# Modelling with Deterministic Computer Models

Jeremy Oakley

December 10, 2009

## 1 Introduction

In this chapter we consider various statistical issues in the use of deterministic computer models. By deterministic (or 'mechanistic') computer model, we mean a mathematical representation of a physical process that has been constructed from the modeller's understanding of the science underlying the process, and is implemented on a computer. To distinguish between a computer model and other types of model (such as statistical model), we refer to a computer model as a *simulator*. We represent the simulator by a function $\mathbf{y} = \eta(\mathbf{x})$, where $\mathbf{y}$ is a vector of simulator outputs, and $\mathbf{x}$ is a vector of simulator inputs. The process of running a simulator at different input settings is known as a *computer experiment*.

The simulator is deterministic in that repeated evaluations of $\eta(\mathbf{x})$ at the same input $\mathbf{x}$ will result in the same output value. The simulator is often sufficiently complex such that a closed form expression for $\eta$ is not known. For example, $\mathbf{y}$ may be the solution to a set of differential equations, with $\mathbf{x}$ representing boundary conditions and other parameters within the system.

Simulators are used in a wide variety of scientific fields, typically when physical observations of the outputs of interest are impossible or impractical to obtain. In climate forecasting, simulators are used to predict future climate for different $CO_2$ emission scenarios. Health economic decision models are used to predict cost-effectiveness of medical treatments, often over time periods far in excess of those observed within clinical trials of those treatments. Mathematical models are used in the design of engineering structures, where building large numbers of different physical prototypes is too costly. Despite the deterministic nature of these simulators, there are various statistical problems associated with their use. Some of the main examples are as follows.

1. *Uncertainty analysis*

   We suppose that there is a vector of 'true' inputs $\mathbf{X}$ corresponding to some

particular situation of interest, but that there is uncertainty about the value of $\mathbf{X}$. If we are uncertain about $\mathbf{X}$, what is our uncertainty about $\mathbf{Y} = \eta(\mathbf{X})$?

2. *Sensitivity analysis*
   Sensitivity analysis is concerned with investigating how simulator outputs respond to changes in simulator inputs. In an extension of the uncertainty analysis problem, we may wish to investigate how individual elements of $\mathbf{X}$ contribute to the uncertainty in $\mathbf{Y}$.

3. *Simulator calibration*
   Given (possibly noisy) physical observations of outputs predicted by a simulator (or functions of those outputs) can we infer the values of the associated simulator inputs?

4. *Simulator discrepancy*
   The simulator will not be a perfect description of reality. What can we say about the discrepancy between the simulator output, and the true values of the physical process?

5. *Computationally expensive simulators*
   Evaluating $\eta(\mathbf{x})$ for a single value of $\mathbf{x}$ (performing one simulator 'run') may take a considerable amount of computing time (perhaps many hours or days). How should we tackle the above problems in this case?

# 2 Metamodels and emulators for computationally expensive simulators

We now review a well-established approach for dealing with computationally expensive simulators. We specify the problem as follows: due to the computational expense of the simulator, we are only able to run the simulator relatively small number of times to obtain $D = \{\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)\}$, but we wish to know $\eta(\mathbf{x})$ at many more values of $\mathbf{x}$. Hence we wish to make joint inferences about some (possibly infinite) set of outputs $\mathbf{y}_{n+1} = \eta(\mathbf{x}_{n+1}), \mathbf{y}_{n+2} = \eta(\mathbf{x}_{n+2}), \ldots$ given the available simulator runs $D$.

The idea is to construct a statistical model, known as a *metamodel*, for $\eta(\cdot)$ based on the available runs, and then use the statistical model as a replacement for $\eta(\cdot)$ in any subsequent analysis of the simulator. This is a statistical regression problem, with the important feature that the data are noise free: we cannot observe

two different values of $\mathbf{y}$ at the same $\mathbf{x}$. Any regression technique can be used to construct a metamodel, with one of the most popular being nonparametric regression using Gaussian processes. Alternatives metamodelling approaches include the use of neural networks (see for example El Tabach et al., 2007) and methods based on the use of "high dimensional model representation", which we discuss further in section 4.

We use the term *emulator* to mean a full probabilistic specification for $\eta(\cdot)$, so that an emulator is a metamodel that provides both an estimate of $\eta(\cdot)$ and quantifies uncertainty about $\eta(\cdot)$ due to only evaluating $\eta(\mathbf{x})$ at a limited number of values of $\mathbf{x}$. In constructing an emulator, we think of $\eta(\cdot)$ as an unknown function. For any computationally expensive simulator, we are unlikely to have a closed form expression for $\eta(\cdot)$, and the value of $\eta(\mathbf{x})$ for any $\mathbf{x}$ will be unknown to us prior to evaluating $\eta(\mathbf{x})$. Hence the process of building an emulator can be thought of naturally within the framework of Bayesian inference, in which we consider prior uncertainty about $\eta(\cdot)$, and update our beliefs as simulator runs become available.

## 2.1 Gaussian processes emulators

Early development of Gaussian process curve fitting and regression can be found in Kimeldorf and Wahba (1970), Blight and Ott (1975) and O'Hagan (1978). More recently, Gaussian processes have become popular in the machine learning community for regression and classification (Neal, 1999; Rasmussen and Williams, 2006). Interpolation with Gaussian processes can be viewed as equivalent to the method of Kriging in spatial statistics (see for example Cressie, 1993).

Within the field of computer experiments, the first use of Gaussian process regression for modelling computer code output was Sacks et al. (1989), with a Bayesian treatment given in Currin et al. (1991). A detailed account can be found in Santner et al. (2003) and online at `http://mucm.aston.ac.uk/MUCM/MUCMToolkit`. Recent developments include diagnostic tools for validating emulators (Bastos and O'Hagan, 2009) and "treed" Gaussian processes for dealing with nonstationarity (Gramacy and Lee, 2008).

We first make the simplification that the output of the simulator is a scalar, in common with much of the literature, before considering extensions to multivariate outputs. In the Gaussian process emulator, for any collection of input values $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, we describe our uncertainty about the corresponding set of outputs $\{\eta(\mathbf{x}_1), \ldots, \eta(\mathbf{x}_n)\}$ with a multivariate normal distribution. The Gaussian process is characterised by the mean of $\eta(\mathbf{x})$ and the covariance function for $\eta(\mathbf{x})$ and $\eta(\mathbf{x}')$.

The mean of $\eta(\mathbf{x})$ is specified as a parametric function

$$E\{\eta(\mathbf{x})|\boldsymbol{\beta}\} = \mathbf{h}(\mathbf{x})^T\boldsymbol{\beta}.$$

The vector $\mathbf{h}(\cdot)$ consists of $q$ known regression functions of $\mathbf{x}$, and $\boldsymbol{\beta}$ is a vector of unknown coefficients. We choose $\mathbf{h}(.)$ to incorporate any beliefs we might have about the form of $\eta(.)$. For example, if we expect there to be an approximate linear trend in $y$ as an input $x_i$ increases then we might include a linear term $x_i$ in $\mathbf{h}(.)$.

The covariance between $\eta(\mathbf{x})$ and $\eta(\mathbf{x}')$ is specified as

$$Cov\{\eta(\mathbf{x}), \eta(\mathbf{x}')|\sigma^2, \boldsymbol{\phi}\} = \sigma^2 c(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi}),$$

where $c(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi})$ is a function of $\mathbf{x}, \mathbf{x}'$ and parameters $\boldsymbol{\phi}$ which decreases as $|\mathbf{x} - \mathbf{x}'|$ increases, and also satisfies $c(\mathbf{x}, \mathbf{x}; \boldsymbol{\phi}) = 1$ for all $\mathbf{x}$. Note that the prior variance of $\eta(\mathbf{x})$ (conditional on $\sigma^2, \boldsymbol{\beta}, \boldsymbol{\phi}$) is $\sigma^2$ for all $\mathbf{x}$, and so the choice of $\mathbf{h}(.)$ should reflect a judgement that (squared) differences between $\eta(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})^T\boldsymbol{\beta}$ are not expected to be larger at some $\mathbf{x}$ than others, a priori.

The covariance function must ensure that the covariance matrix of any set of outputs $\{y_1 = \eta(\mathbf{x}_1), \ldots, y_n = \eta(\mathbf{x}_n)\}$ is positive semi definite. A common choice is the Gaussian covariance function:

$$c(\mathbf{x}, \mathbf{x}'; \boldsymbol{\phi}) = \exp\left\{-\sum_{i=1}^{d}\left(\frac{x_i - x_i'}{\phi_i}\right)^2\right\}, \tag{1}$$

with $\phi_i$ known as the correlation length parameter for input dimension $i$. Note that some authors use different names and parameterisations for $\phi_i$. Rasmussen and Williams (2006) use the term characteristic length scale for $\phi_i$, and Kennedy and O'Hagan (2001) write $b_i = 1/\phi_i^2$ with $b_i$ known as the roughness parameter.

Conventionally, a weak prior for $\boldsymbol{\beta}$ and $\sigma^2$ in the form $p(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$ is used. In Oakley (2002) a means of including proper prior information about the function $\eta(.)$ is presented, through the use of the conjugate prior, the normal inverse gamma distribution. We have

$$p(\boldsymbol{\beta}, \sigma^2) \propto (\sigma^2)^{-\frac{1}{2}(v+q+2)} \exp[-\{(\boldsymbol{\beta} - \mathbf{z})^T V^{-1}(\boldsymbol{\beta} - \mathbf{z}) + a\}/(2\sigma^2)]. \tag{2}$$

### 2.1.1 Choice of training data

We must now choose training data inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ at which to run the simulator. Choosing suitable inputs is an experimental design problem, but as argued in Sacks et al. (1989), in the absence of random error in any single code run, methodology for designing physical experiments may not be ideal for designing computer experiments.

For example, the notions of replication and randomly allocating 'treatments' to experimental units are not relevant here.

A common strategy is to choose a space-filling design, in an attempt to ensure that there will be design points close to $\mathbf{x}$ for any $\mathbf{x}$ at which we wish to predict $\eta(\mathbf{x})$. A popular choice is the maximin Latin hypercube approach suggested in Mitchell and Morris (1995), which involves maximising the minimum distance between any two design points within a Latin hypercube design.

Criterion based designs have also been proposed, for example, Currin et al. (1991) use a design based on maximum entropy, though this requires prior knowledge of the correlation length parameters. A third strategy is to choose design points sequentially, targeting successive design points in regions where uncertainty is greatest, or the emulator is believed to be predicting poorly. A general adaptive scheme is presented in Busby (2009), and Oakley (2004) uses a sequential design for estimating extreme percentiles in the uncertainty analysis problem.

### 2.1.2 The posterior Gaussian process

The output of $\eta(.)$ is observed at the training inputs, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to obtain data $D$.

Given the prior in (2) it can be shown that

$$\frac{\eta(\mathbf{x}) - m^*(\mathbf{x})}{\widehat{\sigma}\sqrt{c^*(\mathbf{x},\mathbf{x})}} | D, \boldsymbol{\phi} \sim t_{v+n}, \tag{3}$$

where

$$\begin{aligned}
m^*(\mathbf{x}) &= \mathbf{h}(\mathbf{x})^T\hat{\boldsymbol{\beta}} + \mathbf{t}(\mathbf{x})^T A^{-1}(D - H\hat{\boldsymbol{\beta}}), && (4)\\
c^*(\mathbf{x},\mathbf{x}') &= c(\mathbf{x},\mathbf{x}') - \mathbf{t}(\mathbf{x})^T A^{-1}\mathbf{t}(\mathbf{x}') + \\
&\quad (\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T A^{-1}H)(H^T A^{-1}H)^{-1}(\mathbf{h}(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T A^{-1}H)^T. && (5)\\
\mathbf{t}(\mathbf{x})^T &= (c(\mathbf{x},\mathbf{x}_1), \ldots, c(\mathbf{x},\mathbf{x}_n)),\\
H^T &= (\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_n)),\\
A &= \begin{pmatrix} 1 & c(\mathbf{x}_1,\mathbf{x}_2) & \cdots & c(\mathbf{x}_1,\mathbf{x}_n) \\ c(\mathbf{x}_2,\mathbf{x}_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ c(\mathbf{x}_n,\mathbf{x}_1) & \cdots & & 1 \end{pmatrix},\\
\hat{\boldsymbol{\beta}} &= V^*(V^{-1}\mathbf{z} + H^T A^{-1} D),\\
\hat{\sigma}^2 &= \{a + \mathbf{z}^T V^{-1}\mathbf{z} + D^T A^{-1} D - \hat{\boldsymbol{\beta}}^T (V^*)^{-1}\hat{\boldsymbol{\beta}}\}/(n+v),\\
V^* &= (V^{-1} + H^T A^{-1} H)^{-1}\\
D^T &= (\eta(\mathbf{x}_1), \ldots, \eta(\mathbf{x}_n)).
\end{aligned}$$

It is not possible to remove analytically the conditioning on $\boldsymbol{\phi}$. The simplest option is to plug in a point estimate such as the posterior mode. Kennedy and O'Hagan (2001) found this to be adequate, in that allowing for uncertainty in $\boldsymbol{\phi}$ had little effect, though we do not always expect this to be the case. Higdon et al. (2008) and Oakley (2009) sample from the posterior distribution of $\boldsymbol{\phi}$ using Markov chain Monte Carlo, though this itself is a computationally expensive procedure due to the need to repeatedly invert the matrix $A$ for each sampled $\boldsymbol{\phi}$. A fast, approximate procedure is proposed by Nagy et al. (2007) in the case of the Gaussian correlation function, in which the posterior distribution of $\log \boldsymbol{\phi}$ is approximated by a multivariate normal distribution.

### 2.1.3 Example

We illustrate the Gaussian process emulator with a simple one-dimensional example. We suppose that the simulator is given by $\eta(x) = x + 2\cos x$. We choose $\mathbf{h}(x)^T = (1,\, x)$, the Gaussian correlation function, with $\phi$ fixed at 1, and a non-informative prior $p(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$. We choose 5 training inputs $(-2, -1, 0, 1, 2)$, obtain the 5 simulator outputs, and derive the posterior emulator. The posterior mean and pointwise 95% intervals are shown in figure 1(a). Note that there is no posterior uncertainty about $\eta(x)$ at the five training inputs.

We now add two additional training data at $x = -1.5$ and $x = 0.5$, update the emulator and show the revised posterior mean and pointwise intervals in 1(b). We now see that the posterior mean is very close to the true simulator over the range of the training data, and that posterior uncertainty is small.

In figure 1(c), we show the same emulator over a wider input range. Note that the posterior mean reverts to the form of the prior mean $\mathbf{h}(x)^T\boldsymbol{\beta}$, with $\boldsymbol{\beta}$ updated given the training data, and that posterior uncertainty is much wider outside the range of the training data. In 1(d) we show an alternative emulator based on the same data, but with the choice $\mathbf{h}(x)^T = (1)$. Within the range of the training data, the alternative choice of $\mathbf{h}(.)$ has had little effect, as the data are fairly dense and are able to adjust the posterior mean away from $\mathbf{h}(.)^T\boldsymbol{\beta}$ as appropriate. However, outside the range of the data, prediction of $\eta(.)$ is poor, with the uncertainty inappropriately represented.

## 2.2 Multivariate outputs

Multivariate outputs are typically handled through the use of *separable* covariance structures. Suppose we have $\mathbf{y} = \eta(\mathbf{x})$ with $\mathbf{y} = (y_1, \ldots, y_r)$. Conti and O'Hagan

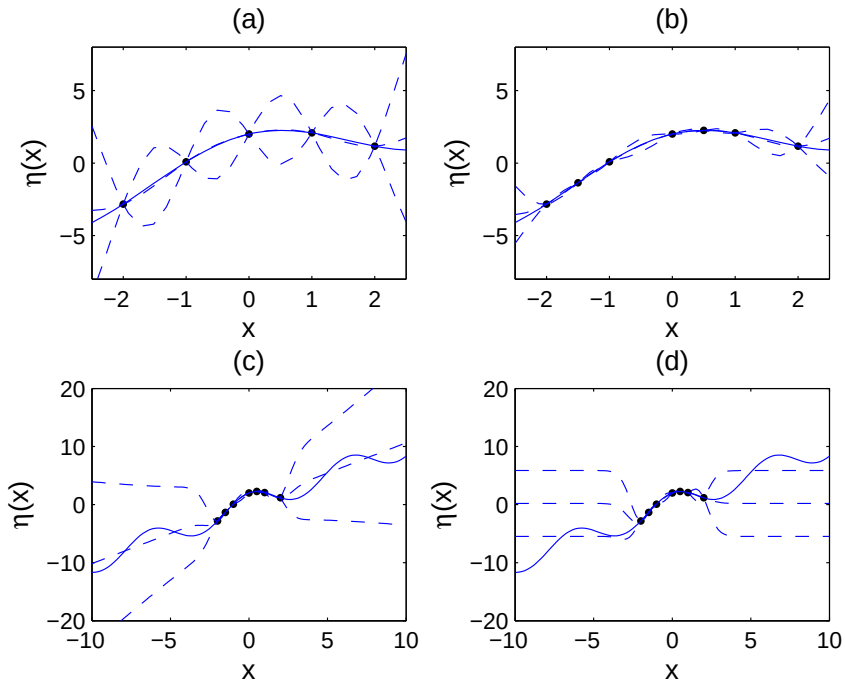


Figure 1: Solid line: true simulator, dashed lines: pointwise mean and 95% intervals, dots: training data. (a) A Gaussian process emulator with 5 training data. (b) An updated emulator with 2 additional training data. (c) Extrapolating with the emulator. (d) Extrapolating with the emulator and a constant prior mean function.

(2007) use an $r$-dimensional Gaussian process emulator with

$$\begin{aligned} E\{\eta(\mathbf{x})|B,\Sigma,\boldsymbol{\phi}\} &= B^T\mathbf{h}(\mathbf{x}), \\ Cov\{\eta(\mathbf{x}),\eta(\mathbf{x}')|B,\Sigma,\boldsymbol{\phi}\} &= c(\mathbf{x},\mathbf{x}';\boldsymbol{\phi})\Sigma, \end{aligned}$$

so that the prior variance matrix of $\eta(\mathbf{x})$ is $\Sigma$, and that the covariance between $\eta(\mathbf{x})$ and $\eta(\mathbf{x}')$ can be expressed as a product of the covariance function $c(\mathbf{x},\mathbf{x}';\boldsymbol{\phi})$ in the input space, and the variance matrix of the outputs $\Sigma$. The variance matrix $\Sigma$ can be further parameterised, if for example, the different outputs represent the same quantity predicted at different points in space or time. Rougier (2008) shows that by restricting the regressor functions in $\mathbf{h}(\cdot)$ to a particular product structure, substantial computational savings can be made, allowing the emulator to be constructed for much larger datasets.

For certain types of multivariate output, it can be beneficial to first reduce the dimension of the output before building an emulator. This can work particularly well for highly correlated outputs, for example, simulators that produce 'similar looking' time series outputs for any choice of input. Bayarri et al. (2007a) use

a wavelet representation of time series output, and build emulators for the wavelet coefficients, and Higdon et al. (2008) use principal component analysis on the output, and then build emulators for the leading principal component scores.

# 3 Uncertainty analysis

In uncertainty analysis, we suppose that there is a unknown 'true' input of interest, $\mathbf{X}$, and that we wish to quantify the uncertainty in $Y = \eta(\mathbf{X})$ induced by the uncertainty in $\mathbf{X}$. In health economic modelling, this is known as *probabilistic sensitivity analysis*, but we give a distinct definition of sensitivity analysis in section 4.

The notion of a 'true' input is contentious, particularly if there are elements within $\mathbf{X}$ that do not correspond to physical quantities, observable in the real world. Even if $\mathbf{X}$ is observable, it does not follow that quantifying uncertainty in $\eta(\mathbf{X})$ will quantify the uncertainty in the true physical output quantity represented by the simulator, as the simulator is unlikely to be a perfect representation of reality. (Modellers sometimes choose to distinguish between the true value of a physical input and the 'best' value for predictive purposes: see the example of molecular viscosity versus eddy viscosity discussed in Goldstein and Rougier, 2009). Regardless of these difficulties, uncertainty about what values of a simulator's inputs to use will contribute to uncertainty in the simulator's predictions, and it can be useful to investigate this uncertainty.

The first step is to specify a probability distribution $G$ to represent the simulator user's uncertainty about $\mathbf{X}$. This may be constructed from data, or solely from expert opinion in the absence of data. Given $G$, there is a straightforward Monte Carlo solution to the uncertainty analysis problem. We sample $\mathbf{x}_1, \ldots, \mathbf{x}_N$ from $G$, and then evaluate $y_1 = \eta(\mathbf{x}_1), \ldots, y_N = \eta(\mathbf{x}_N)$ to obtain a sample from the distribution of $Y$. This is adequate if $\eta$ is computationally cheap, since we can obtain accurate estimates of any summary from the distribution of $Y$ by making $N$ sufficiently large. For computationally expensive functions $\eta$, simple Monte Carlo may not be practical, as we may be limited to a fairly small number of runs $N$. We can improve on simple Monte Carlo using variance reduction methods, such as Latin Hypercube sampling (McKay et al., 1979).

If a Gaussian process emulator has been constructed, uncertainty analysis can be performed without doing any further model runs. Recall that in the emulator framework, we are treating $\eta(\cdot)$ as an uncertain function, and it is important within an uncertainty analysis to distinguish between uncertainty about $Y$ due to uncertainty about $\mathbf{X}$, and uncertainty about $\eta(\cdot)$. Hence we define the conditional

distribution $p\{Y|\eta(\cdot)\}$ to be the uncertainty distribution, which itself is uncertain. By considering uncertainty about $p\{Y|\eta(\cdot)\}$, we are able to quantify how any uncertainty analysis results obtained using the emulator might change if we were to obtain further simulator runs. This would not be possible if we merely considered the marginal distribution of $Y$.

Haylock and O'Hagan (1996) consider inference for the mean and variance of the uncertainty distribution. For example, the mean is given by

$$M = E\{Y|\eta(.)\} = \int \eta(\mathbf{x}) dG(\mathbf{x})$$

which is normally distributed for a Gaussian process $\eta(\cdot)$, and Haylock and O'Hagan (1996) derive expressions for the mean and variance of $M$. Oakley and O'Hagan (2002) consider inferences for the distribution and density functions of the uncertainty distribution using simulation. Both Haylock and O'Hagan (1996) and Oakley and O'Hagan (2002) show that uncertainty analysis with emulators can be considerably more efficient in terms of simulator runs than Monte Carlo.

# 4 Sensitivity analysis

The term 'sensitivity analysis' is used widely in modelling, and broadly refers to the process of investigating whether alternative assumptions or modelling choices lead to different predictions or inferences. Within the field of computer experiments, we make a distinction between 'local' and 'global' sensitivity analysis.

In local sensitivity analysis, the aim is to quantify the change in output due to small perturbations of the input from some 'central' value, and typically involves the consideration of partial derivatives $\partial\eta(\mathbf{x})/\partial x_i$ (see Turanyi and Rabitz, 2000, for a review). If the function $\eta$ is nonlinear in its inputs, and small perturbations of the inputs do not adequately reflect our input uncertainty, then a local sensitivity analysis is unlikely to be sufficient. In this case, we should conduct a global sensitivity analysis, in which we investigate how the output varies as the inputs vary over some range. If we are considering reducing uncertainty about model inputs by collecting more data, a global sensitivity may identify how to prioritize data collection by identifying the most important uncertain inputs.

A simple form of global SA is known as 'one-way sensitivity analysis', in which the effect of an input is determined by varying it over some range while holding all other inputs fixed. This may give misleading results when there are correlations between the inputs, or for a non-linear function $\eta$. We return to this issue in the next section.

In the following sections, we review two different approaches to global SA: variance-based methods, and decision-theoretic approaches based on the expected value of perfect information.

## 4.1 Variance based sensitivity analysis

The variance-based approach to (global) sensitivity analysis is reviewed in Chan et al. (2000), and applications can be found in Saltelli and Tarantola (2002). The two most useful measures of input importance within the variance-based approach are the main effect index and the total sensitivity index. A third concept, related to the main effect index, is the main effect plot, which can be used to display graphically the relationship between an input and the output.

We start with a decomposition of the function $\eta(.)$ into main effects and interactions. We write

$$y = \eta(\mathbf{x}) = E(Y) + \sum_{i=1}^{d} z_i(x_i) + \sum_{i<j} z_{i,j}(x_i, x_j) + \sum_{i<j<k} z_{i,j,k}(x_i, x_j, x_k) + \ldots + z_{1,2,\ldots,d}(\mathbf{x})\,. \tag{6}$$

where

$$\begin{aligned}
z_i(x_i) &= E(Y \mid x_i) - E(Y)\,, \\
z_{i,j}(x_i, x_j) &= E(Y \mid x_i, x_j) - z_i(x_i) - z_j(x_j) - E(Y)\,, \\
z_{i,j,k}(x_i, x_j, x_k) &= E(Y \mid x_i, x_j, x_k) - z_{i,j}(x_i, x_j) - z_{i,k}(x_i, x_k) - z_{j,k}(x_j, x_k) \\
&\quad - z_i(x_i) - z_j(x_j) - z_k(x_k) - E(Y)\,,
\end{aligned}$$

and so on. We refer to $z_i(x_i)$ as the main effect of $x_i$, to $z_{i,j}(x_i, x_j)$ as the first-order interaction between $x_i$ and $x_j$, and so on. The decomposition (6) is referred to as the ANOVA high-dimensional model representation (HDMR) in Rabitz et al. (1999), and is itself used as to construct metamodels of $\eta(.)$ by some authors (see the discussion in section 4.1.1). Wahba et al. (1995) use a form of this decomposition for modelling observational data in their "smoothing spline ANOVA" models.

A main effect plot consists of a plot of $z_i(x_i)$ against $x_i$, and can be helpful in visualising the input-output relationship. We argue that main effects plots are superior to one-way sensitivity analyses in two regards. Firstly, if $X_i$ is correlated with other elements of $\mathbf{X}$, then it is not appropriate to hold these elements fixed regardless of the value of $X_i$, as we would expect these correlated inputs to vary as $X_i$ does. Secondly, if $\eta$ is a nonlinear function of $\mathbf{X}$, plugging in a point estimate of the inputs does not necessarily result in a 'good' estimate of the output, the value of $E\{\eta(\mathbf{X})\}$ may be very different to that of $\eta\{E(\mathbf{X})\}$. Both these issues are addressed

in the main effect plot by averaging over the conditional distribution of $\mathbf{X}_{-i}$ given $X_i$ when computing $E(Y|X_i)$.

The main effect index for input $X_i$ is defined as the variance of the main effect of $X_i$, normalised by the total variance:

$$\frac{Var_{X_i}\{Z(X_i)\}}{Var(Y)} = \frac{Var_{X_i}(E(Y|X_i))}{Var(Y)}.$$

This term is also known as the correlation ratio (McKay, 1997). Note that $Var_{X_i}\{E(Y|X_i)\} = Var(Y) - E_{X_i}\{Var(Y|X_i)\}$, and so the variance of the main effect can be interpreted as the expected proportional reduction in the variance of $Y$ obtained by learning the value of $X_i$. Hence if we choose one input to learn precisely, and wished to obtain the largest reduction in variance, we should choose the input with the largest main effect index.

We illustrate main effects plots and main effects indices in figure 2. Here, we have less uncertainty about $X_i$ than $X_j$, but $X_i$ has a larger main effect index than $X_j$, due to the stronger input-output relationship, as seen in the main effect plot.

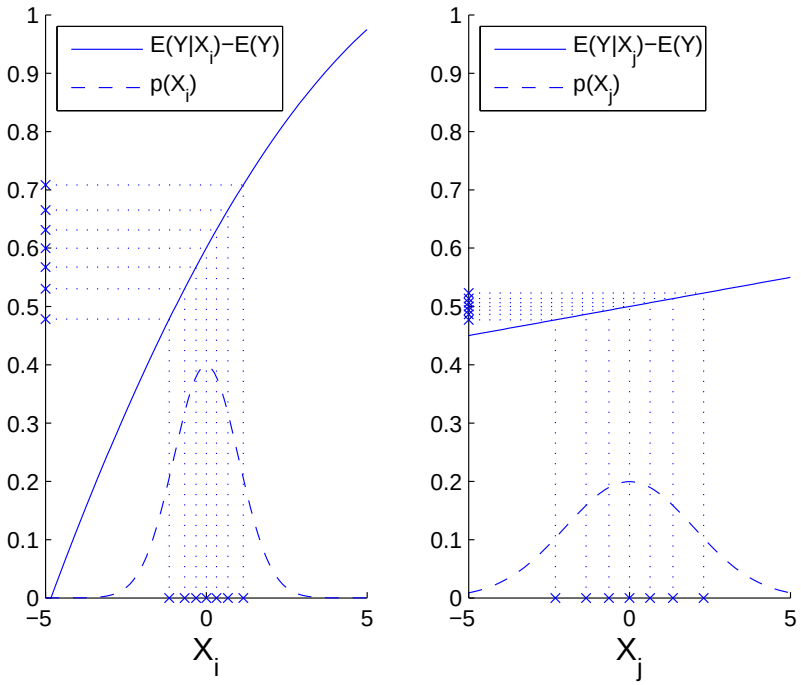


Figure 2: Main effects (solid lines), with the points projected onto the y-axes showing the variance of the main effects. The main effect index for an uncertain input combines uncertainty about the input together with the individual input-output relationship to produce a measure of the input importance.

An input with a small main effect index may still be influential due to interactions

within the decomposition (6). For example, consider the function $Y = X_1 + X_1X_2$, with $X_1, X_2$ independent $N(0,1)$ random variables. The main effect index of $X_2$ is 0, but uncertainty about $X_2$ does contribute to uncertainty in $Y$. This motivates the use of a a second measure of importance, proposed by Homma and Saltelli (1996), known as the total effect index:

$$\frac{Var(Y) - Var\{E(Y \mid \mathbf{X}_{-i})\}}{Var(Y)}.$$

The numerator is the expected reduction in variance obtained by learning all inputs except $X_i$. In the case of independent inputs, insight into the interpretation of total effect index can be obtained through the following decomposition of the output variance. For independent inputs, we can use (6) to obtain the variance decomposition

$$Var(Y) = \sum_{i=1}^{d} W_i + \sum_{i<j} W_{i,j} + \sum_{i<j<k} W_{i,j,k} + \ldots + W_{1,2,\ldots,d}\,, \tag{7}$$

where if $p$ is the set of indices for the subvector $\mathbf{x}_p$,

$$W_p = Var\{z_p(\mathbf{X}_p)\}\,.$$

This holds because in the case of independent inputs, all the terms in (6) are uncorrelated. We can now obtain $Var\{E(Y \mid \mathbf{X}_{-i})\}$ by summing all the $W_p$ terms that do not include the subscript $i$, and so the total effect index for $X_i$ is the sum of the main effect index and all interaction terms involving $X_i$. A total effect index close to 0 implies that $X_i$ has a negligible contribution to the uncertainty in $Y$, and that the model can be simplified by fixing $X_i$.

For correlated inputs, total effect indices may not be appropriate measures of input importance. Consider the function $Y = X_1 + \varepsilon X_2$, where $X_1$ and $X_2$ have a bivariate normal distribution, with $E(X_i) = 0$, $Var(X_i) = 1$ for $i = 1, 2$, and $Cov(X_1, X_2) = 0.5$. For sufficiently small $\varepsilon$, the total effect index of $X_2$ is close to zero, as learning $X_1$ will remove almost all the uncertainty about $Y$. However, the main effect index for $X_2$ is $(0.5 + \varepsilon)^2/(1 + \varepsilon + \varepsilon^2)$, which is non-negligible for small $\varepsilon$.

### 4.1.1 Computation

There are various computational methods for estimating sensitivity indices. One of the earliest proposed approaches was the Fourier amplitude sensitivity test (FAST, Cukier et al., 1973). This involves evaluating simulator outputs at inputs along a

curve which explores the input space, oscillating at different frequencies in each input dimension. Other methods involve improvements on simple Monte Carlo sampling. For example, Morris et al. (2008) propose sampling schemes based on balanced incomplete block designs.

Again, emulators can be used for computationally expensive simulators. Oakley and O'Hagan (2004) use the Gaussian process emulator to compute sensitivity indices and produce main effects plots, and their method has been implemented in the free software package GEM-SA (available from `http://www.tonyohagan.co.uk/academic/GEM/i`). Similar methods based on the Gaussian process emulator are given in Marrel et al. (2009).

The approach of Oakley and O'Hagan (2004) is to first build an emulator of $\eta(.)$, which can then be used to estimate the terms in (6) and (7). An alternative approach is to directly estimate the terms in (6), usually assuming that the higher order terms will be negligible, and then calculate the sensitivity indices from these (and obtain a meta-model for $\eta(.)$ if required). Ratto et al. (2007), use a time series method known as state dependent parameter modelling, and Ziehn and Tomlin (2009) approximate $z_i(x_i)$ and $z_{i,j}(x_i, x_j)$ using orthonormal polynomials. Both papers report computational efficiency comparable with Oakley and O'Hagan (2004) in terms of the simulator runs required. A general metamodelling approach based on (6) is presented in Reich et al. (2009) which involves modelling the individual $z_i(x_i)$ and $z_{i,j}(x_i, x_j)$ terms with Gaussian processes.

## 4.2 Value of information

A more general approach to sensitivity analysis involves the use of decision-theoretic arguments. We suppose that the simulator user is going to choose a decision $d$ from a set of possible decisions $\mathcal{D}$. The simulator user's utility of decision $d$ is dependent on the value of the true output $Y$, and we write the simulator user's utility function as $U(d, Y) = U\{d, \eta(\mathbf{X})\}$. Based on no additional information, the optimal decision will maximise the simulator user's expected utility, obtaining a baseline utility

$$U^* = \max_{d \in \mathcal{D}} E\{U(d, Y)\}.$$

Now suppose the simulator user considers learning the true value of an uncertain input $X_i$ before making a decision. Ignoring any costs in learning $X_i$, the expected utility given the value of $X_i$ is simply

$$\max_{d \in \mathcal{D}} E_{\mathbf{X}_{-i}|X_i}\{U(d, Y)\},$$

and so the expected increase in utility of learning $X_i$ is

$$E_{X_i}\left[\max_{d\in\mathcal{D}} E\{U(d,Y)|X_i\}\right] - U^*. \tag{8}$$

This term is known as the partial EVPI (expected value of perfect information) for input $X_i$.

A similar argument can be used to quantify the expected value of reducing uncertainty by collecting data $S$. This is known as the expected value of sample information, and is given by

$$E_S\left[\max_{d\in\mathcal{D}} E\{U(d,Y)|S\}\right] - U^*.$$

As an example, suppose we have a choice of two decisions, $d_1$ and $d_2$, and that the expected utilities of each decision, conditional on a particular input $X_i$ are as follows:

$$\begin{aligned} E\{U(d_1,Y)|X_i\} &= 7 - X_i - 3X_i^2, \\ E\{U(d_2,Y)|X_i\} &= 6 - X_i, \end{aligned}$$

and suppose $X_i \sim Beta(2,9)$ (see figure 3, left plot). Then, given no further information, $E\{U(d_1,Y)\} = 6.682$ and $E\{U(d_2,Y)\} = 5.818$, so that $d_1$ would be the optimal decision. If it were known that $X_i > 1/\sqrt{3}$, then the optimal decision would switch to $d_2$. However, $P(X_i > 1/\sqrt{3}) = 0.003$, and so it unlikely that learning the true value of $X_i$ would change the optimal decision, and the partial EVPI of $X_i$, given by

$$\int_{1/\sqrt{3}}^{1} (3x^2 - 1)p_{X_i}(x)dx = 4.4 \times 10^{-4},$$

is very small.

Alternatively, suppose we have $X_i \sim Beta(2,3)$ (see figure 3, right plot). Now we have $E\{U(d_1,Y)\} = 6$ and $E\{U(d_2,Y)\} = 5.6$, so $d_1$ is still the optimal decision, but $P(X_i > 1/\sqrt{3}) = 0.21$, and so it more likely that we would change our decision were we to learn $X_i$. The partial EVPI of $X_i$ is now 0.095.

The variance-based measure $Var_{X_i}\{E(Y|X_i)\}$ is a special case of the partial EVPI under a quadratic loss function: $Var_{X_i}\{E(Y|X_i)\}$ will equal (8) if the decision problem is to estimate $Y$, and the utility of the estimate is proportional to $-(d-Y)^2$. For other types of decision problem, the two approaches may give quite different results. A simple example given in Oakley (2009) is the function $Y = X_1X_2$, where $X_1 \sim N(1,1)$ and $\log X_2 \sim N(0,1)$, and the decision problem is to state the sign of $Y$. A variance-based sensitivity analysis would give us $Var_{X_1}\{E(Y|X_1)\} = 2.7$,

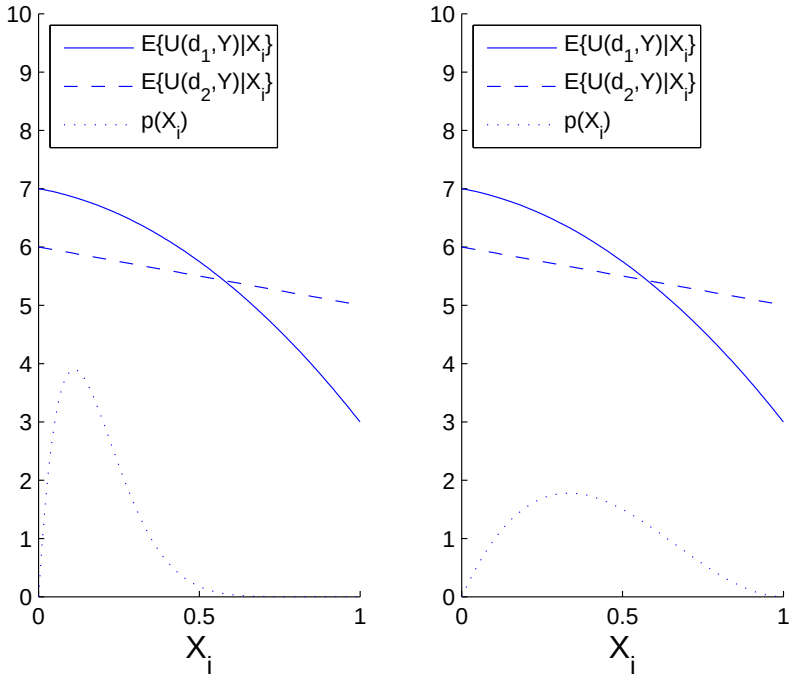


Figure 3: Left plot: $d_2$ is only believed to be better than $d_1$ for very unlikely values of $X_i$, and so $X_i$ has a very small partial EVPI. Right plot: $X_i$ is judged to be more important here, and has a larger partial EVPI, as larger plausible values of $X_i$ would switch the optimal decision from $d_1$ to $d_2$.

and $Var_{X_2}\{E(Y|X_2)\} = 4.6$, indicating $X_2$ is 'more important' than $X_1$. However, for the purpose of establishing if $Y > 0$, the variable $X_1$ is clearly more important than $X_2$, as the event $Y > 0$ is determined by the sign of $X_1$ only.

An obvious practical difficulty in the decision-theoretic approach is in the specification of an appropriate utility/loss function. In the field of health economic modelling, in which there is usually a clearly defined decision problem and utility function, partial EVPIs have been advocated by Felli and Hazen (1998) and Claxton (1999). However, we are not aware of their use in other disciplines.

Monte Carlo sampling can be used to estimate partial EVPIs (see Brennan et al., 2007), but again, in the case of computationally expensive models this may not be practical due to the numbers of model runs typically required. Oakley (2009) shows how Gaussian process emulators can be used to obtain estimates more efficiently in this case.

# 5 Calibration and discrepancy

In the calibration problem, we have physical observations or *field data* $\mathbf{z}_1, \ldots, \mathbf{z}_n$ and we wish to find values of the simulator inputs such that the simulator predictions are consistent with the physical observations. Following Kennedy and O'Hagan (2001), we make the distinction between *control* inputs $\mathbf{x}$ and *calibration* inputs $\boldsymbol{\theta}$, and suppose that the control inputs are known to the simulator user and may vary for each physical observation. For example, the control inputs may represent spatial coordinates, with physical observations taken at different points in space, so that we have $\mathbf{z}_1(\mathbf{x}_1), \ldots, \mathbf{z}_n(\mathbf{x}_n)$. The problem of calibrating simulator inputs to field data is also known as an *inverse problem* or *history matching* in other disciplines.

To make inferences about $\boldsymbol{\theta}$, we must consider the *discrepancy* between the simulator and reality. The simulator is unlikely to be a perfect representation of reality, perhaps due to gaps in our knowledge of the science of the underlying process, or computational limitations in modelling the process. Kennedy and O'Hagan (2001) proposed the following model to relate the observed data to the simulator

$$\mathbf{z}(\mathbf{x}_i) = \rho\eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \varepsilon_i, \tag{9}$$

where $\varepsilon_i$ is a measurement error, and $\delta(.)$ represents the discrepancy between the simulator, evaluated at its true calibration input $\boldsymbol{\theta}$, and reality.

Kennedy and O'Hagan (2001) model $\delta(\cdot)$ as a second Gaussian process, independent of $\rho\eta(., \boldsymbol{\theta})$, for similar reasons to those for modelling the simulator as a Gaussian process: it is tractable, reflects a judgment that errors in simulator predictions are likely to be similar for similar values of the control inputs, but does not require us to assume a particular parametric form for $\delta(.)$.

Given observations $\mathbf{z}_1(\mathbf{x}_1), \ldots, \mathbf{z}_n(\mathbf{x}_n)$ (and training runs of the simulator if it is computationally expensive), we first derive the posterior distribution of $\theta, \rho, \delta(.)$ and the associated Gaussian process hyperparameters. We can then make calibrated predictions, allowing for uncertainty in these parameters and the discrepancy between the model and reality, through the process $\delta(.)$. Illustrations and further developments of this approach can be found in Higdon et al. (2004), Bayarri et al. (2007a), Bayarri et al. (2007b) and Higdon et al. (2008). A similar approach for calibration and prediction within the framework of Bayes linear methods is developed in Craig et al. (1996), Craig et al. (2001) and Goldstein and Rougier (2006).

This model again raises the issue of what is meant by a 'true' input. The parameters $\boldsymbol{\theta}$ may correspond to quantities observable in the real world, and hence have true values in reality. However, if the simulator is not a perfect representation of

reality, then running the simulator at the true inputs may not necessarily produce the corresponding true values of the output, and the simulator user may achieve more accurate predictions with an 'incorrect' choice of input. In (9), $\boldsymbol{\theta}$ cannot be interpreted as the true physical parameter, only as a best-fitting parameter such that the error structure in the residuals is correct for the observed data.

A more advanced approach to modelling simulator discrepancy is presented in Goldstein and Rougier (2009), within the context of what the authors term 'reified' modelling. This involves eliciting expert judgements about the possible effects of improvements to the simulator, for example by including additional inputs. These judgements are used to link $\eta(.,.)$ to an idealised simulator known as the reified simulator, with the reified simulator run at its best input judged to be independent of the (reified) simulator discrepancy. An example is given involving a compartmental simulator of the Atlantic Ocean, in which judgements are made about the effects of enhancing the structure of the simulator through the addition of an extra compartment.

The notion of 'statistically' improving a simulator $\eta(.)$ through the discrepancy term $\delta(.)$ may appear to be in conflict with the process of improving a simulator through better understanding and incorporation of the underlying science, but the appearance of any conflict is illusory. The latter process may require months or years of development, whereas the former can be implemented within minutes or hours, and is simply intended to provide the best predictions with the simulator available at hand. The example presented in Kennedy and O'Hagan (2001) involved calibrating an atmospheric dispersion model and predicting deposition of radionuclides following an accidental release, in which decisions based on model predictions would be required in a short time period following the release. In any case, unless we expect a simulator to be *perfect*, we should consider discrepancy as a means of better quantifying uncertainty in the simulator predictions.

# 6 Discussion

In this chapter we have reviewed various statistical tools for quantifying and investigating uncertainty in deterministic simulators. The techniques are intended to alleviate some of the difficulties associated with complex simulators. Complex simulators can be undesirable due to the prohibitive computing times required to perform simulator runs at large numbers of different input values. This has been addressed with some success with the use of emulators, although Gaussian process emulators are currently restricted to moderate numbers of inputs, and emulating

simulators with very large numbers of inputs remains a challenge.

Global sensitivity analysis methods can be used to better understand the input-output relationships in a complex simulator. They can establish whether a simulator can be simplified, by identifying inputs that have little effect on the output (though not without both building the full simulator and specifying appropriate input distributions in the first place). They can also suggest where more complex modelling may be necessary, for example, if an simulator input is highly influential and is itself the output of some other process.

Finally, we have the discussed the concept of simulator discrepancy. Modelling discrepancy is arguably the most difficult part of the process, particular if there is little or no physical data. Nevertheless, such modelling is an essential component in fully quantifying the uncertainty in any simulator prediction, and is one of the most important research problems in the field.

# References


Bastos, L. S. and O'Hagan, A. (2009). Diagnostics for Gaussian process emulators, *Technometrics*, **51**: 425–438.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donate, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J. and Walsh, D. (2007a). Computer model validation with functional output, *The Annals of Statistics*, **35**: 1874–1906.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H. and Tu, J. (2007b). A framework for validation of computer models, *Technometrics*, **49**: 138–154.

Blight, B. J. N. and Ott, L. (1975). A Bayesian approach to model inadequacy for polynomial regression, *Biometrika*, **62**: 79–88.

Brennan, A., Kharroubi, S., O'Hagan, A. and Chilcott, J. (2007). Calculating partial expected value of perfect information via monte carlo sampling algorithms, *Medical Decision Making*, **27**: 448–470.

Busby, D. (2009). Hierarchical adaptive experimental design for Gaussian process emulators, *Reliability Engineering and System Safety*, **94**: 1183–1193.

Chan, K., Tarantola, S., Saltelli, A. and Sobol', I. M. (2000). Variance-based methods, in *Sensitivity Analysis*, edited by Saltelli, A., Chan, K. and Scott, M., New York: Wiley.

Claxton, K. (1999). Bayesian approaches to the value of information: implications for the regulation of new health care technologies, *Health Economics*, **8**: 269–274.

Conti, S. and O'Hagan, A. (2007). Bayesian emulation of complex multi-output and dynamic computer models, Tech. Rep. 569/07, Department of Probability and Statistics, University of Sheffield, Submitted to *Journal of Statistical Planning and Inference*.

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

Craig, P. S., Goldstein, M., Rougier, J. C. and Seheult, A. H. (2001). Bayesian forecasting for complex systems using computer simulators, *J. Am. Statist. Assoc*, **96**: 717–729.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*, New York: Wiley.

Cukier, R. I., Fortuin, C. M., Schuler, K. E., Petschek, A. G. and Schaibly, J. H. (1973). Study of the sensitivity of coupled systems to uncertainties in rate coefficients, *J. Chem. Phys.*, **59**: 3873–3878.

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

El Tabach, E., Lancelot, L., Shahrour, I. and Najjar, Y. (2007). Use of artificial neural network simulation metamodelling to assess groundwater contamination in a road project, *Mathematical and Computer Modelling*, **45**: 766–776.

Felli, J. C. and Hazen, G. B. (1998). Sensitivity analysis and the expected value of perfect information, *Medical Decision Making*, **18**: 95–109.

Goldstein, M. and Rougier, J. C. (2006). Bayes linear calibrated prediction for physical systems, *J. Am. Statist. Assoc.*, **101**: 1132–1143.

Goldstein, M. and Rougier, J. C. (2009). Reified Bayesian modelling and inference for physical systems (with discussion and rejoinder), *Journal of Statistical Planning and Inference*, **139**: 1221–1239.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling, *J. Am. Statist. Assoc.*, **103**: 1119–1130.

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

Higdon, D., Gattiker, J., Williams, B. and Rightly, M. (2008). Computer model calibration using high-dimensional output, *J. Am. Statist. Assoc.*, **103**: 570–583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A. and Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction, *Siam J. Sci. Comput.*, **26**: 448–466.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis, *Reliability Engineering and System Safety*, **52**: 1–17.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian callibration of complex computer models (with discussion), *J. Roy. Statist. Soc. B*, **63**: 425–464.

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

Marrel, A., Iooss, B., Laurent, B. and Roustant, O. (2009). Calculations of Sobol indices for the Gaussian process metamodel, *Reliability Engineering and System Safety*, **94**: 742–751.

McKay, M. D. (1997). Nonparametric variance-based methods of assessing uncertainty importance, *Reliab. Engng. Syst. Safety*, **57**: 267–279.

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

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

Morris, M. D., Moore, L. M. and McKay, M. D. (2008). Using orthogonal arrays in the sensitivity analysis of computer models, *Technometrics*, **50**: 205–215.

Nagy, B., Loeppky, J. and Welch, W. J. (2007). Fast Bayesian inference for Gaussian process models, Tech. Rep. 230, Dept. Statistics, Univ. British Columbia.

Neal, R. (1999). Regression and classification using gaussian process priors, in *Bayesian Statistics 6*, edited by Bernardo, J. M., Berger, J. O., Dawid, A. P. and Smith, A. F. M., pp. 69–95, Oxford: University Press.

Oakley, J. E. (2002). Eliciting Gaussian process priors for complex computer codes, *The Statistician*, **51**: 81–97.

Oakley, J. E. (2004). Estimating percentiles of computer code outputs, *Applied Statistics*, **53**: 83–93.

Oakley, J. E. (2009). Decision-theoretic sensitivity analysis for complex computer models, *Technometrics*, **51**: 121–129.

Oakley, J. E. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs, *Biometrika*, **89**: 769–784.

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity of complex models: a Bayesian approach, *J. Roy. Statist. Soc. Ser. B*, **66**: 751–769.

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

Rabitz, H., Alis, O. F., Shorter, J. and Shim, K. (1999). Efficient input-output model representations, *Computer Physics Communications*, **117**: 11–20.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*, Cambridge, MA: The MIT Press.

Ratto, M., Pagano, A. and Young, P. (2007). State dependent parameter metamodelling and sensitivity analysis, *Computer Physics Communications*, **177**: 863–876.

Reich, B. J., Storlie, C. B. and Bondell, H. D. (2009). Variable selection in Bayesian smoothing spline ANOVA models: application to deterministic computer codes, *Technometrics*, **51**: 110–121.

Rougier, J. C. (2008). Emulators for multivariate deterministic functions, *Journal of Computational and Graphical Statistics*, **17**: 827–843.

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. and Tarantola, S. (2002). On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal, *J. Am. Statist. Assoc.*, **97**: 702–709.

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

Turanyi, T. and Rabitz, H. (2000). Local methods, in *Sensitivity Analysis*, edited by Saltelli, A., Chan, K. and Scott, M., New York: Wiley.

Wahba, G., Wang, Y., Gu, C., Klein, R. and Klein, B. (1995). Smoothing sline ANOVA for exponential families, with application to the Winsconsin epidemiological study of diabetic retinopathy, *The Annals of Statistics*, **23**: 1865–1895.

Ziehn, T. and Tomlin, A. S. (2009). GUI-HDMR - a software tool for global sensitivity analysis of complex models, *Environmental Modelling and Software*, **24**: 775–785.