of covariance functions but this choice is not always straightforward. Neal (1999) gives some considerations. The authors mentioned the alternative approach of regression on basis functions. The trivially simplest basis, a linear regression, translates into a quadratic covariance function. The Tomsk-7 chemical plant emissions data show a predictable gradient in its contours moving away in a north-easterly direction from the source. In the case of code uncertainty, or what I prefer to think of as code *approximation*, might such non-stationarity in x-space be naturally modelled by a quadratic term in addition to the exponential covariance term? The general issue arises whether to include such terms in the mean or covariance structure. It seems to me that the basis function regression approach is more direct, intuitive and simpler to apply with its conditional independence formulation. When it comes to pure modelling of space–time data on particulate pollution collected at a grid of monitoring stations for model robustness we have preferred to assign covariance structure at the secondary level of the scale matrix of the inverted Wishart prior. We have then been able to inject non-stationarity at this secondary level through a Sampson–Guttorp form of distortion of an isotropic covariance in a latent co-ordinate space. Early multivariate work for several pollutants is given in Brown *et al.* (1994a); more recent work accommodates missing response data with a monotone structure; see Kibria *et al.* (2000), utilizing the generalized inverted Wishart prior of Brown *et al.* (1994b).

The authors have done a very good job at spiking criticism by emphasizing that models are not to be believed and certainly not their models. We should judge the predictions from the model. This is fine but is one prediction problem carefully extracted enough? Let me turn again to the application. Reading Shershakov *et al.* (1995) I began to think about the nature of this prediction problem. If I have gleaned correctly, the aerial data were taken by flight runs back and forth at an altitude of between 70 m and 100 m, which recorded γ-radiation. They were all taken around the same time. There were also comparative measurements at ground level and many snow and soil samples were taken. My question is whether the careful extraction of 10, 15, 20 or 25 aerial observations adequately reflects the sampling schedule in time and space and the real life prediction problem. How transferable is the methodology to the next radiochemical accident?

The authors tackle the problem without code uncertainty so that there is just the Gaussian process of the model inadequacy $\delta(\mathbf{x})$ with $\eta(\mathbf{x}, \boldsymbol{\theta})$ computed directly in equation (5). Still there is the issue of plug-in estimation. This is not second order for predictive uncertainty. The approach is sensible as a first step, but in this case the problem is just n and not $n+N$ dimensional and so would simulation methods be feasible?

A down side to the Gaussian process formulation (with code approximation) as mentioned is the need to invert an $(N+n) \times (N+n)$ matrix $\mathbf{V}_d(\boldsymbol{\theta})$ where N is potentially very large. Apart from the order $(N+n)^3$ of operations, the numerical stability will be affected by the condition number of the matrix. This in turn will be affected by how predictable the targets are. In essence inessential variability for example through trends is better located in the mean function. This can also be influenced by the choice of code design and I presume that Cartesian product forms of code design help here but I would welcome any comments that the authors may have on this aspect.

The authors noted the poor tail behaviour of the standardized residuals as shown in Fig. 2. In the more general problem with code uncertainty, I wonder whether this would be favourably influenced by a discrepancy covariance function $c_2(x, x')$ which does not have the same principal axes as those of the code uncertainty covariances. One could think of generalizing the Gaussian process by scale mixing or the use of stable laws, but reading Shershakov *et al.* (1995) again I suspect that the heterogeneity is caused by 'hot' particles, and this may need prizing open the black box for proper modelling.

This paper offers a challenging new approach to the Bayesian calibration of computer models. I have great pleasure in seconding the vote of thanks.

The vote of thanks was passed by acclamation.

Clive Anderson (*University of Sheffield*)

This paper is welcome because it addresses an important need—to attach uncertainties to predictions from numerical models—and it says illuminating things about it. The problem is currently topical in relation to large scale models for such things as climate change, pollution and weather forecasting, but it is relevant at all levels because it is central to the construction of more faithful scientific models for almost any system—models that combine existing (often very extensive) scientific understanding with a recognition that nevertheless variability and uncertainty may remain.

I have some specific comments and questions, particularly with large models in mind.

(a) In environmental science the output from many numerical models is a map or a temporal sequence of maps. Recognition of the specific spatial–temporal structure implied by this for the covariances of the Gaussian random fields seems important for realism and for exploitation of sparse data.
(b) Prediction *per se* is not the only use for numerical models. They are often used as test-beds for theories about submechanisms. This suggests an extension of the authors' formulation to one in which the computer code consists of several linked parts. Inference techniques that could pinpoint the uncertainty arising from each part, and so identify areas needing further research, would be useful.
(c) Users of numerical models, particularly those producing time series output in the atmospheric and oceanographic sciences, often carry out *data assimilation*, meaning dynamic adjustment of the model in the light of new data to improve predictions. In the paper's terms this is just calibration, but with the extra feature that it is carried out repeatedly. The possibilities of exploiting data sequentially seem worth exploring within the authors' framework.
(d) To extend the authors' approach to really large numerical models it seems likely that some simplification will be needed. With this in mind we might wonder about the rather radical change of sacrificing the assumption that η is stochastic (as done in the Tomsk illustration in the paper). In general some of the parameters and the idea of code uncertainty would disappear, presumably with some gain in tractability, but model inadequacy and other sources of uncertainty could still be dealt with. If the resulting theory had some similarities with Professor Beven's widely used methodology, but with a 'generalized likelihood' founded on more explicit structural assumptions and incorporating model inadequacy, it would be attractive. Have the authors explored any such ideas?

J. C. Rougier (*Durham University*)

We are working in a similar area to that of the authors: large computer models. I would like to inject a cautionary note, based on our own experiences in doing the kind of computations that the authors have done in this paper. Phil Brown alluded to the problem of inverting the data variance, which is an $(n+N)^3$-operation. It is suggested in the paper that this can become computationally feasible if Cartesian product designs are adopted on the $\mathbf{x}^*$-inputs and the $\mathbf{t}$-inputs. I believe that this was done in the examples in the paper.

The cautionary note that I raise here is that we have experimented with the Cartesian product design because of course it gives a Kronecker product structure for the variance of the data. Instead of an $(n+N)^3$-operation, the inverse is an (n^3+N^3)-operation, so it is extremely beneficial computationally. The problem with a Cartesian product design is that the same values for $\mathbf{x}^*$ and the same values for $\mathbf{t}$ keep repeating. This means that there are only a small number of unique $\mathbf{x}^*$-values and a small number of unique $\mathbf{t}$-values in the design over which the computer simulator is run.

If there are only a small number of unique points, correspondingly there are a relatively small number of interpoint distances. This a problem because when we want to estimate the hyperparameters a big range of different interpoint distances is needed to obtain a good empirical semivariogram which will allow the parameters ρ, σ and ω to be determined. If there are only a small number of interpoint distances, having adopted a Cartesian product structure because it is computationally efficient, we can end up with imprecise estimates of the hyperparameters and, in particular, with large correlations across the hyperparameters. This is our experience.

I would like to ask the authors whether they could compute the observed information matrix for the hyperparameters and, if so, how they would respond to the observation that the variances were large and the correlations were also large.

Peter J. Diggle (*Lancaster University*)

My comments concern the specification of the covariance structure of the Gaussian process which the authors use as a (stochastic) model for the (deterministic) response surface generated by the underlying computer code. Models of this kind have been studied extensively within the branch of spatial statistics usually known as *geostatistics*.

(a) In equation (4), the authors use the correlation function

$$r(u) = \exp\left(-\sum \omega_j |u_j|^{\alpha}\right),$$

with $\alpha = 2$ initially, and refer to the ω_j as roughness parameters. This terminology is misleading. A more natural parameterization would be

$$r(u) = \exp\left(-\sum |u_j/\omega_j|^{\alpha}\right),$$

in which case the ω_j have the same physical dimensions as u_j, and a direct interpretation as a set of scaling factors.

Diggle *et al.* (1998) also used this 'powered exponential' family but were rightly criticized for doing so because, in terms of the implied roughness of the underlying Gaussian process, it is rather inflexible; for $0 < \alpha < 2$ the process is mean square continuous but non-differentiable, whereas for $\alpha = 2$ (the maximum allowable value for a legitimate model) the process is mean square infinitely differentiable. In contrast, the Matérn family of correlation functions which the authors later mention as an alternative is also indexed by two parameters but has the useful property that the parameter analogous to α controls the number of times the underlying process is mean square differentiable.

(b) The relative insensitivity of point predictions of $\eta(x)$ to the choice of correlation family is also a feature of geostatistical methods; experience there suggests that this might no longer hold if the models are used to predict non-linear functionals of the response surface $\eta(x)$.

(c) The specification of priors for correlation parameters is not straightforward. Our experience has been that superficially innocuous changes to vague priors can materially affect posterior predictive distributions (Diggle and Ribeiro, 2001). Also, prior independence of α and the ω_j is questionable, although my preference would be to restrict attention to a few discrete values of α with (in the case of the Matérn family) qualitatively different interpretations.

(d) The scientific context in which a specific computer model is developed may suggest that the character of the response surface $\eta(x)$ is different in different parts of the x-space. If so, it might be better to use contextual knowledge to choose an appropriate correlation family, and to let the parameters of the correlation function operate locally, rather than to estimate these parameters globally from sparse data. In similar vein, practical geostatisticians often implement their predictions locally, using only data from points close to x to predict $\eta(x)$, with the aim of making their predictions more robust to model misspecification.

Michael Goldstein (*University of Durham*)

I would like to thank the authors for an excellent treatment of a topic of growing practical importance. The type of full Bayes analysis developed in this paper will undoubtedly be appropriate and valuable whenever

(a) we can give a meaningful prior and likelihood specification for the model in the suggested form and
(b) the resulting full analysis is tractable.

However, there may be substantial difficulties in carrying out the program for large computer models, where the input and output spaces are very high dimensional, and where we may be able to make only a relatively small number of evaluations of the computer code (so that the prior specification and choice of computer evaluations is very important). But, as the authors note, often 'the purpose of calibration is to use the calibrated model for predicting the real process'. In such cases, it is useful to be aware that there are alternative approaches to prediction which do not require explicit preliminary calibration. One such approach is implemented in Craig *et al.* (2001), where, using a similar prior formalism to that of the current paper, we

(a) use evaluations of the computer code to adjust beliefs about the computer model,
(b) assess beliefs about the relationship between observed and unobserved process values based on beliefs derived in (a) and
(c) predict unobserved process values given observed process values using joint beliefs formed in (b).

The advantage of this approach is that it is often much easier to find simple, fairly robust, approximations when the analysis is constructed as above (for example, in Craig *et al.* (2001), we simplify by using various Bayes linear assessments) than is the case for the calibration-led formulation in the current paper. Therefore, the alternative approach should allow us to analyse much larger problems, and even to address the crucial design issue of identifying good choices of computer evaluations to improve prediction, for such large computer models.

Wilfrid S. Kendall (*University of Warwick, Coventry*)
In the course of working with perfect simulation techniques I also have become interested in the behaviour of complicated computer programs! Direct simulation methods lead to relatively simple code, but matters become progressively more involved as one seeks more developed methods. The *coupling from the past* (CFTP) method introduced by Propp and Wilson (1996) relies on correct implementation of a *coupling* construction (simulation practitioners will be more familiar with 'coupling' in its special case of antithetic variable simulation). When correctly implemented, CFTP will deliver exact samples from the equilibrium distribution of suitable Markov chains, but correct implementation is non-trivial (in Kendall (1998) I coined the alternative term 'perfect simulation' deliberately to attract attention to the need for perfect implementation: see the introduction to Kendall and Møller (2000)).

Clearly we can assess the perfection of a CFTP implementation in a brutal but effective way by subjecting it to assault by a battery of statistical tests (e.g. Cai and Kendall (2001)). However, during development we need diagnostics which probe the algorithm in more detail. An ideal tool would be a statistical analogue of the `assert` macro from C, which tests for the truth of its argument and stops execution when the argument is false. While developing the algorithm underlying Kendall and Møller (2000) I used a crude statistical variation of `assert`: at each transition of the underlying coupled Markov chain I arranged for the transition (including a record of the current state) to be recorded in a log-file. Consequently it was possible to carry out a pseudolikelihood statistical analysis at the end of each run; during development this information could be interpreted to indicate *where* in the implementation there might be an error. Details can be found in Kendall and Møller (1999).

Correctness is different from calibration; moreover there has of course been earlier relevant work on testing exact hypotheses, and on searching for bugs in software and typographical errors in manuscripts. However, the requirements of complicated statistically related programs seem to deserve more detailed investigation: what I outline above was my own rough-and-ready initial attempt at a solution, but I would welcome comments from the authors (and others) on how to do better!

Peter Craig (*University of Durham*)
I would like to thank the authors for their paper on a statistical methodology which has a very wide area of potential application.

A fundamental problem with computer models is whether to treat uncertain inputs $\boldsymbol{\theta}$ as representations of physical quantities or as statistical parameters. The authors point out the difficulties with the former but the latter approach raises questions.

(a) Is the true value of $\boldsymbol{\theta}$ operationally defined? The authors suggest that true $\boldsymbol{\theta}$ is the 'best fit' in the sense of representing data faithfully according to the error structure (model inadequacy and observation error) specified for the residuals. In many traditional statistical contexts, where we can envisage a stream of data with unlimited effective degrees of freedom which eventually identifies all parameter values, such a definition works well even when the model is not correct, although non-linearity may lead to non-uniqueness. However, as well as being non-linear, many computer model applications, such as the example in the paper, have an inherent upper bound on the effective number of degrees of freedom. In particular, the value of $\boldsymbol{\theta}$ defined this way is dependent on the structure and parameters for model inadequacy, which are unlikely to be completely identifiable.
(b) If the definition of the true value of $\boldsymbol{\theta}$ is very complicated or is lacking, how can we elicit a distribution for $\boldsymbol{\theta}$ from experts? Coming from different backgrounds, it seems unlikely that the

elicitor and elicitee will have the same meaning in mind for $\boldsymbol{\theta}$. In the example, were the National Radiological Protection Board told that the $\boldsymbol{\theta}$ for which they were providing beliefs was not the physical version but some statistically motivated version? If so, how was the definition explained to them and did they find the concept easy to grasp?

(c) What should we do in applications where there are informative prior beliefs about at least some aspects of the physical quantities being represented by $\boldsymbol{\theta}$? If we formally separate the two ideas of what $\boldsymbol{\theta}$ means, we might model the difference between the two. This still requires an operational definition for the statistical parameter but would then allow elicitation of the difference between physical quantity and statistical parameter.

The following contributions were received in writing after the meeting.

Keith Beven (*Lancaster University*)

This comment will focus on the concept of the model inadequacy function. In the environmental (and other) sciences, models are used because, by including knowledge of the (non-linear) dynamics in prediction to other cases, the predictions will be in some sense more rigorous. The model itself is therefore being used as a tool for the extrapolation of understanding to predictive situations. The danger is that if either the model structure or parameter values are incorrect the predictions may be inaccurate. Thus adding a model inadequacy function or bias parameter into the likelihood function would seem to be an excellent idea.

However, experience with Monte Carlo simulations for complex environmental models used in the generalized likelihood uncertainty estimate (GLUE) methodology (Beven *et al.* (2000) and references therein) suggests that it may be very difficult to formulate an inadequacy function, especially for models that make predictions in both time and space. In addition, parameter interactions and covariation leading to models that fit the data well can be markedly different in different parts of the parameter space, leading to a wide selection of different models that give an acceptable fit to the available data from the process point of view. Model inadequacy might also be expected to vary through the 'model space' (of model structures and parameter sets) and through time and space for individual models in that space. It would be possible, given sufficient computer time, to estimate an inadequacy function for each of the individual models sampled (or at least the subset of those considered 'behavioural' in fitting the data). This was done within the GLUE methodology, for example, by Romanowicz *et al.* (1994), who allowed for the simplest case of a mean bias in integrating over a likelihood function for a given set of model parameters.

Since each set of model parameter values may then have its own inadequacy function, inadequacy then becomes an additional model component but one which may have no clear physical meaning. In addition, depending on the function chosen, the best joint model in calibration may not necessarily be the best physical model (or best model in prediction for other situations). From the physical point of view, model rejection might then be a better strategy than compensation for inadequacy.

Katherine Campbell and Michael D. McKay (*Los Alamos National Laboratory*)

This paper goes far beyond the well-studied problem of parameter uncertainty to formulate a method for the simultaneous estimation of multiple forms of uncertainty associated with model predictions. The area that still puzzles us most is the relationship between the model inputs and 'reality'. The complexity of this relationship is glossed over by the authors' two-part classification of the inputs into 'calibration inputs' *versus* 'variable inputs'. In particular, among the latter would be included well-known control inputs as well as poorly measured inputs (rainfall) and proxies for complex states of nature (atmospheric stability classes).

We would like to sketch an alternative taxonomy which clarifies, in our minds, some of the authors' well-taken points. We denote the factors driving reality by the triplet $(\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\theta})$, with corresponding model inputs $(\mathbf{a}, \mathbf{p}, \mathbf{t})$. Here $(\boldsymbol{\alpha}, \boldsymbol{\pi})$ corresponds to the authors' $\mathbf{x}$. The $\boldsymbol{\alpha}$ are *observable* control variables and have the same meaning for both model and reality, i.e. $\mathbf{a} = \boldsymbol{\alpha}$. The relationships between $\boldsymbol{\pi}$ and $\mathbf{p}$, and between $\boldsymbol{\theta}$ and $\mathbf{t}$, are more complex. We think of them as mappings $\mathbf{p} = \mathbf{P}(\boldsymbol{\pi})$ and $\mathbf{t} = \mathbf{T}(\boldsymbol{\theta})$ from high dimensional *unobservable* reality into low dimensional spaces of model parameters. Model-independent estimates $\mathbf{p}_i$ of $\mathbf{P}(\boldsymbol{\pi}_i)$ are available for the observed data, but the calibration variables $\mathbf{T}(\boldsymbol{\theta})$ must be estimated using the model output.

Now, the observations are modelled by

$$\begin{aligned} z_i &= \zeta(\boldsymbol{\alpha}_i, \boldsymbol{\pi}_i, \boldsymbol{\theta}) + e_i \\ &= \rho\, \eta\{\boldsymbol{\alpha}_i, \mathbf{P}(\boldsymbol{\pi}_i), \mathbf{T}(\boldsymbol{\theta})\} + \delta\{\boldsymbol{\alpha}_i, \mathbf{P}(\boldsymbol{\pi}_i)\} + e_i \\ &= \rho\, \eta(\mathbf{a}_i, \mathbf{p}_i, \mathbf{t}) + \delta(\mathbf{a}_i, \mathbf{p}_i) + e_i. \end{aligned}$$

This version of the authors' equation (5) clarifies several points. First, we see that what the authors call 'residual uncertainty' arises from the limitations of the dimension reducing mappings $\mathbf{T}$ and $\mathbf{P}$. Second, this version makes it clear that reality does depend on some underlying factors $\boldsymbol{\theta}$, factors which the modeller hopes to capture in his model by means of $\mathbf{t}$. The omission of $\boldsymbol{\theta}$ from the authors' representation of reality by $\zeta(\mathbf{x})$ leads to confusion about the meaning of the 'true value' of $\boldsymbol{\theta}$. Although we, also, take $\boldsymbol{\theta}$ to represent something 'true', for modelling purposes we can use only its projection $\mathbf{T}(\boldsymbol{\theta})$. Thus, for us, $\mathbf{T}(\boldsymbol{\theta})$ plays the role assigned to $\boldsymbol{\theta}$ by the authors. Finally, therefore, the calibration problem is to estimate a posterior distribution for the model-dependent parameter $\mathbf{T}(\boldsymbol{\theta})$ which, together with the estimated variable inputs $\mathbf{p}_i$, produces model behaviour consistent with the data $\mathbf{d}$.

Peter Challenor (*Southampton Oceanography Centre*)
I would like to comment on Section 4.2 and the problem of whether parameters should be fixed at their true values or estimated as part of the calibration process. The authors put forward the argument that we should estimate parameters to enable a better fit to be obtained and hence a more accurate prediction. This is a sensible thing to do if the sole purpose of the calibration is to produce a good predictor, but there is another aspect to this problem. The rationale for using a complex code (as opposed to an empirical regression model) is normally that it encapsulates our understanding of a physical process. Such physically based models are dependent on parameters that have a real significance; for example, one parameter might be g, the acceleration due to gravity. If the calibration of the model results in estimates of these parameters that differ in an unrealistic way from the true values, for example if g is estimated as $20\ \mathrm{m\ s^{-2}}$ rather than $9.81\ \mathrm{m\ s^{-2}}$, this not only implies that running the code with the true value will give a poor prediction but also the rationale for using that code for prediction may itself be in error. The calibration is in effect acting as a test of how appropriate it is to use the code to model the physical situation. Little work has been done on the goodness of fit (or goodness of physics) for complex codes and comparing 'true' and estimated parameter values may be a useful approach.

R. M. Cooke (*Delft University of Technology*)
I congratulate the authors on their paper. It is not only a very useful introduction to this new and important area but also provides results which will surely provoke further interest. It is especially laudable that statisticians interact with modellers to understand better the role of uncertainty in complicated engineering models. My comments are referenced to the sections of the paper.

Section 4.3: true parameter values
The argument here does not convince me. Saying to the modellers 'the wind speed is measured as $2\ \mathrm{m\ s^{-1}}$, but I obtain a better fit with my model if I use a wind speed of $10\ \mathrm{m\ s^{-1}}$' will not encourage them to take the model seriously. If a 'calibration variable' has a value which can be measured satisfactorily, then any difference between model predictions and realizations should be regarded as model inadequacy (even if it is not independent of code input), in my view.

Section 4.2: model
When saying that the realizations $\zeta(\mathbf{x}_i)$ are a function of $\mathbf{x}_i$, we are effectively assuming that repeated realizations under conditions $\mathbf{x}_i$ will give the same results. If this is not true then the error in equation (5) has more structure than assumed (zero mean and variance not depending on the variable inputs $\mathbf{x}_i$). I can easily envisage this being strongly wrong in the example the authors discuss. I am a little confused what I should 'believe' about equation (6):

$$\zeta(\mathbf{x}_i) = \rho\, \eta(\mathbf{x}_i, \theta) + \delta(\mathbf{x}_i).$$

Of course I can always find ρ and $\delta(\mathbf{x}_i)$ for which this holds, e.g. $\rho = 0$ and $\delta(\mathbf{x}_i) = \zeta(\mathbf{x}_i)$. Why should I believe anything else? The point must be that we want $\delta(\mathbf{x}_i)$ to be small, but is that believing or wishing (an important distinction for a Bayesian)? Consider the equation $2 = A + B$; now what are your beliefs about A and B? If A and B are not independently observable then this question does not make any sense to me.

Section 6.2: results

The results look impressive, but before being completely overwhelmed I would like to know more about the defeated strategies. Regarding strategy 1, I do not know what a 'Gaussian process interpolation' is, but clearly any method which is the same in the crosswind and downwind directions will not have much chance. It seems strange that the root-mean-squared error increases as n goes from 10 to 20. Regarding strategy 3, are we minimizing squared differences or square log-differences? If the former, then only very near field results will have any effect. It is reasonable to choose the observations near field, as this mimics practice, and thus results for strategy 3 look very promising.

N. A. Higgins and J. A. Jones (*National Radiological Protection Board, Didcot*)

In the month that the Ukrainian nuclear power-station at Chernobyl finally closed, this paper is both timely and important. The paper presents a method of using measured results effectively and efficiently to improve on the predictions that are available from models, thereby increasing confidence in the predictions and the decisions that potentially follow from them. Previous assessment strategies have relied on simple 'plug-in' parameters to a Gaussian plume model, as Kennedy and O'Hagan refer to them. For example, much work has been carried out on global parameter fitting through least squares methods to determine the plug-in values. However, these approaches could not accommodate model inadequacy, a feature that is prevalent when assessing an accident owing to the limitations imposed by ignorance of the conditions of the release and the consequent need to use simple models with simple data demands.

The work of Kennedy and O'Hagan points to a way forward in the assessment of the consequences of a nuclear accident and complements studies at the National Radiological Protection Board using a variety of geostatistical techniques. Geostatistical techniques are analogous to the Gaussian process interpolation of Kennedy and O'Hagan but may be used in conjunction with a Gaussian plume model and other supporting information (Charnock *et al.*, 1998). In conclusion the paper describes a method of reconciling predictions and measurements which offers the potential to assist greatly the understanding of the situation following an accident.

Jack P. C. Kleijnen (*Tilburg University*)

I am a *non-Bayesian* analyst of *stochastic discrete event simulations*. Such simulations represent 'residual variability' (see Sections 2.1.3 and 4.6) through *(pseudo)random numbers*. Examples are queuing simulations in logistics, which generate customers' arrival times through a Poisson distribution with a fixed unknown parameter θ, so waiting times become random outputs.

In such simulations, *calibration* is considered bad practice! For example, observed arrival times should be used to fit an input distribution. Also see the case-study for the Dutch navy in Kleijnen (2000).

I agree that—after estimating the input parameters—uncertainty remains. But, this uncertainty may be modelled through *Monte Carlo* sampling, using a fitted input distribution; see Section 2.3. Alternatively, *bootstrapping* may be used; see Kleijnen (2000). Neither approach assumes *normality* for the input or output distributions!

Code uncertainty (Section 2.1.6) may be caused by programming *bugs*; see verification—not validation—in Kleijnen (2000).

Instead of assuming a *prior* distribution on the linear regression parameters β, we may first estimate β, and then check whether these estimates have signs that agree with experts' prior knowledge. For example, in queuing simulations the estimated main effect of traffic rate θ on waiting time should be positive; otherwise the model violates validation or verification. In a (deterministic) 'global greenhouse' simulation we detected that two computer modules were called in the wrong order.

I think that the *multivariate* character of the code output (Section 4.1) is relevant: their correlations should be incorporated through generalized least squares. These correlations can be estimated from replicated runs.

Besides *experimental design* for deterministic simulation (Section 5.1), there are designs (including screening) for stochastic simulation; see Kleijnen (1998).

Observed input data in historical order can validate the total model, so-called *trace-driven* simulation; see Kleijnen *et al.* (2000).

We should avoid *spurious regressors* (Section 5.2), in parametric regression also: though they improve the fit, they also degrade the predictor. Moreover, besides *prediction* regression serves (parsimonious) *explanation*!

Bayesians and frequentists—and deterministic and stochastic simulators—should learn from each other: the saying 'East is East, . . .' should not apply to the computer simulation area! Fortunately, the authors have succeeded in writing a paper that challenges researchers with diverse backgrounds.

William Notz, Thomas Santner, Brian Williams and Jeffrey Lehman (*Ohio State University, Columbus*)
We would like to thank the authors for a stimulating contribution on an important topic. We appreciate their detailed decomposition of the uncertainty in the calibration process into code, model, parameter and other components. We make the following comments.

In the prosthesis design applications that we have investigated, the calibration inputs vary rather than being fixed. The output is a function of the stress at the bone–prosthesis boundary and depends on the bone elasticities, which vary across the population of patients for whom the prosthesis is to be used. In this case, the calibration involves the parameters that govern these distributions for the population of patients of interest.

Another situation that could be modelled by this approach is one involving a 'real' experiment, a surrogate experiment and the computer experiment. The surrogate only approximates the real experiment and is performed because the real experiment is very expensive. The surrogate is also relatively expensive, so one additionally runs the computer experiment as a cheaper, but cruder, approximation to the real experiment. Models for the real and surrogate experiments will include random deviations from a true response surface. Models for surrogate and computer experiments will include model misspecification components because they are approximations. We want the posterior of the true surface given the surrogate and computer data.

The discussion of 'true parameter values' in Section 4.3 is thought provoking. The choice of the metric that determines the model fit is left to the user, so best fitting only has meaning for a given example and metric. Also, the true parameter value changes as the data increase because it is best in the sense of representing $z_1, \ldots, z_n$. This suggests that the calibration parameters are an artefact of the code and the purpose for which the code was written, rather than quantities having physical meaning.

Another approach to design is the sequential strategy introduced by Schonlau (1997). Williams *et al.* (2000) developed a similar strategy for interpolation in models similar to those discussed by Kennedy and O'Hagan. Modifications for calibration would seem possible.

Andrea Saltelli (*European Commission, Ispra*)
My impression of the work of Kennedy and O'Hagan is very positive, and I would like to compliment them for the excellent quality of their work.

The authors state that 'Crudely put, calibration is the activity of adjusting the unknown rate parameters until the output of the model fits the observed data'. This discounts a vast body of literature based on solving problems of determining fundamental quantities (e.g. in quantum mechanics) using first-principle calculations that solve inverse problems. See Turanyi's review in Saltelli *et al.* (2000a). The authors are correct in noting that these approaches are non-statistical, being mostly based on derivative analysis. Derivatives analysis is also valuable for estimating the output of expensive models at low computational cost (see also Griewank (2000)).

My second remark is about the authors' evaluation of the generalized likelihood uncertainty estimate (GLUE) approach, when they say that it falls short of appreciating model inadequacy. When one uses the GLUE approach, not only can a large dimensionality $\boldsymbol{\theta}$ be explored, but one can also sample from alternative models and model structures. If this is coupled with a quantitative sensitivity analysis, then the total variation in the prediction driven by model variation can be recovered. I would claim that this approach is a valid strategy to appreciate how much prediction uncertainty results from the poor knowledge of the underlying mechanism. Examples are given in Draper *et al.* (2000), Tarantola *et al.* (2000) and Planas and Depoutot (2000) in Saltelli *et al.* (2000a). See also Saltelli *et al.* (2000b).

I would also appreciate it if the authors could express their views on the related topic of equifinality. This term indicates that (as the authors note) all models are wrong, and that several plausible models are in general compatible with the same set of data. This topic is well known to advocates of the GLUE approach. In the example provided by Kennedy and O'Hagan, the plume model is assumed given, i.e. already selected by a peer community. In regions of the space of the input where the model is inadequate, the available data 'take over', and the strategy suggested provides interpolation. I nevertheless wonder whether this approach, based on a single realization from the universe of models, really captures all the uncertainty that might derive from a poor model specification. If the model were to be

used for predictive purposes, or for extrapolation rather than interpolation, then I would consider approaches such as the GLUE approach with more confidence.

Neil Shephard (*Nuffield College, Oxford*)
The use of computationally involved modelling in economics (e.g. in real business cycle research and in financial economics) has motivated the development of so-called 'indirect inference' methods in econometrics. These allow us to estimate models from empirical data when we can only simulate from the model (rather than evaluate the likelihood function). The classic reference to this work is Gourieroux *et al.* (1993). Although this method has the drawback of not being Bayesian, it does have the virtue of avoiding the introduction of the infinite dimensional unknown functional forms that are used in this paper.

Håkon Tjelmeland (*Norwegian University of Science and Technology, Trondheim*)
I would like to compliment the authors on a very interesting and stimulating paper. My interest in the topic is for modelling petroleum reservoirs. The calibration inputs then include spatially varying variables like porosity and permeability. To represent a heterogeneous reservoir, these variables must be defined on a grid, making calibration inputs high dimensional. The available observations are from a (typically) low number of wells, where pressure and production rates are observed as time series. Thus, observations are multivariate. The code, or simulator, predicts pressures and rates for given porosity and permeability values. The simuator is run on a lattice, which may or may not coincide with the grid used for porosity and permeability, and it is believed that the predictions of the simulator improve with the grid resolution. However, the simulation time also increases with larger grids. One run of the simulator takes from a few minutes to hours or even days, dependent on the size of the grid.

I find the Bayesian model formulation in the paper appealing, but I do not understand how the procedure can be implemented for the situation sketched above. I do not see how a realistic and informative prior for f can be specified. With high dimensional input and output variables, the situation is just too complex. Moreover, with high dimensional calibration inputs and only a token prior, I would not trust results based on a very limited number of runs with the simulator. Therefore, I find it more natural to use the simulator for a low resolution grid, to consider the simulator as a known function and to include it as a part of the observation likelihood. This results in a more standard Bayesian model, but, as one run of the simulator still takes minutes to run, the posterior cannot be explored by standard random walk or independent proposal Metropolis–Hastings algorithms. In some of my work I have therefore focused on how to construct better proposal mechanisms in the Metropolis–Hastings algorithm, by including optimizations and specifically consider the possibility of multimodality of the posterior (Tjelmeland and Hegstad, 2001; Tjelmeland, 2000). For the former, it is natural to use the possibility of the simulator to supply gradients in addition to mere predictions.

An intermediate modelling strategy would be to consider the simulator as a known function when run on a coarse grid and unknown for a fine grid. The known coarse grid simulator could then be used to define an informative prior for the unknown fine grid simulator.

The **authors** replied later, in writing, as follows.

We thank all the discussants for their thoughtful and stimulating comments. We have tried to address them via generic themes, and we apologize if this has meant that we have failed to respond to some relevant questions.

Meaning of parameters
Wynn, Craig and Cooke all question in some way whether the parameters of our model are properly identified. Our basic model is equation (5), and for the moment we shall assume that the code $\eta(\cdot, \cdot)$ is known. The way that we model the inadequacy function $\delta(\cdot)$ is as in equation (2), where there is a mean function with unknown hyperparameters β and a zero-mean process. We could rewrite model (5) as

$$z_i = g(\mathbf{x}_i, \lambda) + \epsilon(\mathbf{x}_i), \tag{13}$$

where $g(\mathbf{x}_i, \lambda) = \mathbf{h}(\mathbf{x}_i)^{\mathrm{T}}\beta + \rho\,\eta(\mathbf{x}_i, \theta)$, so that λ includes θ, β and ρ, and $\epsilon(\mathbf{x}_i) = e(\mathbf{x}_i) + e_i$ is an error term which is correlated in the space of the variable parameters $\mathbf{x}$ but includes a 'nugget term' for the variance of e_i. It seems to us that these two components are intrinsically well defined as in any regression problem. Fitting will mean finding a systematic term $g(\cdot, \lambda)$, indexed by parameters λ, to fit the data best according to the error covariance structure (some hyperparameters of which are also to be estimated).

We do not accept Craig's suggestion that the identification of the model is limited by an upper bound on the degrees of freedom, since we can in principle obtain an indefinite quantity of observational data in exactly the same way as for any regression problem. We do not believe that there is any fundamental unidentifiability in our formulation.

There may of course be practical problems associated with how the class of systematic functions $g(\cdot, \lambda)$ is indexed, resulting in many different λ giving good fits to the data; and as Beven points out these good fitting sets may be widely separated. Care will be needed in integrating over $\boldsymbol{\theta}$, since it is here that poor identification (called 'equifinality' by Saltelli) will be focused. It is for this reason that we are wary of Goldstein's suggestion to handle $\boldsymbol{\theta}$ only implicitly. His Bayes linear methods will, we suspect, work best if $\boldsymbol{\theta}$ is implicitly well identified so that its posterior distribution is approximately normal. In the presence of equifinality the Bayes linear approach may oversimplify.

Several questions relating to our suggestion that the value of $\boldsymbol{\theta}$ as defined by us as a best fitting calibration may differ from the modeller's intended physical meaning were raised in the discussion. Craig asks how a prior distribution for $\boldsymbol{\theta}$ could be elicited under our definition, and whether in our examples the distinction was explained to the National Radiological Protection Board. If substantial prior information were available about a parameter, so that an informative prior distribution could be elicited, then we believe that this will typically relate to its intended physical meaning. However, it is important to recognize that often the supposed physical meanings of parameters are also not very well defined. In terms of the notation proposed by Campbell and McKay, the computer model is very often formulated by using parameters $\mathbf{T}(\boldsymbol{\theta})$ that are simplifications of the true physical parameters. An example is the deposition velocity in the atmospheric dispersion model. The deposition velocity is potentially well defined and measurable at a single location but will vary in space because of varying topography and ground cover. The model uses a single value that is obviously a simplification, and intended as a kind of average value. As such, and given all the other simplifications in the model, it is already clear that the deposition velocity in the model is most usefully thought of in terms of a best fitting value. So, to answer one of Wynn's questions, we suspect that there is no real difference between a 'tuning' parameter in a model and a physical parameter. It is in our opinion doubtful whether strongly informative prior distributions for calibration parameters are ever meaningful in practice.

On a related question, Brown, Beven, Challenor and Kleijnen refer to extrapolation of models. The physical meanings of parameters are usually intended to hold in a range of contexts, and may even be thought of as universal. By obtaining observational data and calibrating the model, we hope to learn about these parameters for scientific understanding or for extrapolation. If as in our approach, the calibrated values need not bear any relation to the intended physical values, these purposes are not well served. As just explained, however, we suspect that this problem is intrinsic to the use of models, and not simply a feature of our approach. If all models are wrong, what does it mean to learn the true value of a parameter in a model: what scientific understanding can be gained, ultimately? In practice, learning about the apparent values of parameters as revealed in imperfect models *does* aid understanding and helps to build better models, whereby the meanings and estimates of the parameters themselves are steadily refined.

In proposing to estimate a model inadequacy function, and to use this to improve predictions from the model, we do not intend this to be the end in itself. It is surely true that by studying the estimated model inadequacy terms the modellers may gain insight into how to improve the model structure (or even to find bugs in the code, as suggested by Kleijnen!). In particular, both Challenor and Cooke suggest that, if the fitted value of a physical parameter differs widely from values believed to be realistic, then this is another indication of model inadequacy. Notice that this supposes that there is strong prior information about the physical parameter, but if we use this prior information in the analysis it is unlikely that the posterior estimate would be so different from the prior. Another reason for using deliberately weakened prior information about the calibration parameters (as mentioned in Section 4.3) might be to detect this kind of problem with the model.

Covariance structure

Several discussants offer comments on the modelling of covariance structures. Diggle suggests using a powered exponential family, whereas we generally fix the power at 2, and also the Matérn family. Although Diggle feels that differentiability of the function should guide the choice of covariance model, we find it difficult to imagine that the kinds of functions that we deal with will act more like realizations from one of Diggle's suggested models than from another. Our choice makes realizations analytic, and we acknowledge that in reality the functions may not be differentiable, or even continuous, everywhere.

However, we do not believe in using a model that says that the function will be differentiable *nowhere*, or would have say two derivatives everywhere but third derivatives nowhere.

The real problem, as alluded to by Brown and by Diggle in his final comment, is non-stationarity. The cause of the heavy tails in a Q–Q plot like Fig. 2 is local irregularities in the function, that occur where we have no observations and that are not being allowed for by our covariance function. To some extent these issues are related to Rougier's point about there being little information sometimes about short scale variation (although he was referring more to the function $\eta(\cdot, \cdot)$ than to $\delta(\cdot)$). Nevertheless, we would like to be able to model the possibility of localized features. It is not enough simply to fit locally, as Diggle mentions, because this does not allow prediction into gaps where the local structure may have changed again.

The choice of covariance function effectively determines the smoother, as Wynn points out, and is equivalent to the choice of a class of basis functions as mentioned by Brown. Provided that a reasonably rich family is used, almost all such methods would provide essentially equivalent interpolation (although this may not be entirely true of very local methods such as wavelets). They may, however, differ substantially in the variances that are asserted for those interpolations. This will happen within our class of Gaussian kernels, where a range of values for the roughness parameters may all give very similar interpolations but quite different posterior variances. This is why we feel that the predictive validation used in our examples is important. The fact that the slope of the line at the right-hand side of Fig. 2 is very close to 1 over the great bulk of predictions is the key finding that supports our predictive variances, whereas the fact that it is clearly not 1 on the left-hand side is a criticism of the predictive variance obtained with simple Gaussian process interpolation.

The latter corresponds to simple kriging, but the geostatistical methods referred to by Higgins and Jones are rather more sophisticated. In fact, expression (13) shows an error term analogous to conventional kriging, but the function $g(\cdot, \cdot)$ implies building the model $\eta(\cdot, \cdot)$ into a non-linear form of universal kriging. It would be interesting to see whether our approach would be useful in more conventional geostatistical applications.

More complex features of applications

Another point made by several discussants is that there are features in many kinds of computer codes that would defeat our methods in one way or another. The scale of an application is one factor that has several aspects. Although we allude to extending our results to multivariate outputs, we are aware that much needs to be done. The generalization mentioned by Brown is natural but entails assuming a separability in the covariance structure. Kleijnen's suggestion regarding multivariate outputs may be useful for stochastic computer codes but only addresses that part of covariances across different outputs that is induced by the replication error. As Anderson points out, the outputs may be spatial or spatio-temporal, and this has implications for covariance modelling.

The dimensionality of the calibration parameter $\boldsymbol{\theta}$ is an important limitation for our methods. The simulation approach of the generalized likelihood uncertainty estimate (GLUE) is better suited to high dimensional input spaces, in which the problem of equifinality is more endemic. Both Notz (and colleagues) and Tjelmeland point out that in many situations we need to calibrate on a parameter that varies over a field. The suggestion of Notz to model these with a Gaussian random field (GRF) and to calibrate on the parameters of that field is nice. It can be seen as providing a GRF prior distribution for $\boldsymbol{\theta}$ with extra hyperparameters to be estimated, but in effect $\boldsymbol{\theta}$ is only estimated implicitly, having been integrated out of the analysis in a way that parallels Goldstein's approach.

Although high dimensionality of $\boldsymbol{\theta}$ leads to computational infeasibility for our approach, in practice there are usually only a few inputs in a model that are critical. In the field of sensitivity analysis (Saltelli *et al.*, 2000) we usually find that the model is sensitive to relatively few parameters, and in calibration there may correspondingly be substantive information on which to calibrate only those inputs to which the output is sensitive. The existence of inputs to which the output is not sensitive (over the range of the calibration data) is one reason for what GLUE users call equifinality. We believe that identifying the important input parameters may be a key first step towards effective calibration. In the structure suggested by Notz, this role is played by the hyperparameters of the GRF.

A third way in which the scale of an application may make our methods impractical to apply relates to the number of observations. As Brown says, we require to invert a matrix of dimension $N + n$, which is an order $(N + n)^3$ computation. We believe that local fitting, as mentioned by Diggle, offers a way forward when the quantity of data is very large, but it will be more complex to apply in our context than in simple interpolation.

Anderson points out that large models are often made by linking together smaller models. There is clearly scope for recognizing this structure in the analysis, perhaps by the kind of partial opening of the 'black box' advocated by Anderson. Kendall's comment is related, being concerned with trying to validate components within a model. Another related point is that of Tjelmeland, who indicates how simpler, quicker versions of a model might be used to provide information about the model of interest. His suggestion of using the quick simulator to build prior information seems to be the same idea as that used by Craig *et al*. (1996). Kennedy and O'Hagan (2000) offer perhaps a more ambitious use of such information. The surrogate experiment described by Notz is a similar idea, but the surrogate is intended to be closer to reality than the code.

In general, we agree with Wynn that appreciating what goes on inside the black box may be a key to more realistic modelling. However, it will be important to appreciate the relationships between components, and not just those induced by the model structure itself. Model builders invariably apply informal testing whose effect is that the overall model inadequacy and uncertainty may be much less than a naïve assessment of the adequacy of components might predict.

Data assimilation is mentioned by Anderson, which can be seen as a kind of on-line calibration as new data become available. The timescale over which data arrive (and the quantity of data) may make it quite impractical to apply our methods in this way. Some simplification along the lines of Wynn's suggested use of Kalman filtering will be needed.

Wynn mentions that derivatives of $\eta(\cdot, \cdot)$ (although not of $\zeta(\cdot)$!) are often available. Within our Gaussian process modelling, these are straightforward to incorporate (see a brief mention in O'Hagan (1992)).

Other comments

Saltelli links the calibration problem with more general inverse problems. Although this is a useful extra point of view, the nature of inverse problems is that different methods are needed for different problems. He also makes the useful point that uncertainty about model structure can easily be incorporated by creating new calibration parameters to index model variations (in our method just as easily as in GLUE).

Shephard suggests that our use of 'infinite dimensional unknown functional forms' might be avoided by devices such as the indirect inference method used in econometrics. We have two such functions. The model inadequacy function seems to us to be necessary but not accounted for in his suggestion. We treat the model code as an unknown function in situations where to evaluate it is costly, so that running the code thousands of times is impractical, and this also will still be needed if indirect sampling from the model is costly.

Cooke asks about the example in Section 6.2. Strategy 1 did use different roughness parameters downwind and crosswind. The increase in root-mean-squared error at $n = 20$ for strategy 1 is due to features of the five extra points which altered the fitted roughness parameters substantially. The calibration of strategy 3 used squared errors on the log-scale, so that all n observations contributed to the fit.

References in the discussion

Beven, K. J., Freer, J., Hankin, B. and Schulz, K. (2000) The use of generalised likelihood measures for uncertainty estimation in high order models of environmental systems. In *Nonlinear and Nonstationary Signal Processing* (eds W. J. Fitzgerald, R. L. Smith, A. T. Walden and P. C. Young), pp. 115–151. Cambridge: Cambridge University Press.

Brown, P. J. (1993) *Measurement, Regression, and Calibration*. Oxford: Clarendon.

Brown, P. J., Le, N. D. and Zidek, J. V. (1994a) Multivariate spatial interpolation and exposure to air pollutants. *Can. J. Statist.*, **22**, 489–509.

———(1994b) Inference for a covariance matrix. In *Aspects of Uncertainty: a Tribute to D. V. Lindley* (eds P. R. Freeman and A. F. M. Smith), pp. 77–92. Chichester: Wiley.

Cai, Y. and Kendall, W. S. (2001) Perfect implementation of simulation for conditioned Boolean model via correlated Poisson random variables. *Statist. Comput.*, to be published.

Charnock, T. W., Daniels, W. M. and Higgins, N. A. (1999) Geostatistical estimation techniques applied to radionuclide deposition: an accident response decision aid. In *Geostatistics for Environmental Applications* (eds J. Gómez-Hernández, A. Soares and R. Froidevaux). Dordrecht: Kluwer.

Craig, P. S., Goldstein, M., Rougier, J. C. and Seheult, A. H. (2001) Bayesian forecasting using large computer models. *J. Am. Statist. Ass.*, **96**, in the press.

Craig, P. S., Goldstein, M., Seheult, A. H. and Smith, J. A. (1996) Bayes linear strategies for matching hydrocarbon reservoir history. In *Bayesian Statistics 5* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 69–95. Oxford: Oxford University Press.

Diggle, P. J. and Ribeiro, P. J. (2001) Bayesian inference in Gaussian model-based geostatistics. *Geogr. Environ. Modllng*, to be published.

Diggle, P. J., Tawn, J. A. and Moyeed, R. A. (1998) Model-based geostatistics (with discussion). *Appl. Statist.*, **47**, 299–350.

Draper, D., Saltelli, A., Tarantola, S. and Prado, P. (2000) Scenario and parametric sensitivity and uncertainty analyses in nuclear waste disposal risk assessment: the case of GESAMAC. In *Sensitivity Analysis* (eds A. Saltelli, K. Chan and E. M. Scott). New York: Wiley.

Gourieroux, C., Monfort, A. and Renault, E. (1993) Indirect inference. *J. Appl. Econometr.*, **8**, S85–S118.

Grievank, A. (2000) *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Philadelphia: Society for Industrial and Applied Mathematics.

Kendall, W. S. (1998) Perfect simulation for the area-interaction point process. In *Probability Towards 2000* (eds L. Accardi and C. C. Heyde), pp. 218–234. New York: Springer.

Kendall, W. S. and Møller, J. (1999) Perfect implementation of a Metropolis-Hastings simulation of Markov point processes. *Research Report 348*. Department of Statistics, University of Warwick, Coventry. (Available from `http://www.warwick.ac.uk/statsdept/staff/WSK/papers/348.ps.gz`.)

———(2000) Perfect simulation using dominating processes on ordered state spaces, with application to locally stable point processes. *Adv. Appl. Probab.*, **32**, 844–865.

Kennedy, M. C. and O'Hagan, A. (2000) Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, **87**, 1–13.

Kibria, B. M. G., Sun, L., Zidek, J. V. and Le, N. D. (2000) A Bayesian approach to backcasting and spatially predicting unmeasured multivariate random space-time fields with application to $PM_{2.5}$. *Technical Report 193*. Department of Statistics, University of British Columbia, Vancouver.

Kleijnen, J. P. C. (2000) Strategic directions in verification, validation, and accreditation research: a personal view. In *Proc. 2000 Winter Simulation Conf.*, pp. 910–911. Baltimore: Association for Computing Machinery.

———(1998) Experimental design for sensitivity analysis, optimization, and validation of simulation models. In *Handbook of Simulation* (ed. J. Banks), pp. 173–223. New York: Wiley.

Neal, R. (1999) Regression and classification using Gaussian process priors (with discussion). In *Bayesian Statistics 6* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 475–501. Oxford: Oxford University Press.

O'Hagan, A. (1992) Some Bayesian numerical analysis (with discussion). In *Bayesian Statistics 4* (eds J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith), pp. 345–363. Oxford: Oxford University Press.

Planas, C. and Depoutot, R. (2000) Sensitivity analysis for signal extraction in economic time series. In *Sensitivity Analysis* (eds A. Saltelli, K. Chan and E. M. Scott). New York: Wiley.

Propp, J. G. and Wilson, D. B. (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics. *Rand. Struct. Algs*, **9**, 223–252.

Romanowicz, R., Beven, K. J. and Tawn, J. (1994) Evaluation of predictive uncertainty in non-linear hydrological models using a Bayesian approach. In *Statistics for the Environment II: Water Related Issues* (eds V. Barnett and K. F. Turkman), pp. 297–317. Chichester: Wiley.

Saltelli, A., Chan, K. and Scott, E. M. (eds) (2000a) *Sensitivity Analysis*. New York: Wiley.

Saltelli, A., Tarantola, S. and Campolongo, F. (2000b) Sensitivity analysis as an ingredient of modelling. *Statist. Sci.*, **15**, 377–395.

Schonlau, M. (1997) Computer experiments and global optimization. *PhD Thesis*. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo.

Shershakov, V. M., Vakulovski, S. M., Borodin, R. V., Vozzhennikov, O. I., Gaziev, Y. L., Kosykh, V. S., Makhonto, V. S., Chumiciev, V. B., Korsakov, A. T., Martynenko, V. P. and Godko, A. (1995) Analysis and prognosis of radiation exposure following the accident at the Siberian chemical combine Tomsk-7. *Radian Protectn Dosim.*, **59**, 93–126.

Tarantola, S., Jesinghaus, J. and Puolamaa, M. (2000) Global sensitivity analysis: a quality assurance tool in environmental policy modelling. In *Sensitivity Analysis* (eds A. Saltelli, K. Chan and E. M. Scott). New York: Wiley.

Tjelmeland, H. (2000) An MCMC algorithm for sampling from the posterior conditioned to production history. In *Geostatistics, Cape Town*.

Tjelmeland, H. and Hegstad, B. K. (2001) Mode jumping proposals in MCMC. *Scand. J. Statist.*, **28**, 205–223.

Williams, B. J., Santner, T. J. and Notz, W. I. (2000) Sequential design of computer experiments to minimize integrated response functions. *Statist. Sin.*, **10**, 1133–1152.