SIAM J. Sci. Comput.
Vol. 26, No. 2, pp. 467–487

PROBABILISTIC FORMULATIONS FOR TRANSFERRING INFERENCES FROM MATHEMATICAL MODELS TO PHYSICAL SYSTEMS[*]

MICHAEL GOLDSTEIN[†] AND JONATHAN ROUGIER[†]

Abstract. We outline a probabilistic framework for linking mathematical models to the physical systems that they represent, taking account of all sources of uncertainty including model and simulator imperfections. This framework is a necessary precondition for making probabilistic statements about the system on the basis of evaluations of computer simulators. We distinguish simulators according to their quality and the nature of their inputs. Where necessary, we introduce further hypothetical simulators as modelling constructs to account for imperfections in the available simulators and to unify the available simulators with the underlying system.

Key words. direct simulator, indirect simulator, top simulator, measurable inputs, tuning inputs, Bayesian inference, calibration, history matching, calibrated prediction, uncertainty analysis

AMS subject classifications. 62F15, 62P35, 62M20, 62M40

DOI. 10.1137/S106482750342670X

1. Introduction.

In a computer experiment (sometimes referred to as an *in silico* experiment) we make inferences about a physical system using a computer simulator of that system. Such computer experiments may be used to investigate problems for which it would be difficult to carry out the corresponding physical experiments. Sometimes, the difficulty may be legal, ethical, or financial. But in other cases the experiment may not be possible, usually because the system is too small or too large. Thus in a computer experiment we can watch a single protein folding, we can watch the global climate evolving, or we can watch an entire galaxy coalescing.

We act as though evaluations of our models are informative about the physical system. For example, the debate on global climate change, which is already having a profound effect on policy, is guided by computer-based predictions of future climate, e.g., [9]. This is despite the problems that, in general, (i) our mathematical models of the underlying physical system are incomplete and often mutually inconsistent, (ii) the discretized solvers of these models (the simulators) are often woefully underresolved, and (iii) the simulators require inputs about which we are very uncertain. It is natural in these circumstances to require that model-based predictions about the system are accompanied by a careful evaluation of all sources of uncertainty. This is essential both for informed scientific debate, and also to assist policy makers within a decision-theoretic framework.

Our purpose in this paper is to construct a probabilistic framework that links one or more computer simulators and the underlying physical system, in order that the information available from evaluations of the simulators can feed through, via probabilistic conditioning, using a Bayesian statistical approach, into beliefs about the system. These beliefs include system properties and system behavior (e.g., system

[*] Received by the editors April 30, 2003; accepted for publication (in revised form) February 9, 2004; published electronically December 22, 2004.

http://www.siam.org/journals/sisc/26-2/42670.html

[†] Department of Mathematical Sciences, University of Durham, Durham DH1 3LE, UK (Michael.Goldstein@durham.ac.uk, J.C.Rougier@durman.ac.uk). The research of the second author was supported by the UK Natural Environment Research Council, grant NER/T/S/2002/00987, and by the Tyndall Centre for Climate Change Research, grant T2/13.

prediction). In this paper we have described what we believe to be the minimum amount of modelling necessary to allow us to transfer inferences from one or more simulators to the underlying system.

The outline of the paper is as follows. Our objective is to make probabilistic statements about the physical system using evaluations of our simulator and, where available, data collected from the system. For this purpose we must construct a probabilistic model that links the simulator and the system. We start in section 2 by considering the simplest case, in which we have a high-quality simulator with well-defined inputs. In section 3 we discuss the problem of ill-defined "tuning" inputs, and in section 4 we extend our analysis to include these and the possibility that the simulator is not of high quality. In section 5 we discuss belief models for the simulator itself, and in section 6 we extend our analysis to include multiple simulators for the same system. Section 7 concludes with a discussion. Our analysis is general, covering a wide range of computer experiments which share certain widely occurring characteristics, but we provide illustrations from the important field of climate modelling. More details of this application may be found in [3, 13] and, of particular interest to statisticians, [2, 12].

2. The direct simulator. We start by considering the best possible case for simulator-based inference about a physical system. We denote the system value y, where y typically comprises a collection of space- and time-indexed physical quantities. The possible values that y can take are the set $\mathcal{Y}$. To model this system, we have a deterministic simulator f, usually represented as computer code. For simplicity we will assume throughout that the output of f also takes values in the set $\mathcal{Y}$, so that we can compare the output of the simulator and the actual system directly.

We define the inputs of the simulator at a very general level to comprise the numbers used to initiate the computer code, i.e., the values that need to be specified before the code will execute. Typically, this will comprise (i) parameters in the equations describing the general behavior of the system and (ii) parameters, including boundary values and forcing functions, that tailor the general behavior of the system to a specific instance. There may also be quantities that control the way in which the system of equations is solved. If we wish to model explicitly our uncertainty as to how changes in such quantities would affect the quality of the computer solutions, then we might also represent these quantities as functional inputs into the simulator. However, this would further complicate many aspects of the account of our modelling. Therefore, for clarity of exposition, we treat the different values of such quantities as defining a family of different simulators. This will allow us to subsume our treatment of such quantities within our general treatment of families of related simulators for a physical system.

We want to use evaluations of a computer simulator of the system to help us reduce uncertainty about the system itself. We shall describe an approach that we consider appropriate for such an analysis. This approach will be formulated within a Bayesian statistical framework. In this framework, all probabilities are quantifications of the uncertainties of experts (e.g., [1]). This approach is appropriate for making such inferences, as the uncertainty about the relationship between the simulator and the system can only be captured by expert judgments. In our applications, the quantifications will be based on experience with the simulator, its underlying theory, its history of application, and any other prior knowledge relevant to the physical system. The quality and reliability of the inferences that we are able to make will, therefore, depend on the care and effort that is taken in assessing such considerations.

The Bayesian approach has been formulated to allow the expert to synthesize relevant prior information with data, such data in our case comprising observations on the physical system and evaluations of the computer simulator. Different experts may make different prior judgements and, therefore, reach different conclusions as a result of following this process. Therefore, there are two ways to employ the analysis that we shall describe. First, an individual expert may explore the implications of his or her individual judgments for reducing uncertainty over the quantities of interest. Second, by seeing how such prior judgments vary between experts, the analysis may be used to assess the degree to which such experts may reasonably disagree in their posterior judgments, given empirical evidence and evaluations of the simulator.

First, consider the case where the domain of f comprises only inputs that can, at least in principle, be determined by an experiment independent of the simulator itself. We call these *measurable inputs* and denote the space of measurable inputs as $\mathcal{X}$, with the true input denoted $x_0 \in \mathcal{X}$. Of particular interest is the degree to which $f(x_0)$ is able to represent y. We define the *discrepancy* to be the difference between the true value y of the physical system and the value of the simulator output at the true input value x_0, namely,

$$\epsilon = y - f(x_0). \tag{2.1}$$

In some situations, most notably where we have a perfect simulator and our only uncertainty is about, say, the initial value of the state vector, we may have $\epsilon = \mathbf{0}$. Generally, however, we would not expect the simulator to exactly replicate the system value at input x_0, and consequently we anticipate that $\epsilon \neq \mathbf{0}$.

We are often uncertain about x_0 due to our unwillingness or inability to perform the experiment that would determine x_0 to arbitrary accuracy. Some components of $\mathcal{X}$ are likely to be very well known, e.g., the gravitational constant or the Stefan–Boltzmann constant, but others, e.g., "historic" forcing functions or boundary conditions, are only known approximately. Likewise, we are uncertain about f, in all but trivial cases, due to the fact that we cannot costlessly evaluate f at every input value. We are uncertain about y in any circumstances where y may only be observed with error, or has not yet been observed, e.g., future values. Finally, we are uncertain about the discrepancy ϵ whenever the reasons for the mismatch between the simulator and the system are not fully understood.

When we assign a probability distribution for the quantities y, f, x_0, and ϵ to account for this uncertainty, then this distribution is constrained by (2.1). By making further assumptions that we now describe, we may reformulate the problem so that our uncertainty about ϵ, f, and x_0 jointly determine our uncertainty about y.

For this purpose we now define a certain type of simulator as follows. We say that f is a *direct simulator* if it has two properties. First, its inputs are measurable. Second, the discrepancy ϵ is probabilistically independent of x_0 and f, according to the judgments of the expert. This second property corresponds to the judgment that evaluations of the simulator at inputs other than x_0 are believed by the expert to convey no information about y additional to that in $f(x_0)$. That is, given knowledge of $f(x_0)$, there is no additional information contained in any further simulator evaluations for the purposes of predicting y as well as possible. Further, were the true values of f and x_0 to be revealed to the expert, then the uncertainty that the expert would afterwards specify for y would be the same for each possible value of f and x_0 that could be revealed. Thus, we exclude the kind of problem where we know, for example, that the solution method used by the simulator is very reliable for some parts of the input space but much less reliable for other parts of the input space. In this case the expert might want to make the predictability of the system using the simulator, e.g., as measured by $\mathsf{Var}[\epsilon]$, depend on x_0. Such considerations are unnecessary with a direct simulator.

For a direct simulator, the relationship between f, x_0, the system y, and observed system data z may be represented in the following *Bayesian graphical model*:

$$\begin{tikzcd}
 & x_0 \arrow[d] & \epsilon \arrow[d] & & e \arrow[d] \\
f \arrow[r] & f(x_0) \arrow[r] & y \arrow[r] \arrow[dr] & y_p \arrow[r] & z \\
 & & & y_f &
\end{tikzcd} \tag{2.2}$$

A (directed) Bayesian graphical model consists of a collection of nodes, where each node represents one or a collection of random quantities. The nodes are joined by arcs, and the construction $x \to y$, or x is a parent of y, indicates, informally, that knowledge of the value of x is relevant to the specification of the probability distribution of y, according to the judgment of the expert. The precise property of the graph is that any two nodes on the graph are conditionally independent given the values of all of their parent nodes. For a detailed discussion of such models, see [4, 10]. In particular, graphical models are useful in displaying how the various uncertainties in a problem relate to each other. For example, we read from (2.2) that our model has four independent sources of uncertainty, namely, f, x_0, ϵ, e. We combine f and x_0 to get $f(x_0)$; we combine $f(x_0)$ and ϵ to get y; we split y into y_p and y_f and combine y_p with e to get z. As our models and sources of information become more involved, it becomes increasingly useful to have simple visual representations which display qualitatively how the various aspects of uncertainty are combined.

In (2.2) we have partitioned the system y into (y_p, y_f), heuristically "past" and "future," namely, those quantities for which we already have observations (y_p), and those whose values we would like to assess (y_f), and introduced observational data z, which includes measurement error e, on the "past" system components. We may suppose that the measurement error is additive, so that

$$z = y_p + e, \tag{2.3}$$

although other models for the statistical errors are also possible.

The logical basis for inference using a direct simulator is straightforward. The quantities x_0, ϵ, and e are clearly defined, but there is uncertainty about their values. We may express this uncertainty as (mutually independent) probability distributions. In the general case, there may also be uncertainty about f, but for simplicity at this stage we treat f as known. Using (2.1) and (2.3) we can compute a full joint probabilistic specification over the collection (x_0, y_f, z) which can be used to answer, probabilistically, questions about the system.

For example, we can use observed values for z to help to reduce our uncertainty over x_0, and we can exploit this reduction in uncertainty to reduce our forecast uncertainty for y_f. Reducing uncertainty over x_0 is often termed (probabilistic) *calibration*, or, in the oil industry, *history matching*; see [16]. Reducing uncertainty over the unobserved future values y_f is often termed (probabilistic) *prediction*. Using system data to reduce uncertainty about inputs and, therefore, to help learn about the future values is often termed (probabilistic) *calibrated prediction.* In other cases, we have no system data z, and we can only learn about y_f in terms of our various sources of uncertainty. This is known as *uncertainty analysis.* In many problems of uncertainty analysis, the value of x_0 is selected from some population for each case to which the model is applied; for example, x_0 might be the unknown dose of radiation received by a particular patient, and the model output might be likely symptoms based on a computer model of the effects of different radiation levels on the patient.

We have

$$\mathsf{p}\,(x_0, y_f \mid z) \propto \mathsf{p}\,(x_0, y_f, z) = \mathsf{p}\,(z, y_f \mid x_0)\,\mathsf{p}(x_0). \tag{2.4}$$

Our model states that

$$(z, y_f) = f(x_0) \oplus \epsilon \oplus (e, \mathbf{0}), \tag{2.5}$$

where "$\oplus$" denotes the sum of independent vectors and (x, y) denotes the concatenation of two vectors x and y into a single vector. If we assume that ϵ and e are both Gaussian with mean zero, then the conditional distribution $(z, y_f) \mid x_0$ is Gaussian with mean $f(x_0)$ and variance $\mathsf{Var}[\epsilon] + \mathsf{Var}[(e, \mathbf{0})]$. In this case $\mathsf{p}(z, y_f \mid x_0)$ has the simple Gaussian functional form.

The practical implementation of these calculations may be complicated. In particular, the elicitation of expert knowledge about the variance structure of the discrepancy ϵ may be challenging; for example, if y is a time series, such as some important climate variables aggregated on a weekly basis, then beliefs about ϵ will reflect this time structure, and modelling for the future may be based on similar considerations to those in Bayesian forecasting for complex time series, where, if appropriate, we may use discrepancies between the model and historical data to modify beliefs about the underlying parameters generating the time series. However, despite such practical complications, the way in which the overall analysis should be carried out and the logical meaning of the various uncertainty statements are clear.

Simple example. We now introduce a simple example, which we shall develop in the following sections to illustrate the relevance of each of the features that we shall introduce. Our simulator is

$$f(x) = (a_p x, a_f x); \tag{2.6}$$

i.e., it has a single input x and two outputs, where we consider $a_p x$ to be the "past" output and $a_f x$ to be the "future" output, where a_p and a_f are known scalars but we are uncertain about the true input value x_0. In order to make inferences about x_0 and $y_f = a_f x_0$ using the system data z we need to provide distributions for x_0, ϵ, and e. We might choose mean-zero Gaussian distributions for the latter two, as this simplifies the inference as described above. This requires a 2×2 variance matrix for ϵ and a variance scalar for e. In general, there are no computational advantages that accrue for particular choices for the distribution $\mathsf{p}(x_0)$, so we will leave this undetermined.

With these assumptions, the calibrated prediction calculation for (2.4) is

$$\mathsf{p}\,(x_0, y_f \mid z) \propto \mathrm{N}_2((z, y_f) \mid f(x_0), \Sigma) \times \mathsf{p}\,(x_0)\,, \tag{2.7}$$

where "N_2" is the bivariate Gaussian density function with given mean and variance, and

$$\Sigma = \mathsf{Var}\,[\epsilon] + \begin{pmatrix} \mathsf{Var}\,[e] & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.8}$$

Clearly, this calculation generalizes straightforwardly to many inputs and many outputs with a known simulator, although Monte Carlo methods might be necessary in large problems; see [14] for more details.

3. Indirect simulators.

Unfortunately, very few of the simulators that we use in practice are direct simulators. This is because (i) they include inputs that are not measurable, and (ii) they are not sufficiently "good" that we are prepared to believe in a single "best" input value. Often both of these limitations apply.

3.1. "Tuning" inputs.

Within the full set of inputs to the simulator we find it useful to distinguish two main types: those which may be measured independently of the simulator, and those where no such measurement is meaningful. Measurable inputs were introduced in section 2. We term the other inputs *tuning inputs.* These are inputs which have meaning only with reference to the simulator.

The classification into "measurable" and "tuning" inputs is not always clear-cut. Consider, for example, the four-compartment model of the Atlantic described in [17], hereafter ZSR, for investigating the northward transport of heat from the tropics. We will use this model to illustrate various features of our approach throughout the paper. If we restrict our attention to an equilibrium analysis, then there are 18 inputs and eight outputs, the latter comprising equilibrium temperature and salinity for each compartment. Of the inputs (given in ZSR Table 1), five are measurable inputs, for example, the specific heat capacity and density of sea-water, and five are tuning inputs and are labeled as such. The remaining eight have physical analogues, for example, the volume and depth of each compartment; it is not immediately clear how to classify these inputs. A model of the Atlantic with four simply connected compartments is highly stylized, and we may be reluctant to attach too much physical meaning to each compartment. However, were we to construct a model of the Atlantic with four million small compartments arranged in a three-dimensional lattice that carefully respected the ocean margins, then, arguably, we would be concerned to match depths and volumes in the simulator at least approximately to their natural physical analogues. In section 4.1 we will introduce the possibility of "linking" a measurable input with a tuning input, which can be used to handle ambiguous cases.

It is important to understand the role of tuning inputs. They do not exist solely to permit a good fit to the system data z. Rather they make allowances for simulator imperfections. These imperfections are of two main types.

Poorly understood physics. Often the underlying physics (taken in its broadest sense) of the system is poorly understood or understood on a scale that is not appropriate to the simulator. In studies of climate change, for example, the large-scale behavior of clouds is an important determinant of the earth's albedo. However, cloud formation is not a well-understood process and detailed models reflecting current understanding would require information not available in a typical climate model (e.g., to account for the effect of localized "seeding" by the nonuniform distribution of atmospheric particulate matter). The same points could be made about the oceanic carbon cycle, or sea-ice. These types of subprocesses are represented in simulators in quite general terms, with parameters which do not relate directly to measurable attributes, but which attempt to compensate for aspects of the missing underlying physics, for example to bridge effects assessed at different scales.

Solver deficiencies. The second source of simulator imperfection is solver deficiencies. A given physical model expressed as ordinary or partial differential equations is almost invariably solved on a finite grid. Sometimes circumstances dictate that the grid must be larger than the characteristic scale of important physical processes. A well-known example of this is the treatment of viscosity in large coupled ocean/climate models. At the moment, the solution grid of these models is constrained by computing requirements to have a cell-size that is too large to capture the transport of water and heat by turbulence, which tends to be highly localized. Consequently, this transport is represented in the simulator by a parameterization of local turbulence in terms of tuning inputs such as horizontal and vertical "eddy viscosities." Experiments on the current generation of ocean simulators find that eddy viscosities need to be orders of magnitude larger than underlying molecular viscosity (which is measurable). As computers become more powerful and the solver resolution improves, then we may often find that these types of tuning inputs tend to well-defined limits, although, for any particular simulator, this may be subject to a variety of complex numerical and modelling issues.

Returning to the ZSR example, we may choose to treat the four-compartment volumes as measurable inputs, as we could divide the Atlantic into sections by latitudes from which volumes can be inferred, as is done in the paper. In this case we might want to treat the compartment depths as tuning inputs, because the appropriate averaging method for each compartment will depend upon physics that we are not clear about.

3.2. Indirect simulators. If our simulator is not a direct simulator, then we label it an *indirect simulator*. To include the tuning inputs we write $f : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$, where $\mathcal{X}$ is the space of measurable inputs and $\mathcal{U}$ of tuning inputs; where there are no tuning inputs we set $\mathcal{U} = \emptyset$. Note that we do not have to have tuning inputs for f to be an indirect simulator—we may simply feel that f is not good enough for us to want to connect it directly with the system via the true but unknown value x_0.

The treatment of indirect simulators, both in the literature and in practice, is somewhat inconsistent, falling somewhere between the following two extremes.

1. We pretend that the simulator really is a direct simulator with true physical input value (x_0, u_0) and carry out the analysis just as for the direct simulator as described above. We call this a *pseudodirect* analysis.

2. We take the view that the simulator is a device for forecasting, so that inputs (x, u) for which $\|z - f_p(x, u)\|$ is small are likely to give rise to a prediction error $\|y_f - f_f(x, u)\|$ that is also small. We call this a *black-box* analysis, in which we treat all inputs implicitly as tuning inputs.

The logical problems for the pseudodirect approach are clear. If there is no true value of (x, u), then there is no object over which it is meaningful to specify uncertainties. Further, there is no obvious way to express how much additional error we are introducing by the pretense of such a precise value. The logical problems for the black-box approach are even worse. If there is no physical basis for the terms of our model, in the sense that we do not even claim a generalized relationship between the inputs and some physical counterparts in the underlying system, then it is very difficult to construct a logical argument for the claim that good calibration in the past should result in good forecasts for the future. For example, we may be able to achieve many perfect matches using a purely statistical approach (e.g., by fitting high-order polynomials): are all of our predictions equally good, or equally bad?

Current practice in climate research with large simulators is to perform an ensemble of runs at different input values and use these as the basis of inference. Suppose we are interested in the mean value of future climate, given system data z and runs of the simulator at $(x_1, u_1), \dots, (x_n, u_n)$. Typically, this mean value is estimated as a weighted sum of $f_f(x_i, u_i)$, where the weights are determined as some function of

$\|z - f_p(x_i, u_i)\|$; the norm in this case accounts for factors such as the observation error variance structure. So far, this approach is consistent with the black-box view. However, the (x_i, u_i) are sampled from a prior distribution that reflects beliefs about the underlying inputs—this is more consistent with the pseudodirect view. This combined approach has been adopted, for example, for the innovative www.climateprediction.net experiment.

We should add that statisticians developing methodology in this area, including ourselves, are not exempt from criticism. Our own papers using the Bayes linear approach [5] propose what we now refer to as a pseudodirect analysis. The fully Bayesian approach of Kennedy and O'Hagan [11] tends toward a black-box analysis (see, particularly, their response to the discussion on [p. 461]).

4. Direct analysis using an indirect simulator. We need a way to connect our indirect simulator to the underlying system. In particular, we need to understand the precise role of the tuning inputs. This will allow us to have well-defined beliefs about the tuning inputs which can be used in inference about the system. A natural part of the linkage between an indirect simulator and the system is an assessment of the degree to which the indirect simulator fails to function as a direct simulator. Therefore, we augment f with a description of a further *direct* simulator to which the indirect simulator approximates. Call this hypothetical direct simulator the *direct version* of f, denoted f_D. For the moment we assume that f_D has the same measurable input space as f, so that $f_D : \mathcal{X} \to \mathcal{Y}$. We have now divided the problem of linking our simulator f and the system y into two parts. First, f tells us about f_D; second, $f_D(x_0)$ tells us about y. Note that in this formulation the value $f(x_0, u)$ is not especially informative about the system, except insofar as it is informative about $f_D(x_0)$.

In this context the role of the tuning inputs is to capture some of the difference between f and f_D. There are various levels of detail to which we may describe how tuning works within the simulator. The simplest view (generalized in section 6.1), which is sufficient to demonstrate our general approach and will be adequate for many problems, is to treat the tuning inputs in an analogous way to that in which we have treated the direct simulator in section 2. Thus, we suppose that there is an unknown value u_0 with the property that, if we knew this value, then we would only evaluate the function $f(x, u)$ at $u = u_0$ in order to learn about the form of the direct simulator $f_D(x)$. This does not mean that $f(x, u_0) = f_D(x)$, but rather that the conditional probability distribution $f_D(x) \mid u_0$ depends only on the function $f(x, u_0)$. Equivalently, the simulator provides no information about the *functional discrepancy* between f and f_D,

$$\epsilon_D(x) = f_D(x) - f(x, u_0), \tag{4.1}$$

beyond that which may be obtained from evaluating the simulator at u_0, for each value of x. In particular, we might set $\mathsf{E}[\epsilon_D(x)] = 0$ so that $\mathsf{E}[f_D(x) \mid u_0] = f(x, u_0)$. Typically, we will have some view as to the likely order of magnitude discrepancy between the tuned version of f and the value of f_D for a typical input x, which will suggest the variance for $\epsilon_D(x)$, usually the same for each x. Similarly, an order of magnitude view as to how large a change in x would be required to make a large change in the functional discrepancy may be used to suggest a correlation parameter for the process. More sophisticated views concerning likely differences in tuning achievable in different regions of the input space could similarly be introduced into the belief specification. We have, from (4.1), that

$$\begin{aligned}\mathsf{Var}\left[f_D(x)\right] &= \mathsf{Var}\left[\mathsf{E}\left[f_D(x) \mid u_0\right]\right] + \mathsf{E}\left[\mathsf{Var}\left[f_D(x) \mid u_0\right]\right] \\ &= \mathsf{Var}\left[f(x, u_0)\right] + \mathsf{Var}\left[\epsilon_D(x)\right].\end{aligned} \tag{4.2}$$

The first term in (4.2) expresses the variation in $f_D(x)$ which may be removed by careful tuning, while the second term expresses the residual variation between the tuned version of the simulator and $f_D(x)$. Tuning inputs and the directed version of f are incorporated onto the graphical model as follows:

(4.3)

$$\begin{tikzcd} & \epsilon_D \arrow[dr] & u_0 \arrow[d] & x_0 \arrow[d] & \\ f \arrow[rr] & & f_D \arrow[r] & f_D(x_0) \arrow[r, dashed] & \text{same as (2.2)} \end{tikzcd}$$

Returning to the calibrated prediction calculation, we now add u_0 to the collection of quantities we want to learn about using z, to get

$$\mathsf{p}\left(x_0, u_0, y_f \mid z\right) \propto \mathsf{p}\left(z, y_f \mid x_0, u_0\right) \mathsf{p}\left(x_0\right) \mathsf{p}\left(u_0\right). \tag{4.4}$$

Our model now states

$$(z, y_f) = f(x_0, u_0) \oplus \epsilon_D(x_0) \oplus \epsilon \oplus (e, \mathbf{0}). \tag{4.5}$$

If, in addition to our previous Gaussian assumptions, we assume that ϵ_D is a Gaussian random field with zero mean, then the distribution $(z, y_f) \mid (x_0, u_0)$ is Gaussian with mean $f(x_0, u_0)$ and variance $\mathsf{Var}[\epsilon_D](x_0) + \mathsf{Var}[\epsilon] + \mathsf{Var}[(e, \mathbf{0})]$. Once again the practical implementation may be tricky, involving the elicitation of the distribution of u_0 and the variance kernel of ϵ_D, but the procedure and the meaning of the various quantities are clear.

Note that, computationally, our beliefs about f_D are induced by our beliefs about u_0 and ϵ_D. This is not to say that u_0 and ϵ_D are necessarily the primitive quantities. It may be that our beliefs about f and f_D give rise to beliefs about u_0 and ϵ_D. We can always check that our two sets of beliefs are consistent and plausible by integrating u_0 and ϵ_D out of f_D explicitly.

In the ZSR model, we are already provided with several tuning inputs. But we must first consider whether the experts feel comfortable with the idea of a direct simulator defined only on the five measurable inputs. For a model this simple, we may suppose that the answer is almost certainly "No." Therefore, we must consider alternative rationales for interpreting the output of the simulator. The simplest case for an indirect simulator is that there exists some unknown value u_0 for the tuning inputs such that $f_D(x) = f(x, u_0) + \epsilon_D(x)$, where f_D is a direct simulator, so that $y = f_D(x_0) + \epsilon$. The next simplest case is to assert that our simulator cannot be turned into a direct simulator by appropriate choice of u_0, but that there is *another* simulator defined on $\mathcal{X}$ which can, and to which our own simulator approximates. This simulator, which we call the *top simulator*, will be introduced in section 6.1. But we may not believe that any simulator defined on $\mathcal{X}$ alone is good enough to function as a direct simulator. Perhaps, for example, we may consider that no model this highly aggregated can sufficiently account for the complex spatial patterns of Atlantic currents, but that a model with more compartments might. Therefore, a simple approach based on (4.1) will not be acceptable, and we will have to be more subtle in the way that we relate the simulator and the system. We will discuss such alternative relationships in section 6.3.

Example (cont). We will develop our example both with and without a tuning input. Suppose our beliefs about f_D are that it has the same functional form as f but that x is likely to have a larger impact on f_{Dp} and a smaller impact on f_{Df}. It may be that this is already embodied in a tuning input for f, for example,

$$f(x, u) = ((a_p + u)x, (a_f - u/2)x) \tag{4.6}$$

for some tuning input $u > 0$. We may choose in this case to use a simple stationary Gaussian process for ϵ_D and set $\mathsf{Var}[\epsilon_D](x) = \Sigma^D$ for all x. With these additional assumptions and a prior distribution for u_0, the calibrated prediction calculation is

$$\mathsf{p}\left(x_0, u_0, y_f \mid z\right) \propto \mathrm{N}_2((z, y_f) \mid f(x_0, u_0), \Sigma) \times \mathsf{p}\left(x_0\right) \times \mathsf{p}\left(u_0\right), \tag{4.7}$$

where

$$\Sigma = \Sigma^D + \mathsf{Var}\left[\epsilon\right] + \begin{pmatrix} \mathsf{Var}\left[e\right] & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.8}$$

Alternatively, we may not have a tuning input with this feature, in which case our beliefs will need to be modelled by a more careful choice for ϵ_D. Write $f_D(x)$ as

$$f_D(x) = (a_{Dp}x, a_{Df}x) \tag{4.9}$$

for unknown parameters a_{Dp} and a_{Df}, so that

$$\epsilon_D(x) = f_D(x) - f(x) = ((a_{Dp} - a_p)x, (a_{Df} - a_f)x) = (v_p x, v_f x), \tag{4.10}$$

where v_p and v_f are uncertain quantities. Our beliefs about v_p and v_f then induce beliefs about the random field ϵ_D. In this case, beliefs corresponding to (4.6) would suggest that $\mathsf{E}[v_p] > 0$ and $\mathsf{E}[v_f] < 0$. We have the choice here of modelling v_p and v_f jointly (which would be a generalization of the joint model induced by the tuning input u), or modelling them independently. Either way, our calibrated prediction calculation would be

$$\begin{aligned} \mathsf{p}\left(x_0, y_f \mid z\right) \propto \iint & \mathrm{N}_2((z, y_f) \mid f(x_0) + (v_p x_0, v_f x_0), \Sigma) \\ & \times \mathsf{p}\left(x_0\right) \times \mathsf{p}\left(v_p, v_f\right) dv_p dv_f \end{aligned} \tag{4.11}$$

for some choice of density $\mathsf{p}(v_p, v_f)$, where (v_p, v_f) are treated as nuisance parameters and have been integrated out; Σ is as defined in (2.8).

4.1. Linking to measurable inputs. In many cases we want to treat a measurable input as a tuning input, usually because insight and experience combine to suggest that this is predictively effective. In other words, allowing a measurable input to move away from its actual value can offset some of the deficiencies of the simulator. Viscosity in ocean models, as discussed in section 3.1, provides a good example. Where this is the case, we suggest that the original measurable input remains in $\mathcal{X}$ but that a tuning input is introduced into $\mathcal{U}$, which modifies the impact of the measurable input in the simulator. We term this *linking* to the measurable input. Thus $\mathcal{X}$ contains the original measurable input ν, say, and $\mathcal{U}$ contains the multiplier m_ν, and the effective value of the input in the simulator is $m_\nu \times \nu$, or some other known deterministic function of m_ν and ν. This will require us either to modify the simulator code or to provide a wrapper to the simulator to implement the transformation.

We can put our uncertainty about ν into our beliefs about x_0, and we can put our uncertainty about how the role of x differs in f and f_D into our beliefs about u_0.

In the ZSR example we may choose to link to the thermal and haline expansion coefficients. These enter the model on the presumption of a linear relationship between north-south temperature and salinity gradients and the meridional volume transport (i.e., the quantity of water flowing northward). Insofar as this relationship is approximate, we may choose to replace measurable inputs with more adaptable tuning inputs. This is not necessarily the best response to functional uncertainty—we could, for example, use a random field with a linear mean function—but it serves as a simple "quick fix," allowing us to introduce some uncertainty attributable to the linear approximation without disproportionate effort. Of course, linking to these measurable inputs does not compensate for the more fundamental problems that arise if there is no direct version of the simulator defined on $\mathcal{X}$, as already discussed.

Example (cont). Suppose we wanted to link to x in our example, where for simplicity we do not have the tuning input u. In this case we would create the "wrapper" function $g(x, m) = f(mx)$, where m is our new tuning input, and we would have

$$f_D(x) = g(x, m_0) \oplus \epsilon_D(x) = ((a_p m_0)x, (a_f m_0)x) \oplus \epsilon_D(x). \tag{4.12}$$

Thus our beliefs about m_0 induce beliefs about the parameters of f_D in quite a different way than those induced, for example, by (4.6). If we thought that a_{Dp} and a_{Df} were about the same as a_p and a_f, then we might choose a gamma distribution for m_0 with mean 1 and small variance. For the calibrated prediction calculation we would proceed in a similar way to (4.7), replacing f with g and u_0 with m_0.

In this example we have now seen three different ways to model the relationship between f and f_D: through existing tuning inputs (if they are appropriate), through a careful choice for ϵ_D, and by linking to measurable inputs. They each have different implications for the distribution of $f_D \mid f$ and so provide a range of modelling options for the expert.

5. Statistical emulators. Up until now, we have assumed that the simulator f is known to the analyst. In practice, our precise knowledge of f often extends only as far as a finite collection of evaluations at known inputs, $F = \{f(x_1), \ldots, f(x_n)\}$. This set of inputs can be very small with respect to the dimension of the input space; for example, coupled ocean/climate simulators have large input spaces but can take weeks (or longer) for a single run. Unless we can evaluate the simulator instantaneously, we will have to treat its output at arbitrary x as uncertain for computational purposes. This means that we have to add the data from the simulator evaluations to our graphical model. In the simplest case of a single direct simulator, the graphical model (2.2) becomes

$$\begin{tikzcd}
 & x_0 \arrow[d] & \\
f \arrow[r] \arrow[d] & f(x_0) \arrow[r, dashed] & \text{same as (2.2)} \\
F & &
\end{tikzcd} \tag{5.1}$$

Now we must provide a probabilistic description of our beliefs about f, which is a random field indexed by $x \in \mathcal{X}$, and then update those beliefs using F. The use of stochastic processes to model deterministic functions such as computer simulators has been widely studied; see, for example, [15] and the more recent references in [5, 11]. We give here a brief outline of the approach that we have found effective in physical modelling with large numbers of inputs and outputs.

A natural and convenient way to represent beliefs about f is to represent f as the sum of two components. The first component expresses our beliefs about the systematic variation in f given x, and the second component captures residual variation with local structure

$$f_i(x) = \sum_{j \in J_i} \beta_j^{(i)} L_j(x) + \delta_i(x), \quad i = 1, \ldots, k = \dim \mathcal{Y}, \tag{5.2}$$

where i indexes the components of the simulator output, J_i is a collection of indices, the L_j are known functions of x, and δ_i is a stationary random field with zero mean, independent of $\beta = (\beta^{(1)}, \ldots, \beta^{(k)})$. The expression (5.2) is often termed a *statistical emulator*, or simply an emulator, for the function $f_i(x)$. The emulator expresses the beliefs of the expert about the value of the function for each x, in a convenient and easily computable form. When the expert has specified probabilistic beliefs concerning the coefficients β and the parameters of the stationary field $\delta_i(x)$, then this automatically generates probabilistic beliefs about the value of $f_i(x)$ for each x. The use of such emulators is standard within a wide range of analyses of computer experiment. More details about the use of statistical emulators may be found in [6, 7].

It is often convenient to represent L_j as a product of polynomials in individual components of x so that J_i comprises p-tuples of nonnegative integers (where $p = \dim \mathcal{X}$), and

$$L_j(x) = L_{j_1}(x_1) L_{j_2}(x_2) \cdots L_{j_p}(x_p), \tag{5.3}$$

where $L_v(x_k)$ is a polynomial of degree v in the kth component of x. For example, if $\mathcal{X} = [-1, 1]^p$, then the product of Legendre polynomials provides a basis for continuous squared-integrable functions on $\mathcal{X}$.

The δ_i field is parameterized by its covariance kernel. The choice

$$\mathsf{Cov}\left[\delta_i,\, \delta_{i'}\right](x, x') = \Sigma^{\delta}_{ii'} \exp(-\theta \left\| x - x' \right\|^2), \quad \theta > 0, \tag{5.4}$$

is often used to reflect the belief that the underlying function f is very smooth, with derivatives of all orders everywhere in x. In geostatistics it is common to treat f_i as stationary and model it entirely using δ_i and a careful choice of covariance kernel (or, equivalently, *semivariogram*; see [8]). In large problems with a relatively small number of evaluations we prefer to capture the main effects explicitly in a nonstationary model with several carefully chosen L_j terms, using a combination of expert elicitation with a detailed analysis of data from "coarsened" versions of the simulator f, which run much faster but are correspondingly less accurate. If δ contributes only a small amount of variation to the model, our beliefs about f are driven primarily by our choices for L and our beliefs about the coefficients β. For simplicity we will largely ignore the contribution of δ in the following discussion and treat it as a simple "nugget" with covariance kernel

$$\mathsf{Cov}\left[\delta_i,\, \delta_{i'}\right](x, x') = \begin{cases} \Sigma^{\delta}_{ii'}, & x = x', \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{5.5}$$

While we are at liberty to make any reasonable distributional assumptions for β and δ, the inferential calculations for updating beliefs about f by the evaluations F are much simplified by taking β to be Gaussian and δ to be a Gaussian random field independent of β, in which case f is itself a Gaussian random field—sometimes it may be necessary to transform the output of the simulator to make this appropriate. In this case the updating of beliefs about β using the outcomes of the evaluations F is straightforward because (F, β) is jointly Gaussian.

For inference, (2.5) still holds, but the distribution of $(z, y_f) \mid x_0$ now has mean $\mu(x_0)$ and variance $\kappa(x_0) + \mathsf{Var}[\epsilon] + \mathsf{Var}[(e, \mathbf{0})]$, where $\mu(x) = \mathsf{E}[f](x)$ is the mean function of f and $\kappa(x) = \mathsf{Var}[f](x)$ the variance kernel, both computed using updated beliefs $\beta \mid F$. This distribution is Gaussian if β, δ, ϵ, and e are all Gaussian, and, computationally, this is a compelling reason for making this choice of distribution (perhaps after transforming the output of f) unless both the input and output spaces of f are small.

The ZSR simulator provides an interesting challenge for emulation, because its equilibrium state may be solved analytically, while its transient behavior must be solved numerically for given forcing (in this case, of environmental temperature and fresh-water flux through time). This is not unusual. Dynamic simulations are often started from equilibrium conditions, as a period of stability can be identified in the system record (e.g., preindustrial climate) and this reduces the size of the input space. It is also quite common that we have some analytic knowledge of the equilibrium state (e.g., in the case of climate, from energy balance models). Therefore, the components of the emulator as indexed by time must make a transition from a specific functional form with well-known β coefficients to a more generic functional form (e.g., a subset of a basis) with more uncertain β coefficients. One way to achieve this is to take the collection $\{L_1, L_2, \dots\}$ to be the union of the known and generic components and to arrange for the prior mean and variance of the set of $\beta^{(i)}$ vectors to reflect a transition from one to the other according to the time value of i.

Example (cont). In our example, we have been treating f as a known function, e.g.,

$$f(x, u) = ((a_p + u)x, (a_f - u/2)x). \tag{5.6}$$

If we thought that f was only approximately of this form, then we might express this belief through an emulator of the form

$$f(x, u) = (a_{p0} + a_{p1}x + a_{p2}ux + \delta_p, a_{f0} + a_{f1}x + a_{f2}ux + \delta_f), \tag{5.7}$$

where we are uncertain about the 6-vector $a = (a_p, a_f) = (a_{p0}, \dots, a_{f2})$. We treat the 2-vector $\delta = (\delta_p, \delta_f)$ as a simple nugget with variance Σ^δ, where the magnitudes of the variances quantify our beliefs about the quality of the approximation.

Treating (5.7) as an emulator, we require prior beliefs about the quantities in a; we will assume that they are jointly Gaussian, with given mean vector and variance matrix, and independent of δ. In our prior beliefs we might want conditions such as $\mathsf{E}[a_{p0}] = \mathsf{E}[a_{f0}] = 0$, $\mathsf{E}[a_{p2}] = 1$, and $a_{f2} = -a_{p2}/2$ with probability 1. Having assigned the mean and variance, $f(x)$ is a Gaussian random field with mean function and variance kernel

$$\mu(x,u) = \begin{pmatrix} L(x,u)^\intercal \mu_p \\ L(x,u)^\intercal \mu_f \end{pmatrix}, \tag{5.8}$$

$$\kappa(x,u) = \begin{pmatrix} L(x,u)^\intercal \Sigma_{pp} L(x,u) & L(x,u)^\intercal \Sigma_{pf} L(x,u) \\ L(x,u)^\intercal \Sigma_{fp} L(x,u) & L(x,u)^\intercal \Sigma_{ff} L(x,u) \end{pmatrix} + \Sigma^\delta, \tag{5.9}$$

where $L(x,u) = (1, x, ux)$. Even before we evaluate the simulator we may make inferences about (x_0, u_0, y_f). Under the same assumptions as (4.7), our calibrated prediction calculation for the system is

$$\mathsf{p}\left(x_0, u_0, y_f \mid z\right) \propto \mathrm{N}_2((z, y_f) \mid \mu(x_0, u_0), \Sigma(x_0, u_0)) \times \mathsf{p}\left(x_0\right) \times \mathsf{p}\left(u_0\right), \tag{5.10}$$

where now

$$\Sigma(x,u) = \kappa(x,u) + \Sigma^D + \mathsf{Var}\left[\epsilon\right] + \begin{pmatrix} \mathsf{Var}\left[e\right] & 0 \\ 0 & 0 \end{pmatrix}. \tag{5.11}$$

This is now a slightly more complicated calculation than (4.7), as the variance as well as the mean of the Gaussian density function varies with (x_0, u_0).

The only effect of performing evaluations of f in the above calculation is to change the mean and variance of the vector of coefficients, a. This is because f separates F from the rest of the objects on the graph (5.1), and we have assumed a simple nugget form for δ. Updated beliefs about a change the mean function and covariance kernel, and so affect the conditional distribution $(x_0, u_0, y_f) \mid (z, F)$, where we now add F to the conditioning set. Because calculations based on Gaussian emulators remain tractable even for large simulators we can use the emulator both in the inferential calculation and also off-line, for example, to make informative choices of inputs at which to evaluate the simulator.

6. Multiple simulators. In challenging problems we can expect there to be several different simulators for a given physical system. We have the choice of performing a rigorous probabilistic analysis for each simulator and then attempting an informal synthesis at the end, or generalizing our approach outlined above to link multiple simulators and the system within a single coherent belief model. Either way, we will have to confront two questions: (i) how similar are the simulators to each other? and (ii) how "good" is each simulator? Without answers to these questions we will not know how to weigh the contribution from each simulator to our joint inference. The natural and coherent approach is to link each of the simulators and the system together within a single belief model which fully accounts for the common and the distinctive information about the system that is provided by each simulator.

6.1. The "top" simulator. The simplest case of multiple simulators occurs when we have only a single simulator f, but we want to weaken our assertion that there exist a u_0 and a ϵ_D such that $f_D(x) = f(x, u_0) \oplus \epsilon_D(x)$. As with the direct simulator and the system, this is a belief statement that f is "sufficiently good" that we can link it to f_D via an unknown value u_0 for u. If we do not think that f is tunable in this way, then we need to consider how f should be modified in order for it to be a tunable simulator. To make this link we introduce a second simulator which *is* sufficiently good but which we do not know. We denote this simulator the *top* *simulator* and write it $f^* : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$. This is represented in the following graphical model:

$$\begin{tikzcd}
 & \epsilon_D \arrow[dr] & u_0 \arrow[d] & \\
f \arrow[r] \arrow[d] & f^* \arrow[r] & f_D \arrow[r, dashed] & \text{same as (4.3)} \\
F & & &
\end{tikzcd} \tag{6.1}$$

Thus we use our runs F to tell us about f_D, but in order to pass this information along we must construct a joint model for (f, f^*).

In practice we will have described our beliefs about f in the form of an emulator such as (5.2). It is natural to describe our prior beliefs about the relationship between f and f^* through the relationship between the emulators of the two functions. Therefore we construct an emulator for f^* to express probabilistic beliefs about the value of $f^*(x)$ for each x. For simplicity, at this stage, we suppose that we choose the same form as that of the emulator for f. This corresponds to the class of problems where we have no information which would cause the expert to have qualitatively different beliefs about the effects of changes of x on changes in $f^*(x)$ than about the effect of such changes on $f(x)$. If we had such information, then this would be reflected in the qualitative differences in the emulators that we would construct. Examples of such constructions are given in section 6.3. Thus, for now, we suppose that the emulator is

$$f_i^*(x) = \sum_{j \in J_i} \gamma_j^{(i)} L_j(x) + \delta_i^*(x), \quad i = 1, \ldots, k, \tag{6.2}$$

and treat both (β, γ) and (δ, δ^*) as jointly Gaussian. Then our beliefs about the relationship between f and f^* are described by the covariance between β and γ, and between δ and δ^*.

This type of multiple-emulator construction is particularly useful when we have more than one simulator for a physical system, as we will now discuss.

6.2. Multiple simulators with the same input space. The simplest generalization to more than one actual simulator occurs for a collection of simulators with the same input space. This will often happen when the solution method for constructing the outputs for given inputs is very complicated and computationally expensive. Consider, for example, a set of partial differential equations, to be solved by an implicit time-marching method. At each time point we must perform a large (sparse) matrix inversion. To perform the inversion we choose an iterative method. It may be computationally infeasible with existing resources to iterate all the way to convergence for every time-step, and, therefore, our simulator truncates the iteration after a given number of steps, or when an error estimate becomes small. Hence we can imagine a sequence of simulators, with different numbers of steps, all with the same input space.

Likewise, as technology improves it becomes possible to solve the same differential equations on a higher-resolution grid. With a careful treatment of spatial input fields this can also give rise to a sequence of simulators with the same input space. In both of these cases it would be advantageous to combine the information from different simulators in a formal way. This would allow us to make informed choices about how best to reduce our uncertainty about the system—for example, by doing a few runs of a very expensive simulator, or many runs of a cheap one, or some combination of the two. It also means that we do not have to discard or downgrade the results from an old simulator that becomes superseded by a better one. In weather forecasting, increases in computing power have typically resulted in the construction of larger simulators rather than in increased numbers of evaluations of existing simulators.

Following the introduction of the top simulator, the treatment of multiple simulators with the same input space is conceptually simple. A joint model over all available simulators and the top simulator allows us to pass information from each set of evaluations into our beliefs about f_D. However, the way in which we choose to implement the joint model will depend on the situation. The simplest case, and one that often occurs in practice, is where there is a simple ranking across the simulators, so that we can easily say that simulator 2 is better than simulator 1. Both of the examples given above tend to lead to rankings of this kind, with the better simulators having the greater number of iterations, or the higher spatio-temporal resolution. In this case we might want to impose a Markov structure across simulators. This will both reduce the number of uncertainty assessments that we need to make and also simplify the computations that are required to update beliefs over the model given evaluations of the various simulators. Therefore, we might construct a joint model of the form

$$\begin{tikzcd}
 & & \epsilon_D \arrow[dr] & u_0 \arrow[d] & \\
f_1 \arrow[r] \arrow[d] & f_2 \arrow[r] \arrow[d] & f^* \arrow[r] & f_D \arrow[r, dashed] & \text{same as (4.3)} \\
F_1 & F_2 & & &
\end{tikzcd} \tag{6.3}$$

where we have two simulators, and we understand the statement "f_2 is better than f_1" to mean that f_2 separates f_1 from the top simulator f^*. This Markov structure imposes strong restrictions on the joint distribution of the emulators' coefficients and δ-terms, such that the emulators for f_1 and f^* are conditionally independent given the emulator for f_2.

Of course, in many situations it will not be possible to give such an unambiguous ranking. In such cases, we need to complete our specification in a way which does not impose an ordering on the quality of the simulators. We now describe one such approach.

6.3. General input spaces. Generally, we can expect that different simulators of the same physical system will not be easily rankable. Further, they may have different input spaces because of the different modelling assumptions and simplifications made in translating the model into solvable form. In fact there can often be more simulators than there are research groups. In ocean/climate modelling a single group might develop both a large-scale general circulation model and a faster but less accurate intermediate complexity model, where the latter is not just a low-resolution version of the former but incorporates a different treatment of the physics. It is also common to adopt a modular approach in the simulator options, in which subprocesses, such as an oceanic carbon cycle, can be switched on or off. We cannot necessarily proceed as though the "off" simulator is just the "on" simulator with some inputs set to zero. Integrated models, such as the Integrated Assessment Model being developed by the Tyndall Centre, take this a step further by defining a common interface that will allow "third party" modules to be linked together into a single simulator. In this most general case we do not expect there to be a simple ranking across simulators, although certainly the experts will have beliefs that some simulators are better in some respects than others.

In combining information from these simulators, the distinction between measurable inputs and tuning inputs is helpful. Measurable inputs, being defined outside of any simulator, can be pooled across simulators. Tuning inputs, on the other hand, belong to a given simulator. Consider the case of two simulators,

$$f_A : \mathcal{X}_A \times \mathcal{U}_A \to \mathcal{Y}, \qquad f_B : \mathcal{X}_B \times \mathcal{U}_B \to \mathcal{Y}, \tag{6.4}$$

where $\mathcal{X}_A = \mathcal{X}_1 \times \mathcal{X}_2$ and $\mathcal{X}_B = \mathcal{X}_1 \times \mathcal{X}_3$, so that $\mathcal{X}_1$ is common to both simulators. Following our previous approach, we need to consider a top simulator f^* that both f_A and f_B are informative for. Its measurable input space is the union of the individual input spaces, $\mathcal{X}^* = \mathcal{X}_A \cup \mathcal{X}_B = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, but its tuning input space is the product, $\mathcal{U}^* = \mathcal{U}_A \times \mathcal{U}_B$.

In the ZSR example, we may consider that no simulator defined on the original measurable input space is sufficiently good to be treated as a direct simulator, i.e., that there is no top simulator defined on $\mathcal{X}$ but that a similar model with more compartments might suffice. We could, for example, take each of the original four compartments and subdivide them. In this way the compartment volumes and depths of the original model can be taken as linear combinations of those in the expanded model, so that the old input space is a subset of the new, which will simplify the joint statistical modelling of the two simulators. It is not necessary for us to build this new simulator, but it *is* necessary for us to have beliefs about it, both in relation to the original simulator and to the system, in order that we might construct a probabilistic link between the four-compartment simulator that we have and the Atlantic ocean that we want to make inferences about. We consider that it is easier to think about the relationship between the four-compartment model and the more-compartment model, and the more-compartment model and the Atlantic, than it is to think directly about the relationship between the four-compartment model and the Atlantic.

There is an alternative approach for the ZSR example, which is to introduce a second actual simulator. The ZSR paper concerns the fitting of the four-compartment model to a larger climate simulator of intermediate complexity, CLIMBER-2. It may be that CLIMBER-2 is an indirect simulator for which a direct version is considered to exist. In this case the joint statistical modelling of the ZSR model and CLIMBER-2 is sufficient, in conjunction with beliefs about u_0 and $\epsilon_D(x)$ which relate CLIMBER-2 to its directed version, and about x_0 and ϵ, which relate that directed version to the Atlantic. Alternately, it may be that a direct version of CLIMBER-2 is not considered to exist, but there is a top simulator, perhaps CLIMBER-2 solved with smaller time-steps. In this case the joint statistical modelling of the ZSR model and CLIMBER-2 must be augmented with the joint statistical modelling of CLIMBER-2 and the top simulator, and we may choose to impose a Markov structure on the three simulators. Then the top simulator must be related to the Atlantic through u_0, $\epsilon_D(x)$, x_0, and ϵ, as before.

And, of course, it may be that no top simulator is thought to exist defined on the CLIMBER-2 input space, in which case we must introduce a further simulator, say, an ocean general circulation model (OGCM). In the statistical modelling of this OGCM and its relationship with CLIMBER-2 we may have a specific OGCM in mind, which would be useful if we were able to utilize its evaluations, or we may have an ideal OGCM in mind, which might simplify the modelling.

6.4. Statistical modelling. Modelling the joint behavior of functions with different input spaces is extremely challenging. For reasons of space we can only sketch here an approach based on the construction of additional emulators that are used to share information across simulators. We consider two simulators

$$f_A : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{Y} \quad \text{and} \quad f_B : \mathcal{X}_1 \times \mathcal{X}_3 \to \mathcal{Y}, \tag{6.5}$$

where, for simplicity, all inputs are measurable. We decompose each simulator into orthogonal emulators

$$f_A(x_1, x_2) = g_A(x_1) \oplus g_A(x_1, x_2), \tag{6.6}$$

$$f_B(x_1, x_3) = g_B(x_1) \oplus g_B(x_1, x_3), \tag{6.7}$$

where $g_A(\cdot)$ is the emulator for f_A with x_2 fixed at some baseline value and $g_A(\cdot, \cdot)$ is the "residual," with $g_A(\cdot)$ independent of $g_A(\cdot, \cdot)$, and similarly for f_B. If $\mathcal{X}_2$ in simulator A is providing separate information from $\mathcal{X}_3$ in B, then we can express this in terms of orthogonality of the emulators across the two functions and take $g_A(\cdot)$ and $g_B(\cdot, \cdot)$ to be independent; the same is true for $g_B(\cdot)$ and $g_A(\cdot, \cdot)$. This belief model is more easily understood in the following graphical model (where we include the input values for clarity):

$$\begin{tikzcd}
f_A(x_1, x_2) & & f_B(x_1', x_3') & \\
g_A(x_1) \arrow[u] \arrow[rr, bend right] & g_A(x_1, x_2) \arrow[ul] & g_B(x_1') \arrow[u] & g_B(x_1', x_3') \arrow[ul]
\end{tikzcd} \tag{6.8}$$

Thus, if we evaluate f_A at (x_1, x_2), then this information passes through to f_B at (x_1', x_3') via the relationship between $g_A(x_1)$ and $g_B(x_1')$.

Now we can connect both f_A and f_B to the top simulator f^* after decomposing f^* into four orthogonal emulators

$$f^*(x_1, x_2, x_3) = g^*(x_1) \oplus g^*(x_1, x_2) \oplus g^*(x_1, x_3) \oplus g^*(x_1, x_2, x_3), \tag{6.9}$$

where the final term comprises information that we cannot infer from either f_A or f_B. We connect the three functions through emulators with common input spaces to give the full joint model, in which we also include evaluations F_A and F_B and the direct simulator f_D:

$$\begin{tikzcd}
F_A & & F_B & \\
f_A(x_1, x_2) \arrow[u] & & f_B(x'_1, x'_3) \arrow[u] & \\
g_A(x_1) \arrow[u] \arrow[rr, bend left] \arrow[drr] & g_A(x_1, x_2) \arrow[ul] \arrow[d] & g_B(x'_1) \arrow[u] \arrow[d] & g_B(x'_1, x'_3) \arrow[ul] \arrow[d] \\
g^*(x''_1, x''_2, x''_3) \arrow[drr, bend right] & g^*(x''_1, x''_2) \arrow[dr] & g^*(x''_1) \arrow[d] & g^*(x''_1, x''_3) \arrow[dl] \\
 & & f^*(x''_1, x''_2, x''_3) \arrow[d] & \\
 & & f_D &
\end{tikzcd} \tag{6.10}$$

where the second and third rows are the same as (6.8), and the third and fourth rows connect like emulators into the decomposition of f^*. The graphical model continues through f_D in the same way as (6.3).

If we want to model the belief that f_B is a better model than f_A, we can by strengthening the correlations between $g_B(x'_1)$ and $g^*(x''_1)$ and/or between $g_B(x'_1, x'_3)$ and $g^*(x''_1, x''_3)$ in (6.10). This kind of model gives us a high level of control in stating exactly how it is that B is better than A, because of the multiple but still clearly delineated paths from f_A and f_B to f^*. For example, we could model the situation in which f_B has a higher resolution than f_A, but f_A contains a subprocess that f_B ignores. This often happens in practice, where additional complexity can only be achieved by sacrificing resolution.

The inferential calculations may be computationally quite challenging, not least the accounting necessary to construct and use a belief model such as (6.10). However, the inferential procedure is unambiguous, and the meaning of the various uncertainty statements is clear. The emulators, the top simulator, and the direct simulator form the logical links that relate evaluations of our collection of actual simulators to the system and give meaning to the various uncertainties that separate our simulators from that system.

7. Discussion.

Computer simulators embodying complex mathematical models are increasingly the method of choice for studying large-scale physical systems. While this approach offers many opportunities, it is also open to gross abuse, unless we are very clear as to the limitations of such models when used as surrogates for the system itself. Explicit assessment of the difference between the system and models of the system is, therefore, of fundamental importance.

In this paper, we have outlined a general approach for structuring the uncertainties which arise in transferring inferences from models, typically computer simulators, to physical systems. To do this, we have introduced ingredients which go substantially beyond the kinds of reasoning that are currently offered to justify the relevance of computer-based analyses. Does this mean that our formulation is overelaborate?

We would argue quite the opposite, namely, that our specification has been stripped down to the barest minimum of ingredients which must be in place before we can even attempt the task of relating the model to the system. It is incumbent on the analyst either to make use of a formulation along the lines that we have suggested or to suggest an alternative logic for developing such relationships as may exist between the model analysis and the phenomena in question. However, we know of no systematic alternative approach which may form the basis of such a development.

We consider that there are two principle advantages inherent in using our structured uncertainty approach. The first advantage of the approach is in achieving consistency and accuracy in the specification of all the relevant aspects of uncertainty. It may be daunting for the analyst even to consider how a single model, with many unknown inputs and many outputs, may be linked to a corresponding physical system. Therefore, it is helpful to separate out, for careful, individual consideration, the various aspects of the relationship between the model and the system, within a fully coherent framework. Such representations are even more important for those problems which are informed by a wide variety of computer models. Seldom do the various models represent fully independent sources of information about the system, and it is essential to distinguish information that is common to several of the models from information that is specific to a particular model. We know of no other formulation which will allow us to do this to a high level of generality.

Second, the approach helps to achieve clarity. Model-based inference is not a private activity, and the objective of a scientific analysis is to make a clear and convincing case to the wider community concerning the behavior of the physical system to which each of the models relates. This is not possible unless the logic of the argument is made transparent, and this can only be achieved by attaching uncertainty statements to well-defined quantities, so that the meaning of each part of the analysis is clear, and carrying out a careful sensitivity analysis over each aspect of the specification, so that the degree to which the scientific conclusions depends on each of the underlying assumptions can be fully understood.

In conclusion, we would hope that forms of reasoning similar to those that we have described will be taken up in general as part of the standard methodology for reasoning about computer experiments and, in particular, in specific applications, where highly specialized versions of these structures may be constructed after careful and expert consideration. This points to both the need for a software interface to aid in building and analyzing such extended models, and also the value in developing elicitation tools to aid the experts in specifying beliefs over such elaborate constructions. For a discussion of aspects of the construction of such tools, see [7].

REFERENCES

[1] J. Bernardo and A. Smith, *Bayesian Theory*, John Wiley & Sons, Chichester, UK, 1994.

[2] D. B. Chelton, *Physical oceanography: A brief overview for statisticians*, Statist. Sci., 9 (1994), pp. 150–166.

[3] A. Colling, *Ocean Circulation*, Butterworth-Heinemann, Boston, 2001.

[4] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer-Verlag, New York, 1999.

[5] P. Craig, M. Goldstein, J. Rougier, and A. Seheult, *Bayesian forecasting for complex systems using computer simulators*, J. Amer. Statist. Assoc., 96 (2001), pp. 717–729.

[6] P. Craig, M. Goldstein, A. Seheult, and J. Smith, *Pressure matching for hydrocarbon reservoirs: A case study in the use of Bayes linear strategies for large computer experiments*, in Case Studies in Bayesian Statistics, Vol. 3, C. Gatsonis, J. Hodges, R. Kass,

R. McCulloch, P. Rossi, and N. Singpurwalla, eds., Lecture Notes in Statist. 121, Springer-Verlag, New York, 1997, pp. 37–93, with discussion.

[7] P. Craig, M. Goldstein, A. Seheult, and J. Smith, *Constructing partial prior specifications for models of complex physical systems*, The Statistician, 47 (1998), pp. 37–53, with discussion.

[8] N. A. C. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, New York, 1991.

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

[10] F. Jensen, *Bayesian Networks and Decision Graphs*, Springer-Verlag, New York, 2001.

[11] M. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. R. Stat. Soc. Ser. B Stat. Methodol., 63 (2001), pp. 425–464, with discussion.

[12] National Research Council (NRC), *Report on statistics and physical oceanography*, Statist. Sci., 9 (1994), pp. 167–221, with discussion.

[13] J. Peixoto and A. Oort, *Physics of Climate*, Springer-Verlag, New York, 1992.

[14] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, 1999.

[15] J. Sachs, W. Welch, T. Mitchell, and H. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–435, with discussion.

[16] G. Thomas, *Principles of Hydrocarbon Reservoir Simulation*, International Human Resources Development Corporation, Boston, 1982.

[17] K. Zickfeld, T. Slawig, and S. Rahmstorf, *A low-order model for the response of the Atlantic thermohaline circulation to climate change*, Ocean Dynamics, 54 (2004), pp. 8–26.