Bayes Linear Calibrated Prediction for Complex Systems

Michael Goldstein and Jonathan Rougier*

Department of Mathematical Sciences

University of Durham, U.K.

Abstract

A calibration-based approach is developed for predicting the behaviour of a physical system which is modelled by a computer simulator. The approach is based on Bayes linear adjustment using both system observations and evaluations of the simulator at parameterisations which appear to give good matches to those observations. This approach can be applied to complex high-dimensional systems with expensive simulators, where a fully-Bayesian approach would be impractical. It is illustrated with an example concerning the collapse of the Thermohaline Circulation (THC) in the Atlantic.

Key words: Computer Experiment, Model Diagnostics, Simulator, Emulator, Calibration, Hat Run, Thermohaline Circulation (THC)

*Michael Goldstein is Professor and Jonathan Rougier is Research Fellow, Department of Mathematical Sciences, University of Durham, Science Laboratories, South Road, Durham DH1 3LE, U.K. (E-mail: J.C.Rougier@durham.ac.uk).

1 Introduction

Computer simulators are used to make inferences about physical systems in a very wide range of applications. Much of the interest is in using computer simulators, in conjunction with historic and current system data, to make predictions about future system behaviour. The use of computer simulators is necessary because the medium- and long-term behaviour of many physical systems follows natural laws that cannot be deduced from currently-available data. A high-profile example of the role of computer simulators in policy-making is the ongoing debate on climate change (Houghton et al., 2001).

The computer simulator takes input, describing the properties of the system, and returns output, typically describing the evolution of the state of the system through time. There are three major sources of uncertainty. Firstly, the correct value for the input to the simulator is not known: in many cases the very notion of a correct setting may be debatable. Secondly, the simulator is an imperfect analogue of the system. Part of this is due to uncertainty about the precise values of some of the simulator's parameters, referred to as the simulator input. But even with the best choice for the simulator input, the output will almost certainly not correspond exactly to the system behaviour. Finally, if there are data involved, there is the uncertainty induced by measurement error. These substantial uncertainties, together with necessary involvement of experts, suggest that it is natural to adopt a Bayesian approach (Kennedy and O'Hagan, 2001). However, the severe computational demands restrict the application of a fully-Bayesian treatment to problems of moderate size and complexity. For larger problems the more tractable Bayes linear approach has had some success (Craig et al., 1996, 1997, 2001), and avoids much of the computational burden of the fully-Bayesian approach.

In the Bayesian approaches, the runs of the simulator are used to inform

a belief model of the simulator. This belief model, in conjunction with the various uncertainties linking the simulator to the system, determines a likelihood for the simulator inputs using the observed system data, which may be inverted via Bayes's Theorem to give a posterior predictive distribution for system behaviour. This is in contrast to the more traditional use of computer simulators, in which the objective is to locate the best input and then make inferences based solely on the simulator output at this input. To follow the traditional approach, we must first find, approximately, the best-fitting value of the inputs, by matching the historical data to simulator output. This is often termed 'calibration' or 'tuning' the simulator; often, calibration is important even when there is no requirement for subsequent forecasting. The values of other simulator outputs, for example those components corresponding to future system behaviour, are then used as point predictions for the system. This approach has the virtue of making good use of the simulator by running it at the best-fitting input value.

In this paper we propose a synthesis of these two approaches, that has the advantages of both and yet is tractable enough to be applied in large problems. The essence of our approach is to use the available system data twice. The data are first used to estimate the best-fitting input. The simulator is then evaluated at this input, which we term the 'hat run'. Comparison between the output of the hat run and the corresponding values of the data may be used as a diagnostic assessment of our ability to calibrate the model. Second, the data are used again, in conjunction with the result of the hat run, for linear prediction of the system values. We term this 'Bayes linear calibrated prediction'. The technical aspect of the analysis lies in the assessment of the full covariance structure between the data, the system and the hat run. In this assessment, we make explicit the dependence of the hat run on the system

data, so that when we predict the system behaviour, the double use of the data is correctly accounted for.

In section 2 we outline the fully-Bayesian approach to calibrated prediction, introduce the notion of an *emulator*, and discusses limitations for complex systems and large simulators. Section 3 describes an alternative Bayes linear treatment that is scalable to large simulators, and contrasts this with the fully-Bayesian approach. Section 4 introduces the 'hat run', an extension of the Bayes linear treatment that introduces an element of calibration that, in certain circumstances, may dramatically reduce predictive uncertainty. The hat run approach is illustrated in section 5, where a simple model of the Atlantic is calibrated to data in order to predict the point at which the Thermohaline Circulation (THC) breaks down: a highly topical issue that dominates the discussion of the impact of global warming on Western European climate. Section 6 concludes, and an Appendix contains a description of the mathematical treatment necessary to implement the hat run approach.

2 Calibrated prediction

In this section we outline the fully Bayesian approach to calibrated prediction using a computer simulator. The treatment is drawn from Craig et al. (2001), Kennedy and O'Hagan (2001), Goldstein and Rougier (2005a,b), Higdon et al. (2005) and Rougier (2005).

2.1 The general problem

Our starting point is a physical system. Denote the system value as $y \in \mathcal{Y}$; typically y is a collection of space- and time-indexed quantities. Often we also

have some observations on y, denoted z. We will write

$$z = Hy + e \tag{1}$$

where H is the incidence matrix showing which linear combinations of y have been measured, and e is the measurement error, which is taken to be independent of all other uncertain quantities. Often the rows of H are simply rows from the identity matrix. In large applications, the rows of H might correspond to temporal averaging or spatial interpolation; in the latter case we might include an additional error component in e to account for imprecision in our linear mapping from y to z.

To specify $\mathsf{Pr}(y)$, we start with a simulator for the system, usually implemented as computer code. This simulator can embody dynamic physical laws, such as conservation laws, that would be very difficult to describe probabilistically. The simulator is a deterministic function $f : \mathcal{X} \to \mathcal{Y}$, where $x \in \mathcal{X}$ is a collection of model parameters about which we are uncertain. This uncertainty could arise because the model parameters correspond to physical quantities whose values we do not know, or which are inferred from proxy data (e.g., initial conditions, or 'historic' forcing functions), or because they correspond to quantities with no accurately measurable physical analogue (e.g., in submodels standing in for missing or unknown physics), or because the simulator is of sufficiently poor quality to induce uncertainty about the appropriate values of even measurable quantities (e.g., through excluding the effect of sub-grid-scale processes).

For each x, we are also uncertain about the value of $f(x)$, until we choose to evaluate this value. If f is slow to evaluate and $\mathcal{X}$ is high dimensional, then we will only have a very limited number of evaluations on which to condition

our initial uncertainty about the values taken by f. Suppose that we have n evaluations of our simulator at inputs $X \triangleq \big(x_1, \ldots, x_n\big)$, where we denote the resulting evaluations as $F \triangleq \big(f(x_1), \ldots, f(x_n)\big)$. The evaluations $(F; X)$ are informative for the function f and thus for y. To exploit this information, we must describe how f and y are related. A common formulation in the literature is the *Best Input* assumption (Goldstein and Rougier, 2005a), which we will adopt here:

Best Input Assumption. *There exists a value $x^* \in \mathcal{X}$ where $x^* \perp\!\!\!\perp f$, such that $f^* \triangleq f(x^*)$ is sufficient, both for the function f and for the value x^*, for assessing uncertainties for y.*

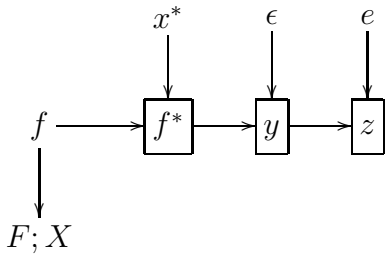
It is helpful to parameterise the relationship between f^* and y in terms of the *discrepancy*,

$$\epsilon \triangleq y - f^*. \tag{2}$$

The Best Input assumption implies that $\epsilon \perp\!\!\!\perp (f, x^*)$. Often we choose to set $\mathsf{E}(\epsilon) = \mathbf{0}$. Our choice for $\mathsf{Var}(\epsilon)$ is indicative of how good a model we believe f to be. The structure of $\mathsf{Var}(\epsilon)$ reflects the structure of $\mathcal{Y}$. For example, spatial and/or temporal indices for $\mathcal{Y}$ would typically be reflected in positive off-diagonal elements in $\mathsf{Var}(\epsilon)$; see, for example, Craig et al. (2001, section 6). Another approach to assessing the off-diagonal structure of $\mathsf{Var}(\epsilon)$ is discussed in Goldstein and Rougier (2005b).

Using the Best Input assumption, we can represent our uncertainties on

the following Bayesian belief net:



(3)

where the boxes indicate vertices that are completely determined by their parents. We require marginal distributions for each member of the collection (f, x^*, ϵ, e). Both z and $(F; X)$ are instantiated, and f, ϵ and e are typically treated as nuisance quantities in order to make inferences for (x^*, y). Our inference takes the form of a *calibrated prediction* based on the conditional distribution $\mathsf{Pr}(x^*, y \mid z, F; X)$. The marginal conditional distribution $\mathsf{Pr}(x^* \mid z, F; X)$ is the Bayesian solution to the *statistical inverse problem*: finding the input value which gave rise to the noisy observations z. We often refer to this analysis as *calibration*. Likewise, the marginal conditional distribution $\mathsf{Pr}(y \mid z, F; X)$ is the *system prediction*. The Bayesian approach shows how these two tasks may be unified, so that their answers are consistent.

2.2 The emulator

The belief net in (3) allows us to integrate out f, replacing the vertex $f^* \mid (x^*, f)$ with $f^* \mid (x^*, F; X)$. The distribution of $f(x)$ conditional on $(x, F; X)$ is termed the *emulator* of f. While terminology may vary, use of emulators in computer experiments is well-established; see, e.g., the review in Santner et al. (2003). The emulator is a stochastic representation of the function updated with evaluations of that function at known inputs. The emulator allows us to interpolate or extrapolate the evaluations $(F; X)$ to beliefs about the sim-

ulator response at any $x \in \mathcal{X}$. Most of the evaluations of the emulator in inferential calculations for a large simulator are extrapolations from X, for which $\mathsf{Var}\big(f(x) \mid x, F; X\big)$ tends to be non-negligible even for a careful choice of X; see, e.g., Koehler and Owen (1996), for a discussion of design strategies for computer experiments.

2.3 Doing the fully-Bayesian calculation

The simplest implementation of the fully-probabilistic approach makes strong parametric assumptions for the nuisance parameters, f, ϵ and e. We assume that these are all gaussian with specified means and variances. In this case, integrating out f gives us a gaussian emulator parameterised by a mean function and a variance function. We can also integrate out ϵ and e. The result is that $(y, z) \mid x^*$ is gaussian, suppressing the conditioning on $(F; X)$, and we can factorise the calibrated prediction distribution as

$$\begin{aligned} \mathsf{Pr}(x^*, y \mid z = \tilde{z}) &\propto \mathsf{Pr}(x^*)\, \phi\big(y, \tilde{z} \mid x^*\big) \\ &= \mathsf{Pr}(x^*)\, \phi\big(\tilde{z} \mid x^*\big)\, \phi\big(y \mid z = \tilde{z}, x^*\big) \end{aligned} \tag{4}$$

where $\phi(\cdot \mid \cdot)$ denotes a gaussian Probability Density Function (PDF) with known mean and variance, and $\tilde{z}$ is the observed value of z. The product of the first two PDFs is proportional to the calibration distribution $\mathsf{Pr}(x^* \mid z = \tilde{z})$. Once we have a sample from this calibration distribution we can easily augment it with sampled values for $y \mid (z = \tilde{z}, x^*)$, given our gaussian assumptions. Therefore the effective dimension of the calibrated prediction problem with the gaussian assumptions is the dimension of the calibration problem, $\dim(\mathcal{X})$.

Any generalisation of this approach will lead to a calculation with an effective dimension larger than $\dim(\mathcal{X})$; e.g., including hyper-parameters for the prior variances of f or ϵ, or using a non-gaussian distribution for e. Sometimes

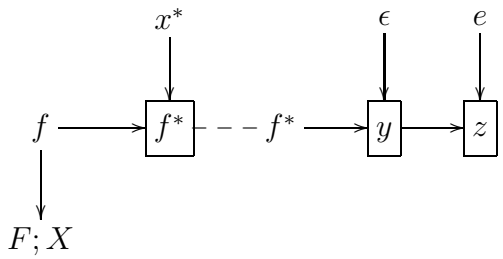
we can use plug-in values for hyper-parameters (Currin et al., 1991; Kennedy and O'Hagan, 2001), but this does not address the fundamental problem of a large input space. A 'large' simulator is one for which $\dim(\mathcal{X})$ is greater than we can sample effectively with current resources. Most simulators of complex physical systems are therefore 'large', bearing in mind that the uncertain quantities should include not just the model parameters but also the initial value of the state vector and 'historic' values of forcing functions. A fully-probabilistic approach can be applied in these circumstances by fixing some of the components of x^*. But in this case we run the risk of undermining the Best Input assumption, because if we are assuming the existence of a 'special' input, our assumption has more chance of being acceptable the larger is the input space (Goldstein and Rougier, 2005a).

Consequently there is a need for an alternative to the fully-probabilistic approach when making inferences about complex systems with large simulators. There is a similar need for systems with smaller simulators where the expert is reluctant to make a gaussian choice for (transformed) values of the nuisance parameters, either because he or she believes that the shape of the gaussian distribution is not appropriate, or, as often happens in practice, because of a general reluctance to make *any* type of fully-probabilistic specification. Finally, there is a similar need where rapid calculations are required for a variety of modelling scenarios.

3 The Bayes linear approach to prediction

We now describe a simpler approach to belief specification and prediction, as given in Craig et al. (2001). The crucial step is to split the fully-Bayesian

belief net given in (3) into two parts:



(5)

On the lefthand side of the belief net, evaluations of the simulator and beliefs about the simulator and the Best Input are used to compute the mean and variance of f^*. On the righthand side, the mean and variance for f^* are combined with beliefs about the simulator discrepancy ϵ and the measurement process to make a prediction for y using the values of z.

The mean and variance for f^* are derived from the mean function and variance function of the emulator for f, derived using $(F; X)$, which are the only features of the emulator that we are required to specify in this approach. These are defined as

$$\begin{aligned} \mu(x) &\triangleq \mathsf{E}\big(f(x)\big) \\ \kappa(x, x') &\triangleq \mathsf{Cov}\big(f(x), f(x')\big) \end{aligned} \tag{6}$$

where $\mu(\cdot)$ and $\kappa(\cdot, \cdot)$ are evaluated using $(F; X)$. In the fully-probabilistic case $\mu(x)$ and $\kappa(x, x')$ would be the mean and variance functions of $f(x)$ after conditioning on $(F; X)$. But the construction of these two functions using $(F; X)$ need not be by conditioning. It could be done more generally using a Bayes linear approach (Craig et al., 1997, 1998, 2001), or by Kriging, or informally in consultation with the modellers who built the simulator. Using the mean and variance functions we can compute the unconditional mean and

variance of f^* by first conditioning on x^*:

$$\begin{aligned} \mathsf{E}(f^*) &= \mathsf{E}\big(\mu(x^*)\big) \\ \mathsf{Var}(f^*) &= \mathsf{E}\big(\kappa(x^*, x^*)\big) + \mathsf{Var}\big(\mu(x^*)\big)\,, \end{aligned} \tag{7}$$

as $x^* \perp\!\!\!\perp f$. Depending on the form of the emulator, we may either need to specify certain prior moments for x^* or provide a full prior probability distribution for x^* from which the expectation of arbitrary functions of x^* can be computed. Either way, integrating x^* out of the emulator can be facilitated using look-up tables which can be prepared in advance (a similar approach is used in variance-based sensitivity analysis; see, e.g., Oakley and O'Hagan, 2004). Because the calculation of the mean and variance of f^* involve only simple algebraic operations it is possible to work with large input spaces, certainly much larger than could be handled using a fully-Bayesian approach and, say, simulation.

Given $\mathsf{E}(f^*)$ and $\mathsf{Var}(f^*)$, it is straightforward to compute the joint mean and variance of the collection (y, z), where

$$y \equiv f^* + \epsilon \qquad \text{and} \qquad z \equiv H(f^* + \epsilon) + e \tag{8}$$

with no gaussian requirement on the error terms. We now evaluate the adjusted mean and variance for y given z using the Bayes linear adjustment formulae,

$$\begin{aligned} \mathsf{E}_z(y) &= \mathsf{E}(y) + \mathsf{Cov}(y, z)\,\mathsf{Var}(z)^{-1}\left\{\tilde{z} - \mathsf{E}(z)\right\}, \\ \mathsf{Var}_z(y) &= \mathsf{Var}(y) - \mathsf{Cov}(y, z)\,\mathsf{Var}(z)^{-1}\,\mathsf{Cov}(z, y)\,. \end{aligned} \tag{9}$$

The only expensive calculation in (9) is the inversion of $\mathsf{Var}(z)$, which is dominated by the Choleski decomposition of the variance matrix (see, e.g., Golub and Van Loan, 1983).

The Bayes linear approach may be viewed either as a pragmatic approximation to a full Bayes analysis or as an appropriate analysis when we have only made prior specifications for means, variances and covariances, in a formalism where expectation rather than probability is the primitive expression of belief; Goldstein (1999) discusses the fundamental features of this approach and addresses the foundational justification for such procedures under partial prior specification. Thus each variance expression corresponds to a direct specification for the expected squared distance between the uncertain quantity and its expectation, rather than as in the fully-probabilistic case, where this variance is often considered to be a true but unknown parameter about which we specify prior beliefs; for a discussion of how data may be used to adjust the assessment of variances, see Goldstein and Wilkinson (1996).

The Bayes linear approach is tractable because there is no explicit learning about x^* using z. This allows us to integrate x^* out to produce a prior variance structure for f^*, and then to introduce z, so that the Bayes linear approach provides a prediction but not a calibration. We may assess if this approach is good enough for the problem at hand, by evaluating the adjusted variances for the quantities we are most concerned to predict. When the adjusted variances are too large, then we may seek to extract more information from the combination of simulator evaluations and observed data, while retaining the essential tractability of the Bayes linear approach. We now describe a form of Bayes linear *calibrated* prediction for this purpose.

4 The 'hat run'

4.1 An illustrative example

To motivate our development of Bayes linear calibrated prediction, we identify by example the general conditions under which we may improve upon the

Bayes linear predictive approach in section 3. Suppose that the simulator f has, say, three output components $f_1(x)$, $f_2(x)$ and $f_3(x)$, where x is a scalar input. We have observations z_1 and z_2 corresponding to f_1 and f_2, and we wish to predict the system value y_3 corresponding to f_3.

Suppose first that we may, to a good approximation, write our emulator for each f_i as

$$f_i(x) \approx a_i + b_i\, x + c_i\, x^2, \qquad i = 1, 2, 3. \tag{10}$$

Provided we may make enough evaluations of the simulator to identify the coefficients to a good approximation, we can invert the first two outputs to find x and x^2 as linear expressions in $f_1(x)$ and $f_2(x)$, and write

$$f_3(x^*) \approx \beta_0 + \beta_1\, f_1(x^*) + \beta_2\, f_2(x^*), \tag{11}$$

for computable values of β_0, β_1 and β_2. This allows us to make a good linear estimate of y_3 in terms of (z_1, z_2), without first having to identify x^*.

In practice,we might need to expand each $f_i(x)$ in higher order functions of the vector of inputs in order to get a good approximation. However, also, we often match to large quantities of system data, and the above argument will still hold whenever there is enough data to generate, at least approximately, the corresponding linear inversion between the components of f. In these situations the Bayes linear approach will work well, and we would not expect a fully-Bayesian treatment to improve our prediction very much.

Now suppose, in our three-component example, that our emulator for each component is

$$f_i(x) = a_i + b_i\, x + c_i x^2 + u_i(x) \qquad i = 1, 2, 3 \tag{12}$$

where $u_i(\cdot)$ is treated as a stochastic process with average variance σ_i^2. If the σ_i^2 values are small then this does not much affect the previous argument. However, if σ_3^2 is large and u_3 is only weakly correlated with u_1 and u_2 then the local behaviour of $f_3(x)$ at $x = x^*$ can only be weakly determined by the global fitting of the emulator based on evaluations for x spanning the input space $\mathcal{X}$. In this case, the fully-Bayesian approach has a potential advantage over the Bayes linear approach, as follows. By explicitly constructing a posterior distribution for x^* given the simulator evaluations and the system data, we may be able to make a reasonable Bayesian estimate, $\hat{x}$ say, for x^*. This calibration step identifies a particular simulator evaluation $f(\hat{x})$, which is likely to be strongly informative for $f_3(x^*)$ by resolving much of the uncertainty about $u_3(x^*)$. Further, because the posterior distribution for x^* is concentrated around $\hat{x}$, the value for $f(\hat{x})$ will be an important element in the prediction of y_3.

4.2 The 'hat run' in practice

The Bayes linear approach is tractable for large problems precisely because it sidesteps the calibration step. However, as we shall now describe, it is possible to put this calibration back into the Bayes linear approach, without sacrificing its tractability. We expect this to be important whenever the local behaviour of the simulator for any general value of x cannot be well-determined from the evaluations $(F; X)$. Our procedure is as follows.

First stage. We compute the Bayes linear estimate of x^* using z, denoted $\hat{x}$. We therefore define

$$\hat{x} \triangleq \mathsf{E}_z(x^*) \equiv v + Wz \tag{13a}$$

where, from (9),

$$W \triangleq \mathsf{Cov}(x^*, z)\,\mathsf{Var}(z)^{-1} \quad \text{and} \quad v \triangleq \mathsf{E}(x^*) - W\mathsf{E}(z)\,. \tag{13b}$$

The mean and variance of z are computed from (7) and (8). As $x^* \perp\!\!\!\perp f$, the covariance between x^* and z is

$$\mathsf{Cov}(x^*, z) = \mathsf{Cov}\big(x^*, Hf(x^*)\big) = \mathsf{Cov}\big(x^*, \mu(x^*)\big)\,H^T \tag{14}$$

where $\mu(\cdot)$ is the mean function from the emulator.

Second stage. We evaluate f at $\hat{x}$. We refer to this evaluation as the 'hat run', denoted $\hat{f} \triangleq f(\hat{x})$.

Third stage. We evaluate the mean and variance of $\hat{f}$ and the covariance between $\hat{f}$ and (y, z). We do this by expressing $\hat{f}$ in terms of the primitive uncertain quantities (f, x^*, ϵ, e), and the known quantities (H, v, W). This allows us to compute the mean and variance of $\hat{f}$ in terms of our beliefs about f, our prior distribution for x^*, and the means and variances of ϵ and e. As y and z are also made up of the same uncertain quantities, we can also compute the covariance of $\hat{f}$ with (y, z). In terms of the primitive quantities,

$$\begin{aligned}
\hat{f} \triangleq f\big(\hat{x}\big) &\equiv f\big(v + Wz\big) \\
&\equiv f\big(v + W\big[Hy + e\big]\big) \\
&\equiv f\big(v + W\big[H\{f(x^*) + \epsilon\} + e\big]\big).
\end{aligned} \tag{15}$$

How easy it is to evaluate this expression depends on the structure of the emulator for f. In particular, if the emulator has an explicit functional representation, then we can can substitute for the occurrences of f. For example,

consider the commonly-used linear form in which the emulator is written as

$$f(x) = a + Ax + u(x) \tag{16}$$

where (a, A) is a set of unknown coefficients and $u(\cdot)$ is a stochastic process over $\mathcal{X}$. In this case the final expression for $\hat{f}$ is

$$\hat{f} = a + A\big(v + W\big[H\{a + Ax^* + u(x^*) + \epsilon\} + e\big]\big) + u(\hat{x}), \tag{17}$$

where the $\hat{x}$ in $u(\hat{x})$ can also be expanded out, if necessary. The mean and variance of $\hat{f}$ can therefore be exactly inferred from the joint distribution of $\big(a, A, u(\cdot), x^*, \epsilon, e\big)$, and well-approximated from the low-order moments of this collection, if we use a moment-based approximation for $u(\cdot)$. The covariance of $\hat{f}$ and (y, z) can be inferred in the same way. The Appendix describes the general calculations for this linear form. For more complicated emulators that include non-linear and interaction terms in x, the calculations are essentially the same, although in this case a symbolic computation step may be helpful to multiply out and group like terms together.

Final stage. We now evaluate the adjusted mean and variance for y using both z and $\hat{f}$ as

$$\begin{aligned} \mathsf{E}_{z,\hat{f}}(y) &= \mathsf{E}(y) + \mathsf{Cov}\big(y, (z, \hat{f})\big)\, \mathsf{Var}\big(z, \hat{f}\big)^{-1} \big\{(\tilde{z}, \tilde{f}) - \mathsf{E}(z, \hat{f})\big\} \\ \mathsf{Var}_{z,\hat{f}}(y) &= \mathsf{Var}(y) - \mathsf{Cov}\big(y, (z, \hat{f})\big)\, \mathsf{Var}\big(z, \hat{f}\big)^{-1} \mathsf{Cov}\big((z, \hat{f}), y\big)\,, \end{aligned} \tag{18}$$

where $\tilde{f}$ is the observed value of $\hat{f}$. The mean $\mathsf{E}_{z,\hat{f}}(y)$ is a prediction for y that combines the global information contained in z with the local information provided by $\hat{f}$. The variance $\mathsf{Var}_{z,\hat{f}}(y)$ expresses the accuracy of the prediction; as we have explicitly accounted for the dependence of $\hat{f}$ on z, this assessment

does not count the system data twice.

We may determine whether or not to perform the hat run by evaluating $\mathsf{Var}_{z,\hat{f}}(y) - \mathsf{Var}_z(y)$, as this does not involve the actual value of the hat run (i.e. $\tilde{f}$). This reveals the potential for reducing our uncertainty by using the Bayes linear calibrated prediction.

Informally, we expect that the hat run will be informative for y when the following two conditions are satisfied. Firstly, we require that $\hat{x}$ is a reasonable estimate for those components of x^* which are important in determining the components of y that we are most concerned to predict accurately. We may judge this by the magnitude of the adjusted variance for x^* using z, namely $\mathsf{Var}_z(x^*) \equiv \mathsf{Var}(x^*) - W\,\mathsf{Var}(z)\,W^T$. Secondly, we expect to gain from evaluation of $\hat{f}$ in those circumstances when the global fitting of the function leaves a large amount of residual local uncertainty for important components of $f(x)$. This can be assessed through the size of the contribution of the residual to the emulator variance at $\hat{x}$.

4.3 More than one hat run

It is to possible perform and incorporate several hat runs at different stages of the analysis. The 'non-hat' runs of the simulator are used to adjust beliefs about the simulator. In our example they modify the mean and variance of the regression coefficients (a, A) in (16), and the mean and variance of the residual function $u(\cdot)$. At the point where a hat run is contemplated, the best choice for x^* is a function of current beliefs about the simulator, through the two quantities v and W in (13), which depend on the evaluations $(F; X)$. If we then do more 'non-hat' runs of the simulator, we further refine our beliefs about the simulator, and this will give us a new hat-run with different v and W to the original, v' and W' say, based on an expanded set of evaluations,

$(F'; X')$ say. It may happen that the new value $\hat{x}'$ is very different from the original value $\hat{x}$. In this case we would want to evaluate the simulator at this new value, and incorporate the result into our inference. We can compute the covariance of this new hat run with x^*, but we can also compute the covariance of the new hat run with the old one, remembering that $\hat{x}$ and $\hat{x}'$ are two *different* linear functions of z. Thus we will adjust our beliefs about y using the observed data z and the outcomes of both hat runs. Naturally, we can extend this process to any number of hat runs. All that is required is that, each time we evaluate the simulator at a choice of inputs which is a (linear) function of the data, we must evaluate the full covariance structure for this run and all preceding hat runs, using an expansion such as (15), as applied to the current adjusted beliefs summarised by the emulator of our simulator.

The precise way in which more than one hat run might be used will depend on the application. For example, an early hat run might be intended primarily for diagnostic purposes. At this point the emulator would be quite uncertain, and a further collection of non-hat runs could materially change our emulator and, consequently, change the location of $\hat{x}$ as well. When we have carried out sufficient simulator evaluations to reduce substantially uncertainty about the regression coefficients in the emulator, then, rather than perform a single hat run at $\hat{x}$, we may choose to perform a collection of evaluations around $\hat{x}$, possibly aligned according to the principal components of $\mathsf{Var}_z(x^*)$. We would assess the benefit of this strategy, for example to optimise the number and location of points in the collection, in terms of its impact on predictive uncertainty. In practice, with more than one hat run the calculations to compute the joint mean and variance, though well-defined, would be intricate, and would best be implemented in a computer algebra package. The expense of this extra step has to be seen in the context of the cost of creating the simulator in the

first place. Large simulators cost thousands of man-hours to construct, and it does not seem unreasonable, in such cases, to expend a proportion of that effort in improving their inferential capabilities.

4.4 Diagnostics

Once we have performed the hat run, we may use the observed outcome of $\hat{f}$, which we have denoted $\tilde{f}$, in a 'whole-system' diagnostic. Define

$$d \triangleq z - H\hat{f}, \tag{19}$$

the vector of deviations between the system data and the appropriate transformation of the hat run. This vector of deviations is similar to those that arise in the traditional practice of tuning the simulator to optimise the match between simulator output and system data. A difficulty with the traditional tuning process is that there is no metric to express what deviation is acceptable given issues such as the inadequacy of the simulator, as measured by $\mathsf{Var}(\epsilon)$, measurement error, $\mathsf{Var}(e)$, and, possibly, non-uniform beliefs about the Best Input x^*. In particular, any Bayesian attempt to quantify the metric probabilistically runs into the difficulty that the data z have been used to calibrate the simulator. Using the hat-run approach, however, we can compute the mean and variance of d from the prior mean and variance of the collection $(z, \hat{f})$. This prediction for the mean and variance of d is based entirely on our assessment of primitive quantities such as x^* and ϵ, and not on the actual observed values of either z or $\hat{f}$; this makes it a *prior* prediction for d. Therefore by comparing the actual observed value for d, namely $\tilde{z} - H\tilde{f}$, with its prior prediction we can assess the quality of our judgements regarding these primitive quantities.

Because our assessments rely on our prior covariance specification, we emphasise the importance of the diagnostic evaluation of d before prediction. For example, if we are concerned about the spatio-temporal structure of the simulator output, then it would be natural to examine a space- and time-indexed plot of the components of d. Problems in this plot will lead us to a re-appraisal of our statistical modelling, for example by changing the variance structure of the discrepancy ϵ, or by dropping outlying, typically long-past, components of the output vector.

In our application this diagnostic proves to be valuable, indicating a problem with our original assessment of the various uncertainties. We discuss this further in section 5.5.

5 Example: Thermohaline circulation collapse

We illustrate our approach with a simple example, comprising a computer simulator with five inputs, eight outputs, thirty evaluations and six system data. To build an emulator, compute the full covariance structure of the collection $(y, z, \hat{f})$, and then compute the adjusted mean and variance for y takes about 1 minute on a standard desktop computer, using code written for the R statistical computing environment (R Development Core Team, 2004). More details of the calculation are given in section 5.7.

5.1 A model of the Atlantic

Zickfeld et al. (2004), hereafter ZSR, develop a compartmental model of the Atlantic designed to manifest Thermohaline Circulation (THC) shutdown. This model is shown schematically in Figure 1. The state vector comprises temperature T_i and salinity S_i for each of the four compartments. The key quantity is the rate of meridional overturning, m (units 'Sv', a Sverdrup is $10^6\,\mathrm{m}^3\,\mathrm{s}^{-1}$),

which is driven by temperature and salinity differences between compartments 1 and 2. The model comprises an expression for m, and an Ordinary Differential Equation (ODE) system describing the conservation of temperature and salinity in the presence of forcing from $(T_1^*, T_2^*, T_3^*, F_1, F_2)$.

The five model parameters that are taken to be uncertain are the three relaxation temperatures, T_1^*, T_2^* and T_3^*, the thermal coupling constant Γ, which affects the reversion of compartment temperature towards relaxation temperature, and the empirical flow constant k, which scales the relation between m and the temperature and salinity differentials. The three relaxation temperatures should satisfy the ordering $T_2^* < T_1^* < T_3^*$, and so the set of five input variables are re-parameterised as

$$x \triangleq \left(T_2^*, T_1^* - T_2^*, T_3^* - T_1^*, \Gamma, k\right) \in \mathcal{X} \subset \mathbb{R}_+^5. \tag{20}$$

The ZSR model is used to investigate the response of the equilibrium value of overturning, m^{eq}, to different amounts of freshwater forcing, F_1. For any given set of parameter values we can plot m^{eq} against the value for F_1. Typically, low values of F_1 are associated with positive values for m^{eq}. As F_1 is increased there comes a point at which m^{eq} goes negative. This value is the critical value F_1^{crit}: it shows where the THC shuts down for that particular parameter set. In our treatment, the model parameters are constrained to a region where F_1^{crit} is always well-defined, and the analysis below should be treated as conditional on this fact. The climate parameter corresponding to F_1^{crit} is a key determinant of the impact of global warming on the Western European climate, since THC shutdown could lower the temperature in the North-Eastern Atlantic by several degrees Centigrade. Therefore our interest is in predicting the value of F_1^{crit} for the Atlantic, using data on the current state of the Atlantic for calibration

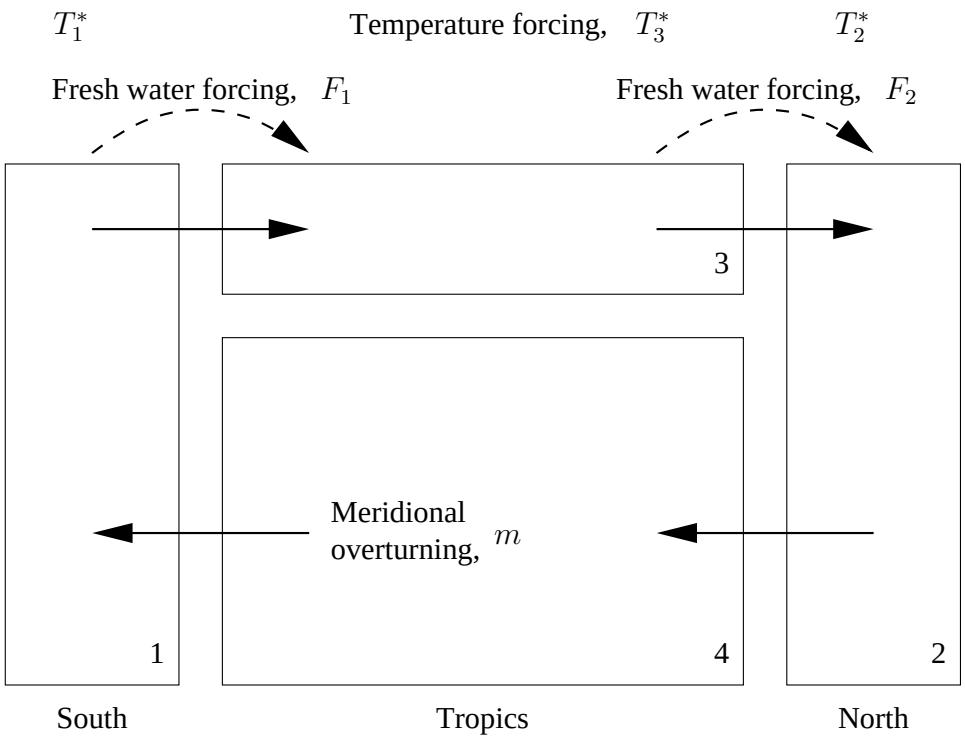


Figure 1: A compartmental model of the Atlantic, as described in Zickfeld et al. (2004).

purposes. There are eight model outputs for this experiment: three equilibrium temperatures and three equilibrium salinities (excluding compartment 4 in each case) and equilibrium overturning, all of these taken at what is believed to be the current value $F_1 = 0.014\,\mathrm{Sv}$, plus the critical value F_1^{crit}. This particular treatment of the ZSR model is described in more detail in Goldstein and Rougier (2005b).

5.2 Building an emulator

Our emulator has the linear form

$$f(x) = a + Ax + u(x) \tag{21}$$

where a and A are an 8-vector and (8×5)-matrix, respectively, $\mathcal{X} = [0,1]^5$ after a linear mapping (see the lower and upper limits in Table 2), and $u(x)$ is an 8-dimensional residual. Adopting a Bayes linear approach, our initial knowledge about the simulator is represented by the mean and variance of the collection $\big(a, A, u(\cdot)\big)$, and updating by $(F; X)$ has the effect of modifying the mean and variance of this collection. Our prior for $u(x)$ is a weakly stationary random field independent of (a, A); this is a reasonable simplification because in our application the linear terms typically explain a large amount of the variation in $f(x)$.

For the prior covariance structure of $u(x)$ in the emulator we choose a product structure with a stationary and isotropic 'gaussian' correlation function:

$$\mathsf{Cov}\big(u(x), u(x')\big) = \rho(x, x')\,\Sigma^u \tag{22a}$$

$$\text{where } \rho(x, x') \triangleq \prod_{j=1}^{5} \exp\left\{ -\left| \frac{x_j - x'_j}{\theta} \right|^2 \right\}. \tag{22b}$$

Output	(Int.)	x_1	x_2	x_3	x_4	x_5	$\sqrt{\Sigma^u_{ii}}$	R^2 (%)
$f_1 = T_1^{\text{eq}}$	0.15	9.99	4.61	0.09	−0.05	−0.04	0.09	99.97
$f_2 = T_2^{\text{eq}}$	−0.68	10.14	1.79	0.40	−0.50	0.91	0.37	99.30
$f_3 = T_3^{\text{eq}}$	0.05	9.98	4.86	9.87	0.12	−0.15	0.06	99.99
$f_4 = S_1^{\text{eq}}$	35.08	−0.01	−0.11	0.08	−0.04	−0.08	0.09	37.06
$f_5 = S_2^{\text{eq}}$	34.59	0.08	0.23	−0.20	0.07	0.33	0.13	69.84
$f_6 = S_3^{\text{eq}}$	35.10	−0.04	0.06	−0.02	0.02	−0.06	0.12	9.64
$f_7 = m^{\text{eq}}$	−23.57	1.04	35.83	−2.21	14.28	17.98	5.21	91.80
$f_8 = F_1^{\text{crit}}$	−0.08	−0.00	0.23	−0.01	0.05	0.07	0.03	89.90

Table 1: Expected value of regression coefficients from the emulator; residual standard deviation; and R^2. The latter two values are prior to updating the residual using $(F; X)$.

We set $\theta = \log(5)^{-0.5} \approx 0.788$, corresponding to $\mathsf{Corr}\big(u_i(0), u_i(1)\big) = 0.2$. This value for θ gives sample paths for $u_i(x)$ that appear moderately non-linear, as $u(x)$ should represent the low-order excluded effects in (21). The role of θ in the inference is discussed in more detail in section 5.6.

Our design matrix X comprises just 30 evaluations of our simulator, reflecting the type of budgetary constraints that exist in large computer experiments. These 30 evaluations were generated from a latin hypercube selected to have low correlations (less than 0.2 in absolute size) and large interpoint distances (greater than 0.25). We estimate the parameters of the emulator using multivariate least squares, and we then update the residual field, ensuring that the mean function of the emulator interpolates the evaluations; further details are given in the Appendix. The resulting expected values for the coefficients (a, A) are given in Table 1, along with the prior residual standard deviation and the R^2 values. The first three outputs (equilibrium temperatures for the surface compartments) are very strongly determined by the three relaxation temperatures, but the remaining five outputs have residuals which explain substantial amounts of variation.

5.3 Linking the model, the system and the system data

The model, being highly aggregated, does not map easily into measurements that can be made on the system itself, i.e. temperature and salinity measurements taken from the Atlantic. ZSR's intention is to calibrate their model to a superior model, CLIMBER-2, which has been carefully tuned to the Atlantic in a separate experiment. Therefore the 'system' in this case is the output from CLIMBER-2, which we think of as a sophisticated measuring device. The output from CLIMBER-2 is measured without error, so that $\mathsf{Var}(e) = \mathbf{0}$. However, there is some ambiguity about the precise values for compartment 1 (ZSR, Table 3), which we must include in the variance of the model discrepancy, ϵ. We choose to use a diagonal variance matrix for ϵ; the non-zero values are given in Table 3. The variance of ϵ is set at one hundredth of that in Goldstein and Rougier (2005b), as it was strongly suggested by the whole-system diagnostic d, given in (19), that our original values were inconsistent with our other modelling choices and the data. This is discussed in more detail in section 5.5.

For consistency with ZSR, we take the Best Input to have independent and uniform components within given ranges; these are shown in Table 2. The CLIMBER-2 measurements comprise temperatures $T_1^{\text{eq}} = 6.0\,^\circ\text{C}$, $T_2^{\text{eq}} = 4.7\,^\circ\text{C}$ and $T_3^{\text{eq}} = 11.4\,^\circ\text{C}$, salinity differences $S_2^{\text{eq}} - S_1^{\text{eq}} = -0.15\,\text{psu}$ and $S_3^{\text{eq}} - S_2^{\text{eq}} = 0.1\,\text{psu}$ and overturning $m^{\text{eq}} = 22.6\,\text{Sv}$ (ZSR, Table 3).

5.4 Prior to evaluating the 'hat run'

After the 30 evaluations used to build the emulator, we may assess the benefit from evaluating the simulator at the hat run input $\hat{x}$. Table 2 shows the value of the hat run input, and the reduction in uncertainty about x^* that follows from adjusting beliefs about x^* by the data z. The data are highly informative about the best value of first three inputs, moderately informative

Simulator inputs		Range for x^*		Expressed on $[0,1]$	
		Lower	Upper	$\hat{x} \triangleq \mathsf{E}_z(x^*)$	$\mathsf{Sd}_z(x^*)$
$x_1 = T_2^*$		0	10	0.350	0.046
$x_2 = T_1^* - T_2^*$		0	5	0.537	0.107
$x_3 = T_3^* - T_1^*$		0	10	0.538	0.031
$x_4 = \Gamma$		10	70	0.579	0.277
$x_5 = k$		5000	100000	0.810	0.173

Table 2: Quantities associated with the simulator inputs. Range for the uniform prior on x^*; mean and standard deviation of x^* adjusted by $z = \tilde{z}$, expressed on $[0,1]$. The original mean and standard deviation for each component of x^* on $[0,1]$ are 0.500 and 0.289, respectively.

System values	$\mathsf{Sd}(\epsilon)$	$\mathsf{Sd}(f^*)$	Evaluated at $x = \hat{x}$		$\mathsf{Sd}\big(\hat{f}\big)$
			$\mathsf{Sd}\big(a + Ax\big)$	$\mathsf{Sd}\big(u(x)\big)$	
y_1	0.206	3.189	0.027	0.017	3.180
y_2	0.200	2.990	0.112	0.073	2.981
y_3	0.200	4.288	0.017	0.011	4.285
y_4	0.013	0.089	0.028	0.018	0.080
y_5	0.010	0.165	0.039	0.025	0.161
y_6	0.010	0.106	0.038	0.025	0.087
y_7	1.000	13.332	1.591	1.033	13.349
y_8	0.005	0.077	0.010	0.007	0.075

Table 3: Quantities associated with the system values. Standard deviation of the discrepancy; standard deviation of $f^* \triangleq f(x^*)$; standard deviation of the emulator at $x = \hat{x}$, broken down into contributions from $a + Ax$ and $u(x)$; standard deviation of the hat run $\hat{f} \triangleq f(\hat{x})$.

	A	B	C	D_1	D_2	E_1	E_2
Std dev., 10^{-2} Sv	7.73	2.31	1.60	2.27	2.26	2.03	1.81
Rel. to $\mathsf{Sd}_z(y_8)$, col. B			-30%	-2%	-2%	-12%	-22%

Table 4: Predictive uncertainties for $y_8 \triangleq F_1^{\text{crit}}$, after 30 evaluations of the simulator. A, prior to adjustment by z; B, after adjustment by z; C, anticipated for adjustment by $(z, \hat{f})$; D_1, anticipated after performing one additional evaluation at the point of maximum emulator uncertainty; D_2, anticipated after performing one additional evaluation at the best point for reducing predictive uncertainty; E_1, anticipated after one order-of-magnitude more evaluations; E_2, anticipated after 2 orders-of-magnitude more evaluations.

about the last, and not informative at all about x_4^*. Overall, the large reduction in uncertainty about x^* that follows from adjusting by z suggests that the hat run will be effective in reducing our uncertainty about y, as discussed in section 4. Referring to Table 3, the residual makes quite a large contribution to uncertainty about $f(x)$ at $x = \hat{x}$; this also suggests that the hat-run will be effective. Table 3 also shows that $\mathsf{Var}(f^*)$ and $\mathsf{Var}(\hat{f})$ are very similar for the first three outputs. This arises because for these outputs the emulator regression coefficients are effectively known, the residual is effectively zero, and the contribution of the discrepancy is small relative to the uncertainty in y that is engendered by uncertainty in x^*.

For simplicity, we concentrate on predicting $y_8 \triangleq F_1^{\text{crit}}$, as this is the quantity that most clearly quantifies the imminence of THC collapse, although our approach allows us to compute a predictive mean and variance for the full collection of system values. Our prior uncertainty about y_8 is shown in column A of Table 4, while our adjusted uncertainty using the system data z in the Bayes linear approach is shown in column B. The Bayes linear approach has reduced our uncertainty about y_8 by 70%. Our interest is how much we can improve on this reduction using a single hat run evaluation, and how this reduction compares with that of other types of evaluation of the simulator.

The effect of the hat run is shown in column C of Table 4. The hat run contributes a further reduction of 30% in our uncertainty about y_8. This seems substantial for a single additional evaluation, bearing in mind that 30 evaluations have already occurred, and bears out the analysis of Tables 1, 2 and 3, where we see that x_2 is the most important input for f_8, that we learn quite a lot about x_2^* from the calibration calculation, and that the emulator residual contributes about half the uncertainty about $f_8(x)$ at $x = \hat{x}$.

It is interesting to compare this reduction with what might be achieved using one or more additional simulator evaluations. First, we try two criteria for selecting a single additional design point. One simple approach is to evaluate the simulator at the point $x \in \mathcal{X}$ at which $\mathsf{Var}\big(f_8(x)\big)$ is maximised. For our evaluations this point is $(1, 1, 1, 1, 0)$: an extreme point is the likely outcome here, given that the design points in X tend not to occupy the corners. We can use the mean function of the emulator to generate pseudo-data for this evaluation, and examine the outcome of adding this evaluation to our set of 30. The result is shown in Table 4, column D_1: there is a 2% reduction in uncertainty. Note that this calculation has not involved the value $\tilde{z}$ in choosing the next evaluation point, and so there is no concern here about double-counting the system data, of the kind we have been careful to account for in our use of $\hat{f}$.

Perhaps a better alternative is the one-step sequential design approach suggested in Craig et al. (2001, section 8), which selects the input value $x \in X$ for which the adjusted variance $\mathsf{Var}_z\big(y_8\big)$ is minimised. For our evaluations this point is $(1, 0, 0, 1, 0.062)$: again, an extreme point is the likely outcome, given that our emulator has a strong linear component. The result is shown in Table 4, column D_2: there is a slightly greater reduction in uncertainty with this alternative, but still about 2%. Again, the value $\tilde{z}$ does not affect the adjusted variance $\mathsf{Var}_z(y)$, so there is no double-counting the system data.

Both of these approaches to selecting an additional evaluation result in a reduction in uncertainty of about 2%, which is much lower than using the additional evaluation in the hat-run approach (30%). The Craig et al. approach is the more effective of the two, although it costs slightly more to implement. However, neither approach can gain the benefit of $\hat{f}$ in resolving local uncertainty in the region of $\mathcal{X}$ most likely to contain x^*. Note that we can also infer that the benefit of including the hat run as the 31st evaluation, without giving it any special treatment, will give a reduction of uncertainty of no greater than 2%; in fact, in this case the standard deviation of y_8 is 2.29×10^{-2} Sv, for a reduction of 0.53%. This illustrates that it is not simply the location of the hat run which is the key to the uncertainty reduction: it is the special treatment of the hat run which takes account of the way it links up with the other uncertain quantities.

Second, we can consider two further calculations to get upper bounds on the amount that uncertainty about y_8 might be reduced by many more simulator evaluations, if we do not carry out the hat run analysis. With an order-of-magnitude more evaluations (say, 100) we might consider that $\mathsf{Var}(Ax) \approx \mathbf{0}$ for all x, in our emulator, while with two orders-of-magnitude (say, 1000) we might also have $\mathsf{Var}\big(u(x)\big) \approx \mathbf{0}$, in our emulator (at this point we are treating the simulator as effectively known). The anticipated effect of these two limits is shown in Table 4, columns E_1 and E_2. In the first case we achieve a 12% reduction in uncertainty; in the second case a 22% reduction in uncertainty. Seen in this context, a single hat run seems to offer extremely good value in terms of reducing our uncertainty about y_8. The superior performance of the hat run in this particular application arises because the reduction in our uncertainty about x^* that it incorporates is more valuable than a reduction in uncertainty about the simulator, even when the latter is taken to the point

where the simulator may be treated as known. However, the comparison is purely for illustration. If we were really able to reduce simulator uncertainty to zero, then we could carry out the hat run analysis, in addition, at no extra cost in simulator time.

5.5 The result of evaluating the hat run

Once we evaluate the hat run we can update our prediction for y_8 for which we now have a new mean value. The Bayes linear prediction after 30 evaluations is $y_8 = 0.148 \pm 0.023\,\mathrm{Sv}$ (mean $\pm$ standard deviation); after performing the hat run evaluation it is $0.132 \pm 0.016\,\mathrm{Sv}$, so there is also a change in the updated mean value. (Also, for completeness, we evaluate the simulator at the two one-step choices for x and update our prediction. In both of these cases our prediction using the pseudo-data was accurate.)

We also have the opportunity to use the outcome of the hat run as a 'whole-system' diagnostic based on comparing the computed value of d from (19) with its prior mean and variance, as explained in section 4.4. As the prior mean of d is not zero and the prior variance shows some strong correlations, we present the diagnostic comparing the observed value of d with its mean and variance in terms of the components of the vector $Q^{-T}\big(d - \mathsf{E}(d)\big)$, where Q is the Choleski decomposition of $\mathsf{Var}(d)$; these standardised components have mean zero, variance one and correlation zero. The results are

$$1.104, -1.039, 0.042, -0.480, -1.420, -1.159$$

These do not suggest any cause for concern. If we were to calibrate the sum of squared values of these six quantities with a χ^2_6 distribution (making a gaussian approximation) the test statistic of 5.889 would be 56^{th} percentile.

This diagnostic result is in marked contrast to the outcome when we proceed using our original choice for $\mathsf{Var}(\epsilon)$, as given in Goldstein and Rougier (2005b), which is 100 times larger. With this larger value for $\mathsf{Var}(\epsilon)$ the standardised quantities from d are

$$0.815, -0.297, -0.297, -0.208, -0.293, -0.601$$

which seem rather small; the χ^2_6 test statistic is 1.330, which is only 3^{rd} percentile. There are many ways in which we might address this diagnostic warning. We have adopted the approach of scaling the discrepancy variance—a quantity about which we do not have strong views—for reasons of parsimony, and because by changing a single parameter we have obtained plausible outcomes for each of the scalar diagnostics. An alternative response might have been to scale differentially our uncertainty about components of ϵ, according to whether or not they corresponded to components of z. We investigated an extreme version of this, where we scaled all components bar ϵ_8, which we left at its original value of $\mathsf{Sd}(\epsilon_8) = 0.05\,\text{Sv}$. In this case the prediction for y_8 using z alone was 0.148 ± 0.055; using z and the hat run it was 0.132 ± 0.052. Uncertainty about y_8 is underpinned by uncertainty about ϵ_8, which is irreducible in our treatment, and consequently the potential for improvement by the hat run is limited in this case.

In our application the whole-system diagnostic d has been instrumental in identifying a problem with our previous choices, and allowing us to re-appraise our statistical modelling.

5.6 Sensitivity analysis for the correlation length

It is interesting to investigate the effect of changing θ, which controls the correlation length in the prior correlation function of $u(\cdot)$, given in (22). This is both because θ can be a hard parameter to set, and also to check our intuition about how the behaviour of $u(\cdot)$ should affect the inference. The outcome for predictive uncertainty about y_8 for various different values of θ is shown in Figure 2; for reference, the value $\theta \approx 0.788$ was used in the analysis of the previous subsections, and in Table 4.

The hat run will be effective when the correlation length of $u(\cdot)$ corresponds approximately to our range of uncertainty in x^* after adjusting by z. In this case an evaluation at $\hat{x}$ will pin down the emulator residual in a region that is likely to contain x^*. If the correlation length is shorter, then the hat run will not perform much better than the standard Bayes linear prediction, because the effect of the evaluation at $\hat{x}$ fails to reach the residual at x^*. On the other hand, if the correlation length is longer, the hat run will pin down the residual over a large part of $\mathcal{X}$, not just the region near to x^*; however a long correlation length would represent a residual that was dominated by a linear effect, which would be undesirable in our emulator where the linear effect should be captured by the regression terms, leaving the residual to account for non-linear effects.

This role for θ is demonstrated in Figure 2, where for low values of θ (short correlation length) the hat run uncertainty C is close to the original Bayes linear uncertainty B, while for high values it is 'bottoming out' at a level that incorporates uncertainty in the regression coefficients as well as uncertainty from the discrepancy. The convergence between E_1 and E_2 in Figure 2 arises because in our emulators we update the residual function, and the amount of uncertainty in the updated function decreases as the correlation length

increases. The emulator for E_2 has no residual uncertainty, and the effect of varying θ in this emulator is felt entirely in the updated mean for the residual function.

In practice we choose θ first, based on our judgement about our simulator and our choice of global regression terms in our emulator. Thus the good performance of the hat run in our example can be attributed to the fact that although our correlation length is quite short, (i) the system data z allow us to make quite an accurate appraisal of x^*, as measured by $\mathsf{Var}_z(x^*)$; and (ii) the residual comprises a large part of our simulator uncertainty around $\hat{x}$.

5.7 Details of the calculation

In general, the expensive calculations in the hat run approach involve finding the expectation of various non-linear functions of x with respect to the prior distribution of x^*, e.g. to find the mean and variance of the residual $u(x^*)$, and the covariance of x^* and $u(x^*)$, where $u(\cdot)$ is updated by $(F; X)$. As these functions tend to be smooth functions of x, and as our prior for x^* itself typically has quite a simple structure (e.g. independent components), they can often be well-approximated by a numerical integration. We use this approach because it is easy to program for a variety of different choices for the correlation structure of $u(\cdot)$, and because it allows us to scale the calculation according to our computing resources, simply by varying the number of points in our numerical integration. For our grid we use a simple 5-dimensional product of a 6-point gauss-legendre integration rule. For larger problems we can adopt a more sophisticated integration approach, using a space-filling grid and variance reduction techniques such as importance sampling with antithetic variables (see, e.g., Evans and Swartz, 2000). This will often remain a feasible calculation for quite large input spaces, because the simulator evaluations used

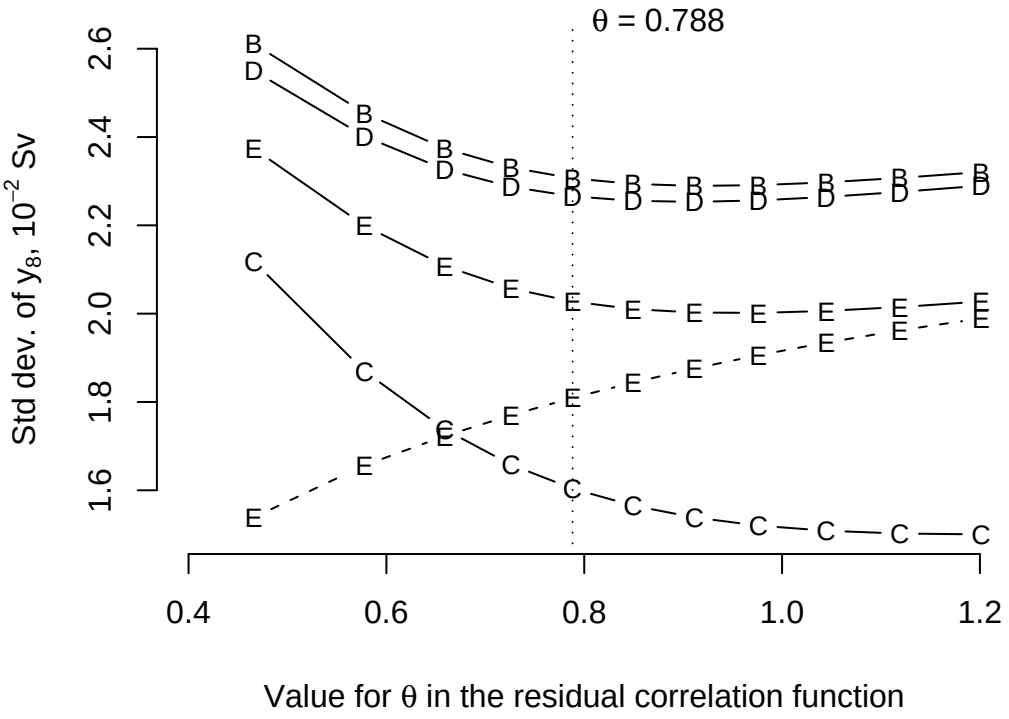


Figure 2: The effect of different values for θ in (22) on the reduction in uncertainty, using the same scenarios as in Table 4; D is D_1, E with the solid line is E_1, and E with the dashed line is E_2.

to update $u(\cdot)$ tend to become sparser in $\mathcal{X}$ as $\dim(\mathcal{X})$ increases, and this sparsity translates into smoothness for the updated mean function and variance function of $u(\cdot)$.

Where $\dim(\mathcal{X})$ becomes very large we have the option of replacing the numerical integrations with expansions based on low-order moments of x^*. Technically the limit to our accuracy is the number of moments we have for x^*. The downside of this approach is that it requires the explicit differentiation of the mean function and variance function of $u(\cdot)$, which makes the calculation dependent on the choice of correlation function for $u(\cdot)$, and consequently less flexible and more complicated to program.

6 Conclusion

The calibrated prediction approach that we have described provides a unified approach for (i) judging many aspects of the adequacy of our overall view of the computer simulator, the corresponding emulator, and the associated discrepancy between the simulator and the underlying system, by comparison of simulator evaluations and system data; and, (ii) using such modelling and observations for predicting system behaviour. This approach offers a powerful compromise between the Bayes linear approach to prediction, which is analytically straightforward but which does not allow us easily to exploit local information about the value of the simulator for inputs which appear to give good matches to historical observations, and the full Bayes approach which can better exploit such information but only at the price of an enormous, and for large problems infeasible, computational burden. In contrast, our Bayes linear calibrated predictions make efficient use of restricted aspects of the prior formulation and remain tractable even for large problems.

Acknowledgements

Jonathan Rougier has been partly funded by grants from the UK Natural Environment Research Council (NERC) and the Tyndall Centre for Climate Change Research. We would like to thank the Associate Editor and two Referees for their very perceptive comments on earlier versions of this paper. We would also like to thank Kirsten Zickfeld for her assistance with the model used in our illustration.

A Appendix: Detailed calculations

In the following calculations many terms will be exactly computable from knowledge of the first- and second-moments of the uncertain quantities. Other terms will be exactly computable using the distribution for the Best Input x^*, such as $\mathsf{E}\big(r(x^*)\big)$, where $r(x)$ is defined below in (A4c). As long as the variance attributable to the residual in the emulator is relatively small, we can approximate the latter terms using low-order gaussian quadrature rules with little loss of accuracy in our inference (where the dimension of the input space $\mathcal{X}$ is large, a computationally efficient alternative is to used moment-based approximations). Finally, there will be some terms for which we cannot make an exact calculation without imposing further distributional assumptions. Again, these concern the emulator residual and we use moment-based approximations. This Appendix gives a complete description of the choices that we make.

A.1 The emulator

Our starting point is the emulator

$$f(x) = Ax + u(x) \tag{A1}$$

where $f(x)$ is a k-vector of simulator outputs, x is a $(1+p)$-vector of a constant plus the model parameters, A is a $k \times (1+p)$ matrix of uncertain coefficients, and $u(\cdot)$, the emulator *residual*, is a vector-valued random field with zero mean, and a product covariance structure

$$\mathsf{Cov}\big(u(x), u(x')\big) = \rho(x, x')\,\Sigma^u \tag{A2}$$

for some given scalar covariance function $\rho(\cdot,\cdot)$. We estimate the mean and variance of A and the variance Σ^u using multivariate least squares, from a collection of evaluations $(F;X)$. For this to be appropriate the points in X should be widely separated so that $\mathsf{Corr}\big(u_i(X_m), u_i(X_{m'})\big) \ll 1$, where X_m is the m^{th} row of X (and likewise for F_m, below). Where this is not possible, perhaps because we are able to perform a large number of evaluations, a 'generalised' multivariate least squares is possible, although in this case it is important to ensure that the residual $u(\cdot)$ is orthogonal to the regressors, which constrains the structure of $\rho(\cdot,\cdot)$.

We can update $u(\cdot)$ using the observed values

$$\hat{U} \triangleq F - X\bar{A}^T, \tag{A3}$$

where $\bar{A}$ is the expected value of A. This is an approximation to the full update, because by ignoring the induced covariance between A and $u(\cdot)$ we cannot preserve the property that $f(X_m) = F_m$ with probability 1. With our approximate update we will have $\mathsf{E}\big(f(X_m)\big) = F_m$ and $\mathsf{Var}\big(f(X_m)\big)$ relatively small. The product structure in (A2) simplifies the update of the residual, as noted by O'Hagan (1998). The updated residual has the form

$$\mathsf{E}\big(u(x) \mid \hat{U}\big) = r(x) \tag{A4a}$$

$$\mathsf{Cov}\big(u(x), u(x') \mid \hat{U}\big) = r(x,x')\,\Sigma^u \tag{A4b}$$

where

$$r(x) \triangleq \hat{U}^T P\, r'(x) \tag{A4c}$$

$$P^{-1} \equiv \big\{\rho(X_m, X_{m'})\big\}_{m,m'=1}^{n} \tag{A4d}$$

$$r'(x) \triangleq \big\{\rho(X_m, x)\big\}_{m=1}^{n} \tag{A4e}$$

$$\text{and } r(x, x') \triangleq \rho(x, x') - r'(x)^T P\, r'(x'). \tag{A4f}$$

A.2 Mean and variance of $(\boldsymbol{y}, \boldsymbol{z})$

We link our simulator to the system itself and the system data using the Best Input approach, for which we have

$$y = f(x^*) + \epsilon \quad \text{and} \quad z = Hy + e \tag{A5}$$

where y is the system value corresponding to the simulator output, ϵ is the simulator discrepancy, z the available system data, H the incidence matrix mapping the system values to the system data, and e the measurement error. The two uncertain quantities ϵ and e are taken to be independent of each other and everything else, with zero means (assumed for simplicity) and given variances Σ^ϵ and Σ^e respectively.

We start by computing the mean and variance of $f(x^*)$, by conditioning on x^* and then integrating out. For the first step,

$$\mathsf{E}\big(f(x^*) \mid x^*\big) = \bar{A}x^* + r(x^*) \tag{A6a}$$

$$\mathsf{Var}\big(f(x^*) \mid x^*\big) = M(x^*) + r(x^*, x^*)\Sigma^u \tag{A6b}$$

where the expression $M(x) \triangleq \mathsf{Var}(Ax \mid x)$ involves the variance of A, technically a four-dimensional object, and is easily evaluated using a tensor approach

(see, e.g., McCullagh, 1987). There are further terms below, also denoted with M's, which need to be similarly treated. When we integrate out we get

$$\mathsf{E}\big(f(x^*)\big) = \bar{A}\,\mathsf{E}(x^*) + \mathsf{E}\big(r(x^*)\big) \tag{A7a}$$

$$\begin{aligned}\mathsf{Var}\big(f(x^*)\big) = {} & \mathsf{E}\big(M(x^*)\big) + \mathsf{E}\big(r(x^*, x^*)\big)\,\Sigma^u + \bar{A}\mathsf{Var}(x^*)\,\bar{A}^T \\ & + \bar{A}\,\mathsf{Cov}\big(x^*, r(x^*)\big) + \text{its transpose} \\ & + \mathsf{Var}\big(r(x^*)\big)\,. \end{aligned} \tag{A7b}$$

Now we can easily compute the mean and variance of the collection (y, z), using (A5), and then we can adjust our beliefs about the mean and variance of y using the observed values of the data z, using the Bayes linear adjustment formulae, as given in (9).

A.3 Mean and variance of $\hat{f}$

In this paper we extend the analysis to include an extra evaluation of the simulator, at a point chosen with reference to the system data. We define $\hat{x} \triangleq \mathsf{E}_z(x^*)$, and denote this the 'hat run' input value. This value (including the constant) can be written as a linear transformation of z, namely

$$\hat{x} \equiv v + Wz \tag{A8}$$

where $W \triangleq \mathsf{Cov}(x^*, z)\,\mathsf{Var}(z)^{-1}$ and $v \triangleq \mathsf{E}(x^*) - W\mathsf{E}(z)$. The calculation of the mean and variance of z is described above. The covariance is

$$\mathsf{Cov}(x^*, z) = \big[\mathsf{Var}(x^*)\,\bar{A}^T + \mathsf{Cov}\big(x^*, r(x^*)\big)\,\big]H^T \tag{A9}$$

using (14) and (A5).

Our interest is is computing the mean and variance of $\hat{f} \triangleq f(\hat{x})$, and the

covariance of $\hat{f}$ with (y, z). Once we have computed these terms we will have the mean and variance of the collection $(y, z, \hat{f})$, and we will be able to adjust our beliefs about the mean and variance of y using both the actual value of z and the outcome of evaluating f at $\hat{x}$, as given in (18).

We can find an explicit expression for $\hat{f}$ in terms of x^* by substituting $v + Wz$ for $\hat{x}$, then $H\big(f(x^*) + \epsilon\big) + e$ for z, then $Ax^* + u(x^*)$ for $f(x^*)$. The resulting expression for $\hat{f}$ after making these substitutions is

$$\hat{f} = A\big[v + G\big(A\,x^* + u(x^*) + \epsilon\big) + We\big] + u(\hat{x}) \tag{A10a}$$

where $G \triangleq WH$. It is helpful to write this as

$$\hat{f} \equiv A\,b(x^*) + AGA\,x^* + u(\hat{x}) \tag{A10b}$$

where $b(x) \triangleq v + G\big(u(x) + \epsilon\big) + We$ and $A \perp\!\!\!\perp b(x^*)$. Conditioning on x^* gives

$$\mathsf{E}(\hat{f} \mid x^*) = \bar{A}\big(v + Gr(x^*)\big) + M'x^* + \mathsf{E}\big(u(\hat{x}) \mid x^*\big)\,; \tag{A11a}$$

here $M' \triangleq \mathsf{E}(AGA)$, which can be evaluated using the first- and second-moments of A. The last term in (A11a) presents a problem, because we do not have an explicit distribution for $\hat{x} \mid x^*$. Rather than impose a distribution, we choose instead to approximate this term as

$$\mathsf{E}\big(u(\hat{x}) \mid x^*\big) = \mathsf{E}\big(r(\hat{x}) \mid x^*\big) \approx r(\hat{x}^*), \tag{A11b}$$

where

$$\hat{x}^* \triangleq \mathsf{E}(\hat{x} \mid x^*) = v + G\big(\bar{A}x^* + r(x^*)\big). \tag{A11c}$$

For the unconditional expectation we then have

$$\mathsf{E}(\hat{f}) \approx \bar{A}\big[v + G\,\mathsf{E}\big(r(x^*)\big)\,\big] + M'\,\mathsf{E}(x^*) + \mathsf{E}\big(r(\hat{x}^*)\big)\,. \tag{A12}$$

To find the unconditional variance we first compute the variance of the conditional expectation, which is

$$\begin{aligned}\mathsf{Var}\big(\mathsf{E}(\hat{f} \mid x^*)\big) \approx{}& M'\,\mathsf{Var}(x^*)\,(M')^T \\ &+ M'\,\mathsf{Cov}\big(x^*,\, \bar{A}Gr(x^*) + r(\hat{x}^*)\big) + \text{its transpose} \\ &+ \mathsf{Var}\big(\bar{A}Gr(x^*) + r(\hat{x}^*)\big)\,.\end{aligned} \tag{A13}$$

Next we need the conditional variance. For this we make the simplifying approximation that 'second-order' covariances can be neglected, i.e.

$$\mathsf{Cov}\big((Ax^*, \epsilon, e), u(\hat{x}) \mid x^*\big) \approx \mathbf{0} \tag{A14}$$

which allows us to drop many small terms in the outer-product of $\hat{f}$. Starting from (A10b) we can expand $\mathsf{Var}(\hat{f} \mid x^*)$ as the approximate sum of the following terms:

$$M^{(2)}(x^*) \triangleq \mathsf{Var}\big(A\,b(x^*) \mid x^*\big) \tag{A15a}$$

$$M^{(3)}(x^*) \triangleq \mathsf{Cov}\big(A\,\mathsf{E}\big(b(x^*) \mid x^*\big)\,, AGAx^* \mid x^*\big)\,, \text{ plus its transpose} \tag{A15b}$$

$$M^{(4)}(x^*) \triangleq \bar{A}\,\mathsf{Cov}\big(b(x^*), u(\hat{x}) \mid x^*\big)\,, \text{ plus its transpose} \tag{A15c}$$

$$M^{(5)}(x^*) \triangleq \mathsf{Var}(AGAx^* \mid x^*) \tag{A15d}$$

$$M^{(6)}(x^*) \triangleq \mathsf{Var}\big(u(\hat{x}) \mid x^*\big) \tag{A15e}$$

where $M^{(3)}$ and $M^{(4)}$ have both been simplified according to the general pattern $\mathsf{Cov}(ac, b) = \mathsf{E}(c)\ \mathsf{Cov}(a, b)$ where $(a, b) \perp\!\!\!\perp c$. To evaluate these terms we

require third- and fourth-moments for A. The simplest expedient, if we are unwilling to specify values, is to adopt the third- and fourth-moment structure of the gaussian distribution. Once the higher-moments are specified, $M^{(2)}$, $M^{(3)}$ and $M^{(5)}$ follow straightforwardly, with $M^{(2)}$ following the general pattern $\mathsf{Var}(bc) = \mathsf{E}(b)^2\ \mathsf{Var}(c) + \mathsf{Var}(b)\ \mathsf{E}(c^2)$ where $b \perp\!\!\!\perp c$.

For $M^{(4)}$ we have

$$\begin{aligned} M^{(4)}(x^*) &\approx \bar{A}G\,\mathsf{Cov}\big(u(x^*), u(\hat{x}) \mid x^*\big) \\ &= \mathsf{E}\big(r(x^*, \hat{x}) \mid x^*\big)\ \bar{A}G\Sigma^u \approx r(x^*, \hat{x}^*)\,\bar{A}G\Sigma^u \end{aligned} \tag{A16}$$

using the same approximations as before; we can treat $M^{(6)}$ in exactly the same manner, giving

$$M^{(6)}(x^*) \approx r(\hat{x}^*, \hat{x}^*)\,\Sigma^u. \tag{A17}$$

When we take the expectation of each of these terms over x^*, that of $M^{(5)}$ can be computed exactly from the first and second moments of x^*, while the other terms can be computed using quadrature. This gives us the expectation of the conditional variance.

Summing the variance of the conditional expectation and the expectation of the conditional variance then completes the calculation of the unconditional variance $\mathsf{Var}(\hat{f})$.

A.4 The two covariances

The calculation of $\mathsf{Cov}\big(\hat{f}, (y, z)\big)$ follows directly from (A5), along very similar lines to that of $\mathsf{Var}(\hat{f})$. To compute $\mathsf{Cov}(\hat{f}, y)$ we need

$$\begin{aligned} &\mathsf{Cov}\big(\mathsf{E}(\hat{f} \mid x^*)\,, \mathsf{E}(y \mid x^*)\big) \approx M'\,\mathsf{Var}(x^*)\ \bar{A}^T \\ &\quad + M'\,\mathsf{Cov}\big(x^*, r(x^*)\big) + \mathsf{Cov}\big(\bar{A}G\,r(x^*) + r(\hat{x}^*), \bar{A}x^* + r(x^*)\big)\,. \end{aligned} \tag{A18}$$

For the expectation of the conditional covariance we start with

$$\begin{aligned}\mathsf{Cov}(\hat{f}, y \mid x^*) \approx\ & \mathsf{Cov}\big(A\mathsf{E}\big(b(x^*) \mid x^*\big), Ax^* \mid x^*\big) \\ & + \bar{A}G\big[r(x^*, x^*)\Sigma^u + \Sigma^\epsilon\big] + \mathsf{Cov}(AGAx^*, Ax^* \mid x^*) \\ & + r(\hat{x}^*, x^*)\Sigma^u, \end{aligned} \tag{A19}$$

and then we integrate out x^*. For $\mathsf{Cov}(\hat{f}, z)$ we have simply

$$\mathsf{Cov}(\hat{f}, z) \equiv \mathsf{Cov}(\hat{f}, Hy + e) \approx \mathsf{Cov}(\hat{f}, y)\, H^T + \bar{A}W\Sigma^e, \tag{A20}$$

referring back to (A5) and (A10a).

References

Craig, P., M. Goldstein, J. Rougier, and A. Seheult: 2001, 'Bayesian Forecasting for Complex Systems Using Computer Simulators'. *Journal of the American Statistical Association* **96**, 717–729.

Craig, P., M. Goldstein, A. Seheult, and J. Smith: 1996, 'Bayes Linear Strategies for Matching Hydrocarbon Reservoir History'. In: J. Bernado, J. Berger, A. Dawid, and A. Smith (eds.): *Bayesian Statistics 5*. Oxford: Clarendon Press, pp. 69–95.

Craig, P., M. Goldstein, A. Seheult, and J. Smith: 1997, 'Pressure Matching for Hydrocarbon Reservoirs: A Case Study in the Use of Bayes Linear Strategies for Large Computer Experiments'. In: C. Gatsonis, J. Hodges, R. Kass, R. McCulloch, P. Rossi, and N. Singpurwalla (eds.): *Case Studies in Bayesian Statistics III*. New York: Springer-Verlag, pp. 37–87. With discussion.

Craig, P., M. Goldstein, A. Seheult, and J. Smith: 1998, 'Constructing Partial Prior Specifications for Models of Complex Physical Systems'. *The Statistician* **47**, 37–53. With discussion.

Currin, C., T. Mitchell, M. Morris, and D. Ylvisaker: 1991, 'Bayesian Prediction of Deterministic Functions, with Application to the Design and Analysis

of Computer Experiments'. *Journal of the American Statistical Association* **86**, 953–963.

Evans, M. and T. Swartz: 2000, *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford: Oxford University Press.

Goldstein, M.: 1999, 'Bayes Linear Analysis'. In: *Encyclopaedia of Statistical Sciences, update vol. 3*. London: John Wiley & Sons, pp. 29–34.

Goldstein, M. and J. Rougier: 2005a, 'Probabilistic Formulations for Transferring Inferences from Mathematical Models to Physical Systems'. *SIAM Journal on Scientific Computing* **26**(2), 467–487.

Goldstein, M. and J. Rougier: 2005b, 'Reified Bayesian Modelling and Inference for Physical Systems'. Submitted to the *Journal of Statistical Planning and Inference*, available at `http://www.maths.dur.ac.uk/stats/people/jcr/Reify.pdf`.

Goldstein, M. and D. Wilkinson: 1996, 'Bayes Linear Adjustment for Variance Matrices'. In: J. Bernardo, J. Berger, A. Dawid, and A. F. M. Smith (eds.): *Bayesian Statistics 5*. pp. 791–799, Oxford: Oxford University Press.

Golub, G. and C. Van Loan: 1983, *Matrix Computations*. Oxford, UK: North Oxford Academic.

Higdon, D., M. Kennedy, J. Cavendish, J. Cafeo, and R. D. Ryne: 2005, 'Combining Field Data and Computer Simulations for Calibration and Prediction'. *SIAM Journal on Scientific Computing* **26**(2), 448–466.

Houghton, J., Y. Ding, D. Griggs, M. Noguer, P. J. van de Linden, X. Dai, K. Maskell, and C. Johnson (eds.): 2001, *Climate Change 2001: The Scientific Basis. Contribution of Working Group 1 to the Third Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, UK: Cambridge University Press.

Kennedy, M. and A. O'Hagan: 2001, 'Bayesian Calibration of Computer Models'. *Journal of the Royal Statistical Society, Series B* **63**, 425–464. With discussion.

Koehler, J. and A. Owen: 1996, 'Computer Experiments'. In: S. Ghosh and C. Rao (eds.): *Handbook of Statistics, 13: Design and Analysis of Experiments*. pp. 261–308, North-Holland: Amsterdam.

McCullagh, P.: 1987, *Tensor Methods in Statistics*. London: Chapman and Hall.

Oakley, J. and A. O'Hagan: 2004, 'Probabilistic Sensitivity Analysis of Complex Models: a Bayesian Approach'. *Journal of the Royal Statistical Society, Series B* **66**, 751–769.

O'Hagan, A.: 1998, 'A Markov Property for Covariance Structures'. Available at `http://www.shef.ac.uk/~st1ao/ps/kron.ps`.

R Development Core Team: 2004, 'R: A Language and Environment for Statistical Computing'. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3, `http://www.R-project.org`.

Rougier, J.: 2005, 'Probabilistic Inference for Future Climate Using an Ensemble of Simulator Evaluations'. Submitted to *Climatic Change*, available at `http://www.maths.dur.ac.uk/stats/people/jcr/CCrevisionA4.pdf`.

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

Zickfeld, K., T. Slawig, and S. Rahmstorf: 2004, 'A Low-Order Model for the Response of the Atlantic Thermohaline circulation to Climate Change'. *Ocean Dynamics* **54**(1), 8–26.