A Framework for Validation of Computer Models

Maria J. BAYARRI

Department of Statistics and O.R.
University of Valencia
46100 Burjassot, Valencia, Spain

James O. BERGER

Institute of Statistics and Decision Sciences
Duke University
Durham, NC 27708

Rui PAULO

Departamento de Matematica
ISEG, Technical University of Lisbon
1200-781 Lisboa, Portugal

Jerry SACKS

National Institute of Statistical Sciences
Research Triangle Part, NC 27709

John A. CAFEO, James CAVENDISH, Chin-Hsu LIN, and Jian TU

Research and Development
General Motors
Warren, MI 48090

We present a framework that enables computer model evaluation oriented toward answering the question: Does the computer model adequately represent reality? The proposed validation framework is a six-step procedure based on Bayesian and likelihood methodology. The Bayesian methodology is particularly well suited to treating the major issues associated with the validation process: quantifying multiple sources of error and uncertainty in computer models, combining multiple sources of information, and updating validation assessments as new information is acquired. Moreover, it allows inferential statements to be made about predictive error associated with model predictions in untested situations. The framework is implemented in a test bed example of resistance spot welding, to provide context for each of the six steps in the proposed validation process.

KEY WORDS: Bayesian analysis; Identifiability; Model discrepancy; Prediction.

1. INTRODUCTION

1.1 Overview

We view the most important question in evaluation of a computer model to be: Does the computer model adequately represent reality? An austere view expressed by Oreskes, Shrader-Frechette, and Belitz (1994) is that validating a computer model cannot be done and that the "primary value of models is heuristic: models are representations, useful for guiding further study but not susceptible to proof." This view has a substantial basis in purely scientific roles, as distinct from a model's use in policy and engineering contexts. But the real question, we contend, is not whether a model is absolutely correct or only a useful guide. Rather, it is to assess the degree to which it is an effective surrogate for reality: Does the model provide sufficiently accurate predictions for the intended use? As we clarify as we proceed, here we intend accuracy to refer to both possible bias in the model and possible uncertainty.

The question and attitude that we set out here are not new; they appear again and again in discussions and comments on validation in many arenas over the years, at least as long ago as the work of Caswell (1976). A detailed discussion of many issues surrounding validation has been given by Berk et al. (2002). But incisive argument on the validity of models, seen as assessment of their utility, has been hampered by the lack of structure in which quantitative evaluation of a model's performance can be addressed. It is our purpose here to explore structure and methodology to produce such evaluations.

Numerous other issues are crucial to computer model development and use of computer models (in, e.g., optimization). Indeed, in practice, the processes of computer model development and validation often occur in concert; aspects of validation interact with and feed back to development; for example, a shortcoming in the model uncovered during the validation process may require a change in the mathematical implementation. In this article, however, we address these other issues only to the extent to which they interact with the framework that we envision for answering the foregoing basic question. General discussions of the entire validation and verification process have been given by Roache (1998), Oberkampf and Trucano (2000), Cafeo and Cavendish (2001), Easterling (2001), Pilch et al. (2001), Trucano, Pilch, and Oberkampf (2002), and Santner, Williams, and Notz (2003).

The main goal of this article is to outline a step-by-step process to produce tolerance bounds that take into account the key uncertainties in the problem. This step-by-step process is illustrated on an engineering example involving spot welding. The inferential ideas behind the methodology were given by Kennedy and O'Hagan (2001) but are still not well understood. Hence the remainder of the section focuses on motivating the methodology and supplying background material. The methodology itself is presented in Sections 2–6, with illustrations using the test bed model. The presentation is designed so that both engineers and statisticians can follow the process before technical details, notation, and other topics are introduced. The reader

© 2007 American Statistical Association and
the American Society for Quality
TECHNOMETRICS, MAY 2007, VOL. 49, NO. 2
DOI 10.1198/004017007000000092

wishing to jump directly to the specifics of the methodology can focus on Sections 2.2 and 5.

1.2 Motivation for Tolerance Bounds

To motivate our approach to model evaluation, it is useful to begin at the end and consider the type of conclusions that will result from the methodology. As noted earlier, we focus not on answering the yes/no question "is the model correct?," but rather on assessing the accuracy of predictions in uses of the model. We do this by presenting *tolerance bounds*, such as 5.17 ± .44, for a model prediction of 5.17, with the interpretation that there is a specified chance (e.g., 90%) that the corresponding true process value would lie within the specified range. Such tolerance bounds should be given whenever predictions are made; that is, they should be routinely included along with any predictions arising from use of the model.

This focus on giving tolerance bounds rather than stating a yes/no answer as to model validity arises for three reasons:

- Models rarely give highly accurate predictions over the entire range of inputs of possible interest, and it is often difficult to characterize regions of accuracy and inaccuracy.
- The degree of accuracy needed can vary from one application of the computer model to another.
- Tolerance bounds account for model bias, the principal symptom of model inadequacy; accuracy of the model cannot be represented simply by a variance or standard error.

All of these difficulties are obviated by the simple step of routinely presenting tolerance bounds along with model predictions. Thus, at a different input value, the model prediction and tolerance bound might be 6.28 ± 1.6, and it is immediately apparent that the model is considerably less accurate at this input value than at the previous input value, for which the tolerance bound was ±.44. Either of the bounds, .44 or 1.6, might be acceptable or unacceptable, depending on the model's intended use.

Producing tolerance bounds is not easy. Here is a partial list of the hurdles:

- Uncertainties in model inputs or parameters can arise in several ways: based on data, on expert opinion, or simply on a prior "uncertainty range."
- When model runs are expensive, only limited model run data may be available.
- Field data of the actual process under consideration may be limited and noisy and may be of various types, including functional data.
- Model runs may be made at input values different from those at which field data are observed.
- One may desire to "tune" unknown parameters of the computer model based on field data and at the same time (because of sparse data) apply a validation methodology; sometimes there are even more tuning parameters than data.
- The computer model itself typically will be highly nonlinear and often will be biased, that is, will differ systematically from the real process.
- Validation should be viewed as an accumulation of evidence to support confidence in the model outputs and their use, and the methodology should allow updating of current conclusions as additional information arrives.

This article describes an approach that deals with these hurdles and, using a mix of Bayesian and likelihood techniques, can produce usable tolerance bounds for computer model predictions, thereby giving specific quantitative meaning to validation. Technical details of the approach are given in Section 5. The remainder of this section is given over to added discussion of validation and the approach that we recommend. Details addressing some of the listed hurdles are not given because they are not needed for the test bed problem. For example, uncertainties in inputs based on data are not addressed but can be accommodated in a straightforward way.

Implementation of the suggested methodology has the following added implications:

- The methodology allows explicit estimation of the bias of the model (together with the uncertainty in the bias) through comparison with field data. This allows direct judgment as to the model's validity in various regions of the input space. In addition, the methodology allows one to adjust the prediction by the estimated bias and provides tolerance bounds for this adjusted prediction. Depending on the size of the bias, this can result in considerably more accurate predictions than can be achieved using the model alone (or using field data alone). Note, however, that this adjustment might have limited utility in extrapolation to new situations, unless we are willing to make strong assumptions about how the bias extrapolates.
- Predictions and tolerance bounds can be given for applications of the computer model to new situations in which there are little, or no, field data, assuming information about "related" scenarios is available; this can be done through hierarchical Bayesian analysis. We do not address this issue in the current article; our predictions in the test bed example are made within the context of the given scenarios.
- Fast approximations to the computer code are used (typically, needed) for the proposed methods; these approximations have additional utility for use with complex computer codes in other contexts, such as in optimization.

1.3 Comparison With Statistical Model Validation

Computer model validation has developed with some significant differences from standard statistical model validation. It is useful to understand these differences and how they shape the particular methodology used for computer models.

Some of the methodology is driven by the unusual nature of many computer models: the very limited availability of data and the expense of running the computer model in particular. Indeed, it is not uncommon for computer models to take hours or days to run, so analysis of computer models often involves developing approximations to the model that can be used in the analysis.

The other major unusual aspect of the analysis of computer models is that it often focuses on what is to be done with a "rejected" computer model, rather than simply answering the question of whether or not the computer model is valid in the sense of hypothesis testing. It is useful to illustrate this key issue with a simple nonlinear regression example.

Pedagogic Example. At various times t_i, independent data were obtained from the nonlinear regression model

$$y(t_i) = g(t_i) + \epsilon_i, \qquad (1)$$

where the ϵ_i are independently $N(0, \sigma^2)$, with σ^2 unknown. Three independent replicate observations were obtained at each point on an equally-spaced grid of 10 values of t in the interval $(.11, 3.01)$. These data are given in Table 1.

The hypothesized regression model is

$$H_0: g(t) = 5\exp(-ut), \qquad (2)$$

with u unknown. (Suppose that the data are thought to have arisen from a chemical reaction process with initial chemical concentration 5 and reaction rate u; this is then a standard physical model for the amount of chemical remaining at time t.) One might begin the statistical analysis by finding the best fit of this function to the data; the maximum likelihood fit is at $\hat{u} = .63$, and the resulting function is graphed in Figure 1 along with the data.

Visually, it is clear that the function is not a good fit to the data, and formal statistical tests would agree. For instance, a modified F test rejects H_0 at a p value of .0004; even if transformed to an error probability scale as $-ep\log p = .0086$ (see Sellke, Bayarri, and Berger 2001), the indication is that there is indeed quite strong evidence against H_0.

A next step might be to look at the residuals from Figure 1 and seek missing structure. The residuals are graphed in Figure 2, along with a linear fit. It might be tempting to think that such additional structure found in the residuals is real, but a number of issues are involved with doing so:

- If the hypothesized model is incorrect, then "over-fitting" typically will have occurred; the fit attempts to make up for the model inadequacy by overshifting u to compensate.
- This overfitting makes it problematic to believe any structure found in the residuals (see, e.g., the linear structure in Fig. 2).
- Uncertainties must be taken into account when proceeding.

Table 1. Pedagogic Example, With Data Consisting of 3 Replicate Observations of the Regression Function (plus noise) at Each of 10 Times

t		$y(\cdot)$	
.110	4.730	4.720	4.234
.432	3.177	2.966	3.653
.754	1.970	2.267	2.084
1.077	2.079	2.409	2.371
1.399	1.908	1.665	1.685
1.721	1.773	1.603	1.922
2.043	1.370	1.661	1.757
2.366	1.868	1.505	1.638
2.688	1.390	1.275	1.679
3.010	1.461	1.157	1.530

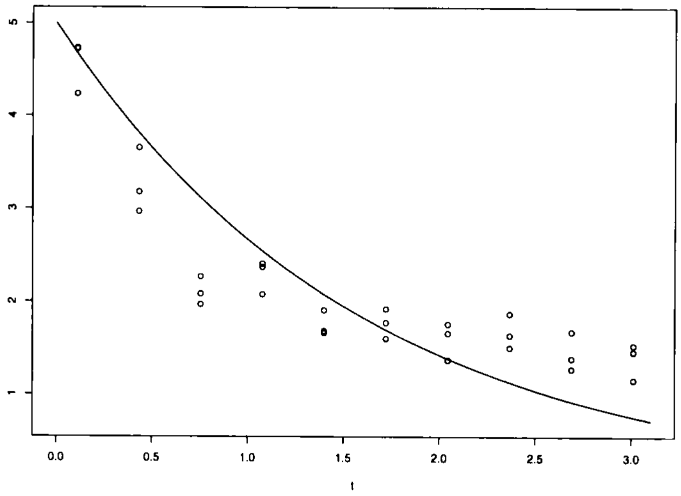

Figure 1. Maximum Likelihood Fit of Model (2) and Data for the Pedagogic Example.

At this point, it would be natural from a statistical standpoint to simply postulate a different form (perhaps nonparametric) for the unknown $g(t)$. In the computer model world [i.e., when $g(t)$ is the computer model itself] this is not directly possible, unless the analysis so far has suggested a possible improvement (which is subsequently implemented) in the computer model. There are several reasons for this, the most important of which is that the computer model is typically crucially needed for extrapolation beyond the range of the data; a purely statistical model is usually not trustworthy for such extrapolation.

The approach to dealing with this situation introduced by Kennedy and O'Hagan (2001) formally introduces a bias function, $b(t)$, so that the situation is modeled as

$$y(t_i) = 5\exp(-ut_i) + b(t_i) + \epsilon_i, \qquad (3)$$

where $b(t)$ is an unspecified function. One then jointly attempts to determine u and $b(\cdot)$; simultaneous inference can prevent the overfitting of u and can properly account for all uncertainties.

The major difficulty with this approach is a lack of identifiability of u and $b(\cdot)$ (and also of σ^2 when replicates at the design points t_i are not available). To see this, imagine that we had

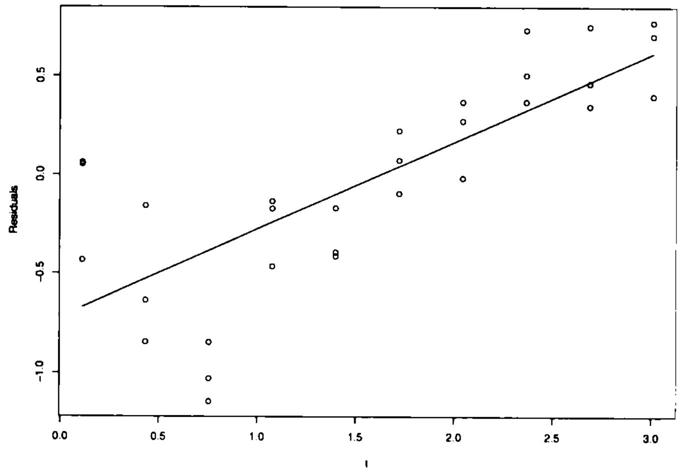

Figure 2. Residuals of the Fit to Model (2), and a Linear Fit to the Residuals.

an infinite amount of data at a dense set of t_i, so that we have completely observed the function $y(t)$ and know precisely that $y(t) = 5\exp(-ut) + b(t)$. Because $b(\cdot)$ is arbitrary, for each different u there is a $b(\cdot)$ that exactly fits the equation, so the two are severely confounded. (There seems to be some confusion about this issue; see the discussion in Kennedy and O'Hagan 2001.)

Dealing with such severe confounding is not commonly done in statistics and may seem unreasonable, yet here we have little choice; simply stopping the analysis with the statement "the computer model is rejected and must be improved" does not typically suffice in the computer modeling world. There is also growing understanding that bias and resulting confounding are more common in statistics than might be thought (see, e.g., Gustafson 2006 and the references therein).

The most straightforward method of handling such confounding is Bayesian analysis; one places a prior distribution on u and $b(\cdot)$, a distribution that contains as much expert knowledge as available. Some considerations here are as follows:

- In the computer model scenario, u may have physical meaning (e.g., the reaction rate in the example), or at least physical limits, so that experts may be able to construct a fairly tight prior distribution for u.
- The prior distribution typically "encourages" $b(\cdot)$ to be 0, allowing a correct computer model to emerge with little bias if supported by the data.

The posterior distributions of u and $b(\cdot)$ typically will be highly correlated and sensitive to the priors, but useful information as to the nature of the bias and its inherent uncertainty still can be obtained from the analysis. Some inferences, such as certain types of prediction, and uncertainty in prediction, seem to remain very stable over different prior specifications for the bias. We do not illustrate these inferences with this example, but rather use the spot weld test bed for illustration.

It should be noted that extrapolations to prediction in new regions (e.g., to $t > 3$ in the pedagogic example) will tend to be very sensitive to prior assumptions on u and $b(\cdot)$. There is no magic here; what the framework does is simply give a reasonable context in which to talk about such extrapolation.

As a postlogue to the pedagogic example, the true model used to generate the data was

$$y(t_i) = (3.5)\exp(-1.7t_i) + 1.5 + \epsilon_i, \tag{4}$$

with $\sigma^2 = (.3)^2$. Thus the actual bias function was $b(t) = 1.5[1 - \exp(-1.7t)]$. The "error" in the modeling of the chemical reaction was in not recognizing that there would be a residual of the chemical (here, 1.5 units) unreacted. Note that if we had used the best-fitting model, then the estimated reaction rate $\hat{u} = .63$ would be quite far from the true reaction rate $u = 1.7$, the overtuning that can result from not recognizing the existence of bias. (When the methodology proposed in this article was applied to the example, it yielded a posterior mean for u of 1.72. Although such a close match was undoubtedly just luck, given the lack of identifiability, this at least demonstrates the resistance of the methodology to overtuning.) Gustafson (2005) discussed nonidentifiability in other statistical situations and compared Bayesian and non-Bayesian methods of attempting to deal with the issue, indicating the considerable problems with the latter.

1.4 Background

The key components of the approach outlined here are the use of Gaussian process response-surface approximations to a computer model, following the work of Sacks, Welch, Mitchell, and Wynn (1989), Currin, Mitchell, Morris and Ylvisaker (1991), Welch et al. (1992), and Morris, Mitchell, and Ylvisaker (1993), and introduction of Bayesian representations of model bias and uncertainty, following the work of Kennedy and O'Hagan (2001) and Kennedy, O'Hagan, and Higgins (2002). The Gaussian process approximations have proven valuable in real settings where functions are complex and data limited (e.g., Gough and Welch 1994; Chapman, Welch, Bowman, Sacks, and Walsh 1994; Aslett, Buck, Duvall, Sacks, and Welch 1998) and their adoption here is both natural and convenient.

The approach taken here results in a computational burden that significantly increases with large numbers of model inputs, large numbers of unknown parameters, or large amounts of data (model run or field). Hence a primary concern is to focus on methods that have the potential for significant scale-up. Thus, as described in the companion article (Higdon, Kennedy, Cavendish, Cafeo, and Ryne 2004), a fully Bayesian approach to the problem was originally developed, but this has difficulties in appropriately scaling up and also requires considerable expertise in Markov chain Monte Carlo (MCMC) computation. Hence we have focused instead on simplifications such as "modularity" (analyze components of the problem separately to the extent possible) and the use of maximum likelihood or other methods to reduce the computational burden and allow the Bayesian part of the analysis to be stable.

Validation is an intrinsically hard statistical problem, and analyses that produce tolerance bounds for computer model predictions in complex situations can require considerable additional methodological development. Two such extensions of the methodology to functional data have been considered by Bayarri et al. (2006a, b). These extensions also consider uncertainty in the computer model inputs. Other extensions for dealing with high-dimensional output data include Higdon, Gattiker, and Williams (2005), who used principal components; Schmidt and O'Hagan (2003), who used a singular value decomposition; Lee, Higdon, Bi, Ferreira, and West (2002), who used a Cholesky decomposition with pivoting; Higdon (2002), who used a spatial moving average; and Lee, Higdon, Calder, and Holloman (2005), who used convolutions of Markov random fields with smoothing kernels. Generalizations to nonstationary scenarios have been addressed by Gramacy, Lee, and Macready (2004), and dynamic emulators were considered by Conti, Anderson, O'Hagan, and Kennedy (2005).

A related approach to Bayesian analysis of computer models is that of Craig, Goldstein, Seheult, and Smith (1997), Craig, Goldstein, Rougier, and Seheult (2001) and Goldstein and Rougier (2003, 2004), which focuses on using linear Bayes methodology to address the problem. Another significant body of work in computer modeling is that addressing the importance and uncertainty of input variables and/or the corresponding output distributions (propagation of error) (e.g., Saltelli, Chan, and Scott 2000; Oakley and O'Hagan 2002, 2004; Oakley 2004). Of course, the propagation of error issue appears in a host of scientific applications, a notable recent one being that of Stainforth et al. (2005).

1.5 Test Bed

The test bed provides a context for implementing each step of the framework and also prompts consideration of various issues. It is an application drawn from engineering practice.

Test Bed: The Spot Weld Example. In resistance spot welding, two metal sheets are compressed by water-cooled copper electrodes under an applied load, L. Figure 3 is a simplified representation of the spot welding process, illustrating some of the essential features for producing a weld. A direct current of magnitude C is supplied to the sheets by two electrodes to create concentrated and localized heating at the interface where the two sheets have been pressed together by the applied load (the so-called "faying surface"). The heat produced by the current flow across the faying surface leads to melting, and, after cooling, a weld "nugget" is formed.

The resistance offered at the faying surface is particularly critical in determining the magnitude of heat generated. Because contact resistance at the faying surface, as a function of temperature, is poorly understood, a nominal function is specified and "tuned" to field data. The effect of this tuning on the behavior of the model is the focus of the example.

The physical properties of the materials will change locally as a consequence of local increases in temperature. Young's modulus and the yield stress of the sheet will fall (i.e., the metal will "soften"), resulting in more deformation and an increased size of the faying contact surface, further affecting weld formation. At the same time, the electrical and thermal conductivities will decrease as the temperature rises, all of which will affect the rate of heat generation and removal by conduction away from the faying surface.

The thermal/electrical/mechanical physics of the spot welding process are modeled by a coupling of partial differential equations that govern heat and electrical conduction with those that govern temperature-dependent, elastic/plastic mechanical deformation (Wang and Hayden 1999). Finite-element implementations are used to provide a computer model of the electrothermal conceptual model. Similarly, a finite-element implementation is made for the equilibrium and constitutive equations that compose the conceptual model of mechanical/thermal deformation. These two computer models are implemented using a commercial code (ANSYS). Key inputs of the model are summarized in Table 2. Interesting outputs are given in Section 2.2.

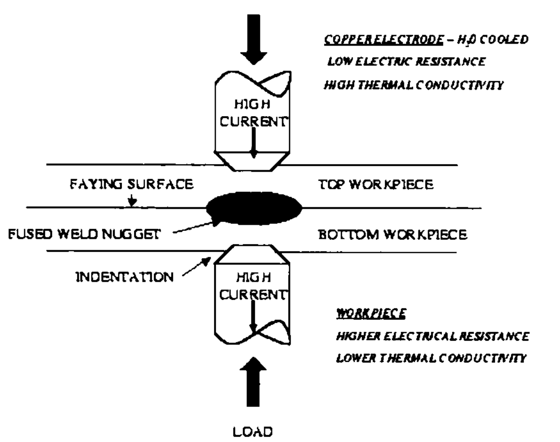

Figure 3. Schematic Representation of the Spot Welding Process.

2. THE MODEL AND ITS USES (STEPS 1 AND 2)

Understanding the uncertainties associated with the computer model and how the model is used are initial steps in the validation process.

2.1 Step 1: Specify Model Inputs and Parameters With Associated Uncertainties or Ranges. The Input/Uncertainty Map

A convenient way to organize information about inputs and their uncertainties is through what we call the input/uncertainty (I/U) map. [This is related to the idea of a PIRT (Phenomena Identification and Ranking Table); see Pilch et al. 2001.] The map has four attributes:

- A list of model features or inputs of potential importance
- A ranking of the importance of each input
- Uncertainties, either distributions or ranges of possible values, for each input
- Current status of each input, describing how the input is currently treated in the model.

The I/U map is dynamic; as information is acquired and the validation process proceeds, the attributes (especially the second, third, and fourth ones) may change or require updating. The inputs are drawn from the development process and will include parameters inherent to the scientific/engineering assumptions, the mathematical implementation, and the numerical parameters associated with the implementing code. In short, the inputs are the ingredients necessary to make the model run. Because this list can be enormous, more important parameters must be singled out to help structure the validation process by providing a sense (albeit imperfect) of priorities. We adopt a scale of 1–5 for ranking the inputs, with 1 indicating only a minor likely impact on prediction error and 5 indicating a significant potential impact.

Spot Weld. The purpose of the spot welding model is to investigate the process parameters for welding aluminum. The I/U map of the model is given in Table 2. The list of inputs in Table 2 was more fully described by Bayarri et al. (2002). Initially, only three inputs have rank 5 based on the model developer's assessment. These three parameters (and gauge) are the focus of the validation experiments; earlier experiments by the model developer led to the impact assessments appearing in the table. The specified ranges of the controllable parameters (current, load, and gauge) are given in step 2. No uncertainty about these inputs, either in the computer model or in the laboratory data collected for the validation exercise, is assumed. (In contrast, if validation of the model were required at the production level, then uncertainties in current and load might be significant; the I/U map is context-dependent.)

Several items connected with the I/U map in Table 2 are worth noting. First, the most significant specified uncertainty (impact factor 5) in the model is contact resistance. The model incorporates contact resistance through an equation that for the faying surface has a multiplicative constant u about which it is

Table 2. The I/U Map for the Spot Welding Model

Input		*Impact*	*Uncertainty*	*Current status*
Geometry	Electrode symmetry-2d	3	Unspecified	Fixed
	Cooling channel	1	Unspecified	Fixed
	Gauge	Unclear	Unspecified	1, 2 mm
Materials		Unclear	Aluminum (2 types ×2 surfaces)	Fixed
Stress/ strain	Piecewise linear	4	Unspecified (worse at high T)	Fixed
	C_0, C_1, σ_s	3	Unspecified	Fixed
Contact resistance	$1/\sigma = u \cdot f$; f fixed	3	Unspecified	Fixed by modeler
	$u = 0$ for electrode/sheet $u =$ tuning for faying	5	$u \in [0.8, 8.0]$	Tuned to data for 1 metal
Thermal conductivity κ		2	Unspecified	Fixed
Current		5	No uncertainty	Controllable
Load		5	No uncertainty	Controllable
Mass density (ρ)		1	Unspecified	Fixed
Specific heat (c)		1	Unspecified	Fixed
Numerical parameters	Mesh	1	Unspecified	Convergence/speed compromise
	M/E coupling time	1	Unspecified	
	Boundary conditions	1	Unspecified	Fixed
	Initial conditions	1	Unspecified	Fixed

only known that u lies in the interval [.8, 8.0]. It will be necessary to tune this parameter of the model with field data. The second most significant uncertainty in the model (impact factor 4) is the linear approximation for stress/strain. Because the modeler is unable to specify the uncertainty regarding this input, error in this input will simply enter into the overall unknown (and to be estimated) bias of the model.

Initial impact assessments are based on experience to reflect a combined judgment of the inherent sensitivity of the input (the extent to which small changes in the input would affect the output) and the range of uncertainty in the input. These may be revised through sensitivity analyses and "tuning with data" that occur later in the process. Inputs about which we are "clueless" might be singled out for attention at some point along the validation path, but the effect of "missing" inputs (i.e., nonmodeled features) may never be quantifiable or may emerge only after all effects of "present" inputs are accounted for.

In model validation, attention may need to be paid to the numerical accuracy of the implemented model, for instance, in assessing whether numerical solvers and finite-element codes have "converged" to the solution of the driving differential equations. This can be important and, as detailed by Cafeo and Cavendish (2001), is an issue of model and code verification. Ideally, numerical accuracy should be addressed early in the model development process and before the validation activity emphasized in this article. It is often the case that convergence is not obtained, however; for example, modelers may simply use the finest mesh size that is computationally feasible, even if it is insufficient for assuring convergence. The method that we propose for validation still works: the error introduced by a lack of convergence becomes part of the "bias" of the model that is to be assessed (see Sec. 5). The I/U map should of course clearly indicate the situation involving such convergence. The possible confounding effect of parameters, such as grid size, on other assumptions about the model will make improving the model more difficult. Ideally, identifying this effect could be done through designed experiments, varying values of the numerical parameters to assess numerical accuracy.

2.2 Step 2: Determine Evaluation Criteria

Evaluation of a model depends on the context in which it is used. Key elements of evaluation are as follows:

- Specification of an evaluation criterion (or criteria) defined on model output
- Specification of the domain of input variables over which evaluation is sought.

Even if only one evaluation criterion is initially considered, other evaluation criteria inevitably emerge during the validation process. The overall performance of the model may then depend on the outcomes of the validation process for several evaluation criteria (the model may fail for some and pass for others), leading ultimately to follow-on analyses about when and how the model should be used in prediction.

Informal evaluations (i.e., does the computer model produce results that appear consistent with scientific and engineering intuition) are typical during the development process. Later in the validation process these informal evaluations may need to be quantified and incorporated in the formal process. Sensitivity analyses may in some respects be considered part of the evaluation if, for example, the sensitivities confirm (or conflict with) scientific judgment.

Spot Weld. Two evaluation criteria were initially posed:

1. Size of the nugget after eight cycles

2. Size of the nugget as a function of the number of cycles.

Criterion 1 is of interest because of the model's primary production use; criterion 2 is of interest as a possible aid in reducing the number of cycles to achieve a desired nugget size. Ideally, the evaluation would be based directly on the strength of the weld, but weld diameter is taken as a surrogate because of the feasibility of collecting laboratory data on the latter. (Of course, if nugget size were not strongly correlated with weld strength, then these criteria probably would be inappropriate.) In production, the spot welding process results in a multiple set of welds, but the evaluation criterion considered here involves only a single weld. Criterion 2 was later discarded as a result of the difficulty during data collection of getting reliable computer runs producing output at earlier times than eight cycles.

The feasible domains of the input variables were specified as follows:

- Material: aluminum 5182-O and aluminum 6111-T4
- Surface: treated or untreated
- Gauge (mm): 1 or 2
- Current (kA): 21–26 for 1-mm aluminum; 24–29 for 2-mm aluminum
- Load (kN): 4.0–5.3.

Material and surface enter the model through other input variables relating to properties of materials. The initial specification in Table 1 views material and surface as fixed. The tuning parameter, u, has the range indicated in Table 1 and is the only other input not fixed.

3. DATA COLLECTION (STEP 3)

Both computer and field (laboratory or production) experiments are part of the validation and development processes and produce data essential for the following functions:

- Developing needed approximations to (expensive) numerical models
- Assessing bias and uncertainty in model predictions
- Studying sensitivity of a model to inputs
- Identifying suspect components of models
- Designing and collecting data that build on and augment existing or historical data.

The iterative and interactive nature of the validation and development processes will result in multiple stages of computer experiments and even field experiments.

Intuitively, designs should cover the ranges of the key input values, and "space-filling" strategies can be devised to accomplish this in an effective way (Sacks et al. 1989; Bates, Buck, Riccomagno, and Wynn 1996). The specific strategy we use is to select a latin hypercube design (LHD) minimizing $\max_{\text{LHD}} \min_{i,j} \delta(z_i, z_j)$, where δ is Euclidean distance. We use code from W. Welch to produce such designs.

Spot Weld. The inputs to be varied were $C =$ current, $L =$ load, $G =$ gauge, and the unknown tuning parameter u; the other inputs were held fixed. The cost (30 minutes per computer run) is high, so a limited number (26) of runs were planned for each of the 2 gauge sizes. The 26 runs for 1-mm metal covered the three-dimensional rectangle, $[20, 27] \times [3.8, 5.5] \times [1] \times [1.0, 7.0]$, in the (C, L, G, u) space, whereas those for the 2-mm metal covered the three-dimensional rectangle, $[23, 30] \times [3.8, 5.5] \times [2] \times [.8, 8.0]$. The explicit values of the 26-point maximin LHDs, along with the resulting model output for the nugget diameter, are given in Table 3. The computer runs exhibited some aberrant behavior. Many (17) runs failed to produce a meaningful outcome at cycle 8; these runs were eliminated. For reasons that are not yet clear, many runs were unable to produce reliable data for earlier cycle times; as a result, evaluation criteria depending on early cycle times were abandoned. The data retained (35 runs) are used in the subsequent analyses.

Field data usually will be harder to obtain than computer experimental data and, as in the spot welding example, are often a result of other experiments not designed for the validation study. Typically, field data will depend crucially on the specifications in Section 2.2 and what can be feasibly obtained; specific design strategies usually seem to have little affect. The field data for the test bed are as follows.

Spot Weld. The field data for spot weld are given in Table 4. They were obtained by physical experimentation, the details of which make reasonable the assumption that the measurement errors are independent normal with mean 0 and unknown variance.

Note that replicated data were available at the various input values. Having such replicate data is highly desirable, in that doing a reasonable job of pinning down the measurement error variance makes the validation analysis considerably more accurate.

4. MODEL APPROXIMATION (STEP 4)

4.1 Introduction

Unless the computer model code is very cheap to run, using the code directly to perform the validation analysis is difficult, because validation (see Sec. 5) typically requires many code evaluations. Thus it is common to use approximations to the computer model—based on a limited number of runs—for validation. There are other reasons for desiring such approximations, such as ease of use "in the field" (compared with use of the original code), in optimization (where typical algorithms may again require many evaluations of the code), and in "output analysis" (i.e., analysis of the sensitivity of outputs to inputs or analysis of output distributions based on random inputs.)

A very useful general tool for models whose output depends smoothly on inputs (very common in engineering and scientific processes) is the Gaussian process response surface technique (GASP) advanced by Sacks et al. (1989) and frequently used subsequently (Currin et al. 1991; Morris et al. 1993; Kennedy and O'Hagan 2001; Santner et al. 2003; Higdon et al. 2004). This technique meshes well with the validation analysis proposed in step 5.

More formally, denote model output by $y^M(\mathbf{x}, \mathbf{u})$, where $\mathbf{x}$ is a vector of controllable inputs and $\mathbf{u}$ is a vector of unknown calibration and/or tuning parameters in the model. The goal is to approximate $y^M(\mathbf{x}, \mathbf{u})$ by a function $\hat{y}^M(\mathbf{x}, \mathbf{u})$ that is easy to compute. In addition, it is desirable to have a variance function $V^M(\mathbf{x}, \mathbf{u})$ that measures the accuracy of $\hat{y}^M(\mathbf{x}, \mathbf{u})$. We now turn to the details of how the GASP approach achieves these goals.

Table 3. Spot weld Data From 52 Model Runs

Gauge (mm)	*u (–)*	*Load (kN)*	*Current (kA)*	*Nugget diameter (mm)*	*Gauge (mm)*	*u (–)*	*Load (kN)*	*Current (kA)*	*Nugget diameter (mm)*
1	6.52	4.072	26.44	–	2	4.544	3.936	27.76	7.15
1	4.60	4.684	21.68	5.64	2	5.696	4.14	25.52	6.39
1	3.64	5.024	23.64	–	2	1.088	4.684	28.32	6.38
1	7.00	4.412	23.36	–	2	0.8	4.276	24.40	4.87
1	6.76	4.888	25.04	–	2	3.68	4.412	26.08	6.47
1	1.00	4.82	22.52	4.36	2	4.832	4.616	23.00	6.68
1	3.40	4.616	27.00	–	2	7.136	4.344	27.20	6.71
1	5.32	4.48	20.84	6.12	2	4.256	5.228	24.68	6.54
1	2.92	5.092	20.56	5.00	2	3.392	4.004	23.28	5.97
1	1.48	5.364	21.12	4.53	2	1.952	4.48	23.84	5.72
1	2.20	4.004	21.40	5.20	2	2.528	3.8	24.96	6.23
1	2.68	4.344	25.88	–	2	2.24	4.208	29.72	–
1	2.44	5.50	23.08	–	2	1.376	5.024	25.80	5.46
1	4.36	3.80	25.32	–	2	7.424	4.072	28.88	–
1	1.24	4.208	24.76	6.06	2	6.272	4.548	29.16	7.36
1	6.04	4.752	20.00	–	2	6.848	5.364	23.56	–
1	5.56	5.432	25.60	–	2	3.968	4.888	29.44	7.16
1	1.96	4.956	26.16	6.69	2	3.104	5.432	28.60	6.61
1	5.80	3.936	23.92	7.17	2	5.12	5.5	26.64	5.98
1	4.84	4.14	22.80	–	2	6.56	3.868	26.36	6.74
1	3.16	3.868	22.24	5.71	2	5.984	4.956	24.12	5.32
1	6.28	5.228	21.96	5.38	2	8	5.092	28.04	–
1	1.72	4.548	24.20	5.85	2	2.816	4.82	26.92	6.70
1	5.08	5.16	26.72	–	2	5.408	5.16	30.00	–
1	4.12	5.296	24.48	6.87	2	1.664	5.296	27.48	6.02
1	3.88	4.276	20.28	4.91	2	7.712	4.752	25.24	5.50

NOTE: Run failures are indicated by –.

4.2 The GASP Response-Surface Methodology

Let $\mathbf{y}^M = (y^M(\mathbf{x}_1, \mathbf{u}_1), \dots, y^M(\mathbf{x}_m, \mathbf{u}_m))$ denote the vector of m evaluations of the model at inputs $D^M = \{(\mathbf{x}_i, \mathbf{u}_i) : i = 1, \dots, m\}$ and write $\mathbf{z} = (\mathbf{x}, \mathbf{u})$. The computer model is exercised only at the inputs D^M, so that $y^M(\mathbf{z})$ is effectively unknown for other inputs $\mathbf{z} \notin D^M$. Before seeing $\mathbf{y}^M$, we assign $y^M(\cdot)$ a prior distribution, specifically a stationary Gaussian process with mean and covariance functions governed by unknown parameters $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$. (In essence, we are assuming that the output of the code at any finite number of locations has a multivariate normal distribution.)

The mean function of the Gaussian process is assumed to be of the form $\boldsymbol{\Psi}'(\cdot)\boldsymbol{\theta}^L$, where $\boldsymbol{\Psi}(\mathbf{z})$ is a *specified* $k \times 1$ vector function of the input $\mathbf{z}$ and $\boldsymbol{\theta}^L$ is a $k \times 1$ vector of unknown parameters. A constant mean [$k = 1$, $\Psi(\mathbf{z}) = 1$, and $\boldsymbol{\theta}^L = \theta$] is often satisfactory if one plans to use the model approximation only within the range of the available model run data. A more complicated mean function can be useful if the model approximation is to be used outside the range of the data because, outside of this range, the Gaussian process approximation to the model will gradually tend toward its estimated mean function. This can be especially important when such features as temporal trends are present.

The parameter λ^M is the precision (the inverse of the variance) of the Gaussian process, and the other parameters $(\boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$ control the correlation function of the Gaussian process, which we assume to be of the form

$$c^M(\mathbf{z}, \mathbf{z}^\star) = \exp\left(- \sum_{j=1}^{d} \beta_j^M |z_j - z_j^\star|^{\alpha_j^M} \right). \qquad (5)$$

Here d is the number of coordinates in $\mathbf{z}$, the α_j^M's are numbers between 1 and 2, and the β_j^M's are positive parameters. The product form of the correlation function (each factor is itself a correlation function in one-dimension) aids later computations.

Table 4. Spot Weld Example, With Field Data Consisting of 10 Replicate Observations of Nugget Size at Each of 12 Input Values

L	*C*	*G*					*$y^F(\cdot)$*					
4.00	21.0	1	4.81	5.08	5.09	4.84	5.40	5.14	4.92	5.31	4.95	4.80
4.00	23.5	1	5.31	6.52	5.89	5.51	5.77	4.96	5.04	5.22	5.54	6.36
4.00	26.0	1	5.52	6.62	5.97	5.76	6.13	5.82	5.81	6.00	6.00	6.52
5.30	21.0	1	5.09	4.43	4.63	5.01	5.07	4.14	4.03	4.30	4.09	4.02
5.30	23.5	1	5.11	5.17	5.71	5.60	5.85	4.60	5.51	4.82	6.37	5.23
5.30	26.0	1	5.34	5.19	5.86	5.94	5.98	5.09	5.43	5.14	5.21	5.73
4.00	24.0	2	6.78	5.89	6.49	6.78	6.81	7.00	7.16	6.68	6.68	6.98
4.00	26.5	2	6.62	6.54	6.30	6.00	6.67	6.89	7.15	5.99	5.90	7.29
4.00	29.0	2	7.28	6.98	7.46	7.87	8.02	6.97	8.15	7.14	7.55	7.75
5.30	24.0	2	6.62	6.74	6.59	6.39	6.45	6.64	5.59	6.30	5.64	6.05
5.30	26.5	2	7.25	6.80	6.50	6.36	7.67	7.14	5.95	7.10	7.57	7.08
5.30	29.0	2	7.62	7.71	8.14	7.26	8.37	7.68	6.95	6.41	8.35	7.50

Prior beliefs about the smoothness properties of the function will affect the choice of $\boldsymbol{\alpha}^M$. The choice of $\alpha_j^M = 2$ for all j reflects the belief that the function is infinitely differentiable, which is plausible for many engineering and scientific models.

This can be summarized by saying that, given the hyperparameters $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$, the prior distribution of y^M is $\mathrm{GP}(\boldsymbol{\Psi}'(\cdot)\boldsymbol{\theta}^L, \frac{1}{\lambda^M}c^M(\cdot,\cdot))$, that is, a Gaussian process with the given mean and covariance functions. As before, let $\mathbf{y}^M$ denote the vector of model evaluations at the set of inputs D^M. Conditionally on the hyperpartameters, $\mathbf{y}^M$ is a priori multivariate normal with covariance matrix $\boldsymbol{\Gamma}^M = \mathbf{C}^M(D^M, D^M)/\lambda^M$, where $\mathbf{C}^M(D^M, D^M)$ is the matrix with (i,j) entry $c^M(\mathbf{z}_i, \mathbf{z}_j)$, for $\mathbf{z}_i, \mathbf{z}_j$ in D^M.

After observing $\mathbf{y}^M$, the conditional posterior distribution of y^M given the hyperparameters, $p(y^M(\cdot)|\mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, is a Gaussian process with updated mean and covariance functions given by

$$\mathrm{E}[y^M(\mathbf{z})|\mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M] = \boldsymbol{\Psi}'(\mathbf{z})\boldsymbol{\theta}^L + \mathbf{r_z}'(\boldsymbol{\Gamma}^M)^{-1}(\mathbf{y}^M - \mathbf{X}\boldsymbol{\theta}^L) \quad (6)$$

and

$$\mathrm{cov}\big[y^M(\mathbf{z}), y^M(\mathbf{z}^\star)|\mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M\big]$$
$$= \frac{1}{\lambda^M}c^M(\mathbf{z}, \mathbf{z}^\star) - \mathbf{r_z}'(\boldsymbol{\Gamma}^M)^{-1}\mathbf{r}_{z^\star}, \quad (7)$$

where $\mathbf{r_z}' = \frac{1}{\lambda^M}\,(c^M(\mathbf{z}, \mathbf{z}_1), \ldots, c^M(\mathbf{z}, \mathbf{z}_m))$, $\boldsymbol{\Gamma}^M$ is as given earlier, and $\mathbf{X}$ is the matrix with rows $\boldsymbol{\Psi}'(\mathbf{z}_1), \ldots, \boldsymbol{\Psi}'(\mathbf{z}_m)$.

With specifications for $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$, the GASP behaves as a Kalman filter, yielding a posterior mean function (6) that can be used as the fast approximation or inexpensive emulator for $y^M(\cdot)$. Thus [given $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$], the response surface approximation to $y^M(\mathbf{z})$ at any point $\mathbf{z}$ is simply $\mathrm{E}[y^M(\mathbf{z})|\mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M]$ given by (6), and the variance measuring the uncertainty in this approximation is, following (7), $\mathrm{var}[y^M(\mathbf{z})|\mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M] = 1/\lambda^M - \mathbf{r}'_\mathbf{z}(\boldsymbol{\Gamma}^M)^{-1}\mathbf{r_z}$. Note that the variance is 0 at the design points at which the function was actually evaluated.

The hyperparameters $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$ are typically unknown. Two possibilities then arise:

- Plug in some estimates in the foregoing formulas, for instance maximum likelihood estimates (MLEs), as in the GASP software of W. Welch, pretending that they are the "true" values. For MLE estimates $(\hat{\boldsymbol{\theta}}^L, \hat{\boldsymbol{\theta}}^M)$, this produces the following model approximation for input $\mathbf{z}$:

  $$\hat{y}^{\mathrm{MLE}}(\mathbf{z}) = \boldsymbol{\Psi}'(\mathbf{z})\hat{\boldsymbol{\theta}}^L + \hat{\mathbf{r}}'_\mathbf{z}(\hat{\boldsymbol{\Gamma}}M)^{-1}(\mathbf{y}^M - \mathbf{X}\hat{\boldsymbol{\theta}}^L),$$

  where $\hat{\boldsymbol{\theta}}^M = (\hat{\lambda}^M, \hat{\boldsymbol{\alpha}}^M, \hat{\boldsymbol{\beta}}^M)$ is used to compute $\hat{\boldsymbol{\Gamma}}^M$ and $\hat{\mathbf{r}}_\mathbf{z}$. Similarly, $\mathrm{var}[y^M(\mathbf{z})|\mathbf{y}^M, \hat{\boldsymbol{\theta}}^L, \hat{\boldsymbol{\theta}}^M] = 1/\hat{\lambda}^M - \hat{\mathbf{r}}'_\mathbf{z}\,(\hat{\boldsymbol{\Gamma}}^M)^{-1}\hat{\mathbf{r}}_\mathbf{z}$ is used as the estimate of the approximation variance. This results in an underestimate of the true variability, because the uncertainty in the estimates of $\hat{\boldsymbol{\theta}}^L$ and $\hat{\boldsymbol{\theta}}^M$ is not taken into account, although the prediction variance can be adjusted using standard estimates of this uncertainty.
- Integrate the hyperparameters with respect to the posterior distribution in a full Bayesian analysis (as detailed in Paulo 2005), leading to a more appropriate approximation: the integral of (6) with respect to the posterior distribution of $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$, $p(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M|\mathbf{y}^M)$. This is done in practice by using MCMC techniques to generate a sample of size N, $\{(\boldsymbol{\theta}^{L(i)}, \boldsymbol{\theta}^{M(i)})\}$ from this posterior distribution, evaluating (6) at these generated values, and averaging. The variance of this approximation is obtained by adding two terms: the posterior expectation of (7) and the posterior variance of (6). In practice, these terms are estimated by the sample average of (7) and by the sample variance of (6) evaluated at the generated values $(\boldsymbol{\theta}^{L(i)}, \boldsymbol{\theta}^{M(i)})$. Alternatively, one may wish to draw realizations from the marginal posterior of $y^M(\mathbf{z})$, $p(y^M(\mathbf{z})|\mathbf{y}^M)$ directly and then compute appropriate summary statistics. This can be done in practice for each generated value $(\boldsymbol{\theta}^{L(i)}, \boldsymbol{\theta}^{M(i)})$ by computing (6) and (7) and then drawing a normal random variable with mean and variance given by these numbers.

Spot Weld. The vector of controllable inputs is $\mathbf{x} = (C, L, G)$, and the tuning parameter is u. Using a GASP full Bayesian analysis with the data from Table 3 leads to the response surface approximation to $y^M(C, L, G, u)$ shown in figure 8 of Higdon et al. (2004). The MLE approximation is very similar and hence is omitted here.

4.3 Maximum Likelihood Estimate Plug-in or Full Bayes?

The full Bayesian analysis is theoretically superior, because the resulting variance takes into account the uncertainty in the GASP parameters. When the function being approximated is very smooth, the additional uncertainty is not really needed, but it could be relevant when approximating less smooth models. The advantage of using the MLE plug-in approach is computational; implementing a GASP with fixed parameters is easier than averaging GASPs over a posterior sample of parameters.

The primary focus in this article is not in model approximation itself, but in the validation/prediction analysis discussed in the next section. In such analyses, we have found that using the MLEs of the GASP parameters typically yields much the same answers as the full Bayesian analysis, at least when tuning/calibration parameters are present in the computer model. The reason for this is that the uncertainty in calibration and tuning parameters, together with the uncertainty in the "bias" of the computer model, tend to overwhelm the uncertainty in the model approximation. Hence our current (cautious) recommendation is to use MLE plug-in GASPs, together with Bayesian analysis of the validation/prediction process. This allows implementation of the validation methodology in vastly more complicated scenarios (Bayarri et al. 2006b) than would otherwise be possible.

5. ANALYSIS OF MODEL OUTPUT (STEP 5)

In this section we describe the structure (statistical model) and analysis that we use for computer model evaluation, and illustrate the methods using the test bed example. Some technical details that threaten to cloud the exposition are relegated to the appendixes. We begin by describing the statistical structure and necessary notation. In Section 5.2 we address computation of the posterior distributions, predictions, and tolerance bounds, the heart of the matters at hand.

5.1 Notation and Statistical Modeling

The computer model approximates reality, and the discrepancy between the model and reality is the model bias. Accounting for this bias is the central issue for validation. There are (at least) three possible sources for this bias:

- The science or engineering used to construct the model is incomplete.
- Calibrated/tuned parameters may be in error.
- Numerical implementation may introduce errors (e.g., may not have converged).

The first two sources are typical; the third occurs with some frequency.

The computer model alone cannot provide evidence of bias. Either expert opinion or field data are needed to assess bias; here we focus on the latter. If field data are unavailable (even from experiments involving related models), then strict model validation is impossible. Useful things may still be said, but the ultimate goal of confirming accuracy of predictions is not attainable.

Recall that $y^M(\mathbf{x}, \mathbf{u})$ denotes the model output when $(\mathbf{x}, \mathbf{u})$ is input. When $\mathbf{u}$ is not present, we formalize the statement "reality = model + bias" as

$$y^R(\mathbf{x}) = y^M(\mathbf{x}) + b(\mathbf{x}), \tag{8}$$

where $y^R(\mathbf{x})$ is the value of the "real" process at input $\mathbf{x}$ and $b(\mathbf{x})$ is the (unknown) bias function. When $\mathbf{u}$ is present as a calibration parameter, we call its true (but unknown) value $\mathbf{u}_\star$, and then bias is defined through

$$y^R(\mathbf{x}) = y^M(\mathbf{x}, \mathbf{u}_\star) + b_{\mathbf{u}_\star}(\mathbf{x}). \tag{9}$$

In situations where $\mathbf{u}$ is viewed as simply a tuning parameter, there is no "true value," so $\mathbf{u}_\star$ should be thought of as some type of best-fitting value of $\mathbf{u}$, with the bias defined relative to this. Note that there is confounding between $\mathbf{u}_\star$ and the bias function, that is, they are not statistically identifiable. This important issue was discussed in Section 1.3 and has profound implications for the possible types of analysis; in particular, the natural way to deal with a lack of identifiability is to use prior information to provide identification or at least use Bayesian analysis to properly account for the uncertainty caused by the nonidentifiability. For notational convenience, we often drop the dependence of b on the true value of the calibration parameter.

Field data at inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are assumed to be "reality" measured with error. Specifically,

$$y^F(\mathbf{x}_i) = y^R(\mathbf{x}_i) + \epsilon_i^F, \tag{10}$$

where the ϵ_i^F are independent normal random errors with mean 0 and variance $1/\lambda^F$. This equation may be reasonable only after suitable transformation of the data, and often more complicated error structures (such as correlated errors) are needed; these typically can be accommodated with some additional computational effort. Note that $\mathbf{u}$ is not an input in determining the field data.

The assumption that ϵ^F has mean 0 implies no bias in the field measurements; that is, the measurement process is "well calibrated." Otherwise, the situation is problematic; the estimated bias will be a combination of both model and field bias, and there is no data-based way to separate the two. Additional insight or expert opinion is required be necessary to permit such separation. Unfortunately, it is quite common for "existing field data" (e.g., historical data, data acquired for different purposes but now used for validation) to be biased (see, e.g., Roache 1998), so obtaining unbiased field data may be challenging in its own right (see Trucano et al. 2002 for further discussion).

For the Bayesian model to be complete, the priors must be specified for the unknowns: $\mathbf{u}$, λ^F, and $b(\mathbf{x})$. These are chosen as follows:

- $p(\mathbf{u})$ is specified in the I/U map; it is often uniform on a given range.
- $p(\lambda^F)$ is exponential (see Sec. A.3).
- The prior for the bias function will be a GASP (see Sec. A.3).

If computation of y^M is fast, the Bayesian analysis can proceed directly. Otherwise (as in the spot welding example, where a single model run may take 30 minutes), we need to also incorporate the model approximation from Section 4 into the Bayesian analysis. We must then either add the GASP (hyper)parameters for y^M to the list of unknowns for a complete Bayesian analysis or use the plug-in MLE method if required by computational limitations; see Section 5.2.1 for details.

We choose the GASP for the bias to have correlation function of the same form as in (5), with its own set of covariance parameters $(\lambda^b, \boldsymbol{\beta}^b, \boldsymbol{\alpha}^b)$ but with all components of $\boldsymbol{\alpha}^b$ set at 2. Restricting $\boldsymbol{\alpha}^b$ at 2 (or even at some other value, such as 1.9) reduces the number of hyperparameters that must be taken into account. Because the bias cannot be observed directly and field data are usually scant, the information about the hyperparameters is limited, and reducing their number is computationally advantageous. Moreover, predictions and their error bounds will be only marginally affected by imposing this restriction. In fact, the restriction implies that the bias is very smooth, a condition all but certain to hold where reality, y^R, is smooth, a typical state in engineering and scientific applications; this smoothness assumption is also of help in deconfounding the bias and $\mathbf{u}$.

The mean function of the GASP for the bias process is typically chosen to be either zero or an unknown constant μ^b. Because the bias is not directly observed, it is doubtful whether more complicated mean structures are viable. For interpolation, the choice between zero mean and unknown level will have only a marginal effect on the results of the analysis; for extrapolation, however (as in the case of the mean of the GASP approximation to the code output described in Sec. 4) the latter choice might be more appropriate, because it may affect predictions and associated tolerance bounds (defined precisely in Sec. 5.2.3). As in the case of the GASP approximation to the computer model, a plug-in method can be used to determine the GASP correlation parameters, if required for computational simplification. This is discussed in Section 5.2.1.

5.2 Bayesian Inference

5.2.1 The Posterior Distribution and Its Computation. Assume first that approximation of y^M is not necessary. The modeling assumptions from Section 5.1 are that, for each field input **x**,

$$y^F(\mathbf{x}) = y^R(\mathbf{x}) + \epsilon^F,$$
$$y^R(\mathbf{x}) = y^M(\mathbf{x}, \mathbf{u}) + b_{\mathbf{u}}(\mathbf{x}),$$

and

$$\epsilon^F \sim \mathrm{N}(0, 1/\lambda^F).$$

Given the unknowns, these produce a multivariate normal density for the collection of all field data, $\mathbf{y}^F$, denoted by $f(\mathbf{y}^F | \mathbf{u}, \lambda^F, b)$. (Strictly, we should write $\mathbf{u}_\star$ instead of **u**, but in the Bayesian approach, all unknowns are considered random, and so we drop the $\star$ subscript for notational simplicity. We also suppress the dependence of b on **u**.) Denote the prior distribution of the unknown elements $(\mathbf{u}, \lambda^F, b)$ by $p(\mathbf{u}, \lambda^F, b)$. (Prior construction was already described briefly in Sec. 5.1; details are given in App. A.) Write the posterior density of these unknowns, given the data $\mathbf{y}^F$, as

$$p(\mathbf{u}, \lambda^F, b | \mathbf{y}^F) \propto f(\mathbf{y}^F | \mathbf{u}, \lambda^F, b)\, p(\mathbf{u}, \lambda^F, b). \qquad (11)$$

The posterior distribution is determined through MCMC techniques (cf. Robert and Casella 1999). Carrying out the MCMC analysis requires evaluating $y^M(\mathbf{x}, \mathbf{u})$ at each generated value of **u** and **x** in the field design space D^F. This is infeasible when model runs are expensive, in which case we resort to the GASP approximation of y^M, described in Section 4, to carry out the computations. This (unavoidably) introduces additional uncertainty into the predictions.

Two Key Simplifications. For reasons that have to do with achieving a stable MCMC algorithm, we recommend two simplifications, which together we call a *modular-MLE* analysis:

1. Use a modular analysis, in which the GASP hyperparameters for the computer model are determined only from the computer model data. In a full Bayesian analysis, the field data also could influence these hyperparameters. There are scientific as well as computational reasons for using the modular approach. These are discussed in Appendix A.
2. Rather than keeping GASP hyperparameters random in the Bayesian analysis, fix them (for both the computer model and the bias) at their MLEs; leave only the precisions and calibration parameters random. (Details on how these estimates are computed are given in App. A.) The reason for doing this is partly computational and partly to ensure that the methodology is stable. Further discussion of this is given in Appendix A.

Despite the fact that the modular-MLE analysis is only approximately Bayes, the resulting answers seem to be close to those from a full Bayesian analysis, at least when it comes to prediction (see Sec. 5.2.4). We note that this type of approximation was also used by, for example, Kennedy and O'Hagan (2001).

The resulting MCMC analysis (see App. B for details) produces a set of N draws from the posterior distribution of the unknowns $\mathbf{u}, \lambda^F, y^M(\mathbf{x}, \mathbf{u})$, and b. To be more precise, the output of the computations is a sample $\{\mathbf{u}^{(i)}, \lambda^{F(i)}, y^M(\mathbf{x}, \mathbf{u}^{(i)}), b^{(i)}(\mathbf{x}), i = 1, \ldots, N\}$. The posterior distribution of all quantities of interest can be estimated from these samples.

As an example, the posterior distributions of calibration or tuning parameters can be estimated by a histogram computed from the samples of the $\mathbf{u}^{(i)}$. From these samples, an estimate, $\hat{\mathbf{u}}$, of the unknown **u** also can be formed; for instance, the average of the samples is an approximation to the posterior mean of **u**. Credible intervals for **u** can be formed by taking appropriate percentiles of the ordered samples.

Spot Weld. The vector of controllable inputs is $\mathbf{x} = (C, L, G)$, and there is a tuning parameter u. Figure 4 gives the posterior density of u based on the modular-MLE approach. The estimated posterior mean is $\hat{u} = 3.28$. Clearly, there is considerable uncertainty in the values for u. Assessments of prediction accuracy (described in Sec. 5.2.2) account for this uncertainty and help alleviate the danger of overtuning that can result if one were to simply pick and use a single fixed parameter value, such as 3.28.

The considerable right tail here is likely due to the fact that there were data from two thicknesses (gauges) of material. The "optimal" tuning parameter for each gauge would be different. This again indicates how misleading it would be to simply choose a best estimate of the tuning parameter and proceed as if it were known. Note that the full-Bayesian analysis of Higdon et al. (2004) leads to a qualitatively similar posterior.

Similarly, the estimated bias function is given by

$$\hat{b}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} b^{(i)}(\mathbf{x}).$$

Separately graphing this bias function is not particularly useful, because of its very considerable posterior dependence on **u**. Thus, when we present bias functions in later figures, we give them conditionally on interesting values of **u**.

5.2.2 Predictions and Bias Estimates. The central issue for validation is assessing whether the accuracy of the predictions produced by the computer model is adequate for the model's intended use. The MCMC samples described earlier can be used to produce predictions with associated uncertainties, thus quantifying validation.

For instance, to predict the real process $y^R(\mathbf{x})$ at a set of (new) inputs D^F_{NEW} (denoting the resulting vector by $\mathbf{y}^R_{\mathrm{NEW}}$), all we need is access to draws from the posterior predictive distribution of

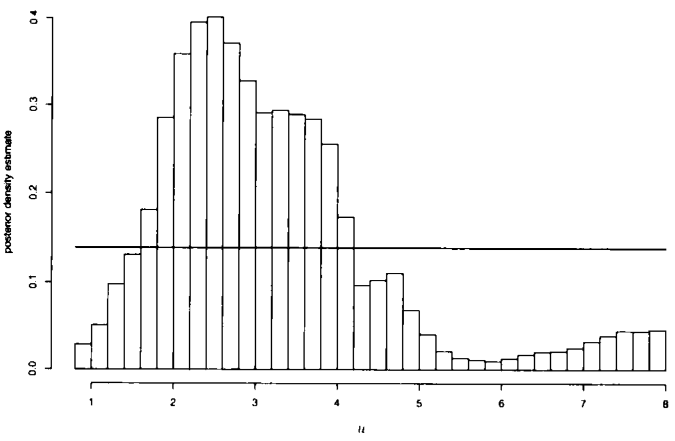

Figure 4. The Posterior Distribution of the Tuning Parameter u in the Spot Weld Example.

$\mathbf{y}^R_{\text{NEW}}$, $p(\mathbf{y}^R_{\text{NEW}}|\mathbf{y}^F, \mathbf{y}^M)$, where $\mathbf{y}^F$ and $\mathbf{y}^M$ are the available field and model data. Because of (9), these are obtained from draws from the joint posterior predictive of $\mathbf{y}^M_{\text{NEW}}$ and $\mathbf{b}_{\text{NEW}}$. Denote these draws by

$$\mathbf{y}^{M(i)}_{\text{NEW}},\qquad \mathbf{b}^{(i)}_{\text{NEW}},\qquad i=1,\dots,N. \tag{12}$$

Details on how to obtain such draws are given in Appendix C.

Pure-Model Prediction. If there are no calibration/tuning parameters, then define, for any $\mathbf{x} \in D^F_{\text{NEW}}$, the pure-model prediction of $y^R(\mathbf{x})$ simply as $\hat{y}^M(\mathbf{x})$. If we have available a new model run at input $\mathbf{x}$, then we do not need the approximation and can use $y^M(\mathbf{x})$; indeed, modelers often plan to perform a new model run if a prediction is desired at a new $\mathbf{x}$. If there are calibration/tuning parameters, we can use an estimate $\hat{\mathbf{u}}$ based on the previous data, evaluate $\hat{y}^M$ (or y^M if possible) at input $(\mathbf{x}, \hat{\mathbf{u}})$ and define the pure-model prediction as $\hat{y}^M(\mathbf{x}, \hat{\mathbf{u}})$. For $\hat{\mathbf{u}}$, we can use the posterior mean or mode of $\mathbf{u}$, although we could make other choices. Denote the pure-model prediction of $\mathbf{y}^R_{\text{NEW}}$ by $\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}})$.

For the spot welding example, the entire pure-model prediction function $\hat{y}^M(L, C, G, \hat{u})$, based on the computer model approximation and $\hat{u}$ the posterior mean, is (for four different values of load and gauge) plotted as a solid line in the top graphs of Figure 5.

Bias-Corrected Prediction. The bias-corrected prediction of the true process y^R at D^F_{NEW} is given by the estimate of the posterior predictive mean of $\mathbf{y}^R_{\text{NEW}}$,

$$\hat{\mathbf{y}}^R_{\text{NEW}} = \frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\right]. \tag{13}$$

If the code is fast, then the draw from the approximation to the code in the foregoing formula is replaced by its actual value.

When bias is present, the bias-corrected prediction improves on the pure-model prediction. For example, in the spot welding example the entire bias-corrected prediction function, $\hat{y}^R(L, C, G)$, is (for four different values of load and gauge) plotted as the solid line in the bottom graphs of Figure 5.

Bias of the Pure-Model Prediction. Because common practice today is to use some variant of pure-model prediction, it is useful to explicitly look at the bias of this procedure. The bias function of the pure-model prediction is clearly given by

$$\hat{\mathbf{b}}_{\hat{u}} \equiv \hat{\mathbf{y}}^R_{\text{NEW}} - \hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}}).$$

If one were actually trying to establish that the computer model is uniformly valid in some sense, then one would have to show that this bias function is effectively zero.

In the spot welding example, the bias function, $\hat{b}_{\hat{u}}(L, C, G)$, for pure-model prediction is plotted (for four different values of load and gauge) as the solid line in the middle graphs of Figure 5.

Variances of These Predictors. The covariance matrices corresponding to the pure-model predictor and the bias-corrected predictor can be estimated by

$$\operatorname{cov}(\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}})) = \frac{1}{N}\sum_{i=1}^{N}\left[\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}}) - \left(\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\right)\right] \times \left[\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}}) - \left(\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\right)\right]'$$

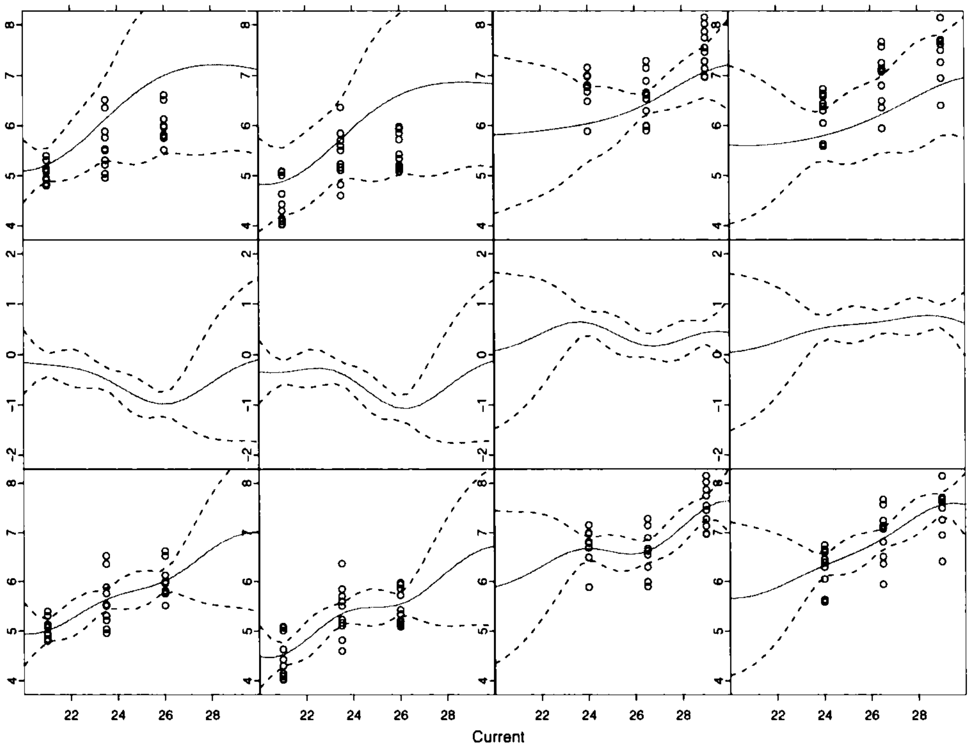

Figure 5. The Spot Welding Example. The first two columns correspond to G = 1 mm, with L = 4N and L = 5.3N, respectively; the next two columns correspond to G = 2 mm, with L = 4N and L = 5.3N. The first row gives the pure-model weld diameter predictions, $\hat{y}^M(L, C, G, \hat{u})$, and 90% tolerance bands. The middle row gives the associated biases, $\hat{b}_{\hat{u}}(L, C, G)$, and 90% tolerance bands. The last row gives the bias-corrected predictions, $\hat{y}^R(L, C, G)$, and 90% tolerance bands. The circles represent the field data observed at those input values.

and

$$\operatorname{cov}(\hat{\mathbf{y}}^R_{\text{NEW}}) = \frac{1}{N}\sum_{i=1}^{N}\big[\hat{\mathbf{y}}^R_{\text{NEW}} - \big(\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\big)\big] \times \big[\hat{\mathbf{y}}^R_{\text{NEW}} - \big(\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\big)\big]'.$$

It is easy to see that

$$\operatorname{cov}(\hat{\mathbf{y}}^R_{\text{NEW}}) = \operatorname{cov}(\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}})) - \hat{\mathbf{b}}_{\hat{u}}\,\hat{\mathbf{b}}'_{\hat{u}},$$

so that bias-corrected prediction clearly will have smaller variance than pure-model prediction (a strong incentive to use bias-corrected prediction).

5.2.3 Tolerance Bounds. As discussed in Section 1, we are concerned primarily with the predictive accuracy statement: "With probability γ, the prediction is within (tolerance) τ of the true $y^R(\mathbf{x})$." Such tolerance bounds for pure-model prediction are obtained straightforwardly from the samples (12). For a given γ, we can estimate $\boldsymbol{\tau} = (\tau(\mathbf{x}) : \mathbf{x} \in D^F_{\text{NEW}})'$ by making sure that $\gamma \times 100\%$ of samples satisfy

$$\big|\hat{\mathbf{y}}^M_{\text{NEW}}(\hat{\mathbf{u}}) - \big[\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\big]\big| < \boldsymbol{\tau}.$$

[In the previous formulas, all operations should be interpreted in a componentwise fashion, i.e., $|\mathbf{x}| = (|x_i|, i = 1, \dots, n)$ and $\mathbf{x} < \mathbf{y}$ iff $x_i < y_i$, $i = 1, \dots, n$.]

Similarly, for the bias-corrected prediction, the tolerance $\boldsymbol{\tau}$ is estimated by making sure that $\gamma \times 100\%$ of the samples satisfy

$$\big|\hat{\mathbf{y}}^R_{\text{NEW}} - \big[\mathbf{y}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}}\big]\big| < \boldsymbol{\tau}.$$

The tolerance bands for the bias of pure-model prediction follow from simply subtracting the pure-model prediction function, $\hat{y}^M(\mathbf{x}, \hat{\mathbf{u}})$, from the bands for bias-corrected prediction.

It can be convenient and straightforward—although we do not pursue the matter here—to modify the definition of tolerance bounds by making them asymmetric and to determine $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ such that $\gamma \times 100\%$ of the predictive samples satisfy

$$\hat{\mathbf{y}}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}} - \boldsymbol{\tau}_1 < \hat{\mathbf{y}}^R_{\text{NEW}} < \hat{\mathbf{y}}^{M(i)}_{\text{NEW}} + \mathbf{b}^{(i)}_{\text{NEW}} + \boldsymbol{\tau}_2,$$

subject to minimizing $\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2$ componentwise. This would be useful if bias were very large and the tolerance bounds would be one-sided or nearly so.

5.2.4 Comparison of Full Bayes and Modular-MLE Analyses. The spot welding example was examined from a fully Bayesian perspective by Higdon et al. (2004). Although their prior specification differs from ours, the results of the two studies are very similar. Recall that the advantage of our approach is that the sampling mechanism is quite stable and allows nonexperts to directly apply the methodology with relatively simple codes. In addition, for comparison purposes, a full Bayesian (modular) analysis of the pedagogic example was implemented, using methodology of Bayarri et al. (2002) and Paulo (2005). The two approaches yielded qualitatively very similar answers; in particular, the bias-corrected predictions and tolerance bands were almost identical.

6. FEEDBACK AND FEEDFORWARD (STEP 6)

The analyses in steps 4 and 5 will contribute to the dynamic process of improving the model and updating the I/U map by identifying the following:

- Model inputs whose uncertainties need to be reduced
- Needs (such as additional analyses and additional data) for closer examination of important regions or parts of the model
- Flaws that necessitate changes to the model
- Revisions to the evaluation criteria.

In the spot welding example, for instance, the posterior distribution of u (Fig. 4) will now replace the uncertainty entry in the I/U map. Another aspect of feedback is using the steps 4 and 5 analyses to further refine the validation process, for example, to design additional validation experiments.

The feed-forward notion is to develop the capability of predicting the accuracy of new models that are related to models that have been studied, but for which no specific field data are available. This can be done by using hierarchical Bayesian techniques, and we will explore it elsewhere.

7. CONCLUDING COMMENTS

Here we collect some relevant comments that otherwise would have impeded the flow of the article.

1. Combined validation and calibration. It is generally believed that data used for calibration/tuning cannot be used simultaneously for model validation. However, the Bayesian methodology described herein readily accommodates such simultaneous use of data by incorporating the posterior distribution of the tuning parameters in the overall assessment of uncertainties. In contrast, simply replacing a tuning parameter by some optimal "tuned" value $\hat{\mathbf{u}}$ (commonly done using least squares) obscures the interaction between bias and tuning and can lead to overly optimistic assessments of validity.
2. New model runs for prediction. In performing predictions, it is often sensible to include new model runs, if feasible, to obtain $y^M(\mathbf{x}, \hat{\mathbf{u}})$ for some key values of $\mathbf{x}$. In this article we emphasized prediction when such new runs are unavailable, but the analysis can easily incorporate such new runs (cf. App. C), without having to redo all of the computations from scratch. Using such model runs may be particularly helpful in assessing changes arising from moving from input $\mathbf{x}$ to nearby input $\mathbf{x}'$. We forego further consideration of this modification.

ACKNOWLEDGMENTS

This research was supported in part by grants from General Motors and the National Science Foundation (grant DMS-00-73952 to the National Institute of Statistical Sciences and DMS-01-03265), and by the Spanish Ministry of Science and Technology (grant MTM2004-03290). David Higdon and Marc Kennedy were central to the development of an earlier version of this framework. The authors are very grateful to the editor, associate editor, and referees for suggestions that greatly improved the manuscript.

APPENDIX A: THE STATISTICAL MODEL

A.1 Likelihood

Here we present the more complicated case of a slow computer model, when the approximation detailed in Section 4 must be used. The situation where y^M is fast follows as a particular case.

Recall that the design space for the model data is $D^M = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{u}_i)$, $i = 1, \ldots, m$. The model data are represented by $\mathbf{y}^M = (y^M(\mathbf{z}_1), \ldots, y^M(\mathbf{z}_m))'$. The design space for the field data is $D^F = \{\mathbf{x}_1^\star, \ldots, \mathbf{x}_n^\star\}$. The data consist of n_i replications taken at each point in D^F. Denoting these replications by $\{y_j^F(\mathbf{x}_i^\star),\ j = 1, \ldots, n_i\}$, given $y^R(\mathbf{x}_i^\star)$, $i = 1, \ldots, n$, we can replace the field data with the independent sufficient statistics $\bar{\mathbf{y}}^F = (\bar{y}^F(\mathbf{x}_1^\star), \ldots, \bar{y}^F(\mathbf{x}_n^\star))'$, where $\bar{y}^F(\mathbf{x}_i^\star) = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j^F(\mathbf{x}_i^\star)$, and $s_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n_i} [y_j^F(\mathbf{x}_i^\star) - \bar{y}^F(\mathbf{x}_i^\star)]^2$.

We denote the field design space augmented by the calibration parameters $\mathbf{u}$ by $D_\mathbf{u}^F$, which is the same design space as D^F, except that we simply replace each $\mathbf{x}_i^\star$ by $(\mathbf{x}_i^\star, \mathbf{u})$. It is useful to augment the observed data $(\mathbf{y}^M,\ \bar{\mathbf{y}}^F,\ s_F^2)$ with the bias function evaluated at D^F, $\mathbf{b}$, and the computer model evaluated at points in $D_\mathbf{u}^F$, denoted by $\mathbf{y}_\star^M$. In what follows, $\mathbf{n}' = (n_1, \ldots, n_n)$.

Define $C^f(D^g, D^h)$ to be the matrix with (i,j) entry $c^f(\mathbf{w}_i, \mathbf{w}_j)$, and define $\mu^f(D^f)$ to be the vector with component i equal to $\mu^f(\mathbf{w}_i)$, where $\mathbf{w}_i$ and $\mathbf{w}_j$ are the ith and jth points in the design spaces D^g and D^h. Also, let $C^f(D^g, D^g) \equiv C^f(D^g)$. Then

$$\begin{aligned} f(\bar{\mathbf{y}}^F, s_F^2, \mathbf{b}, \mathbf{y}_\star^M, \mathbf{y}^M | \boldsymbol{\theta}^L, \boldsymbol{\theta}^M, \mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u}) \\ = f(s_F^2 | \lambda^F) \times f(\bar{\mathbf{y}}^F | \mathbf{b}, \mathbf{y}_\star^M, \lambda^F) \times f(\mathbf{b} | \boldsymbol{\theta}^b, \mu^b) \\ \times f(\mathbf{y}_\star^M | \mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M, \mathbf{u}) \times f(\mathbf{y}^M | \boldsymbol{\theta}^L, \boldsymbol{\theta}^M), \end{aligned} \tag{A.1}$$

where, letting $\boldsymbol{\mu} = \mu^M(D_\mathbf{u}^F) + C^M(D_\mathbf{u}^F, D^M)\,[C^M(D^M)]^{-1}(\mathbf{y}^M - \mu^M(D^M))$ and $\boldsymbol{\Sigma} = C^M(D_\mathbf{u}^F) - C^M(D_\mathbf{u}^F, D^M)\ [C^M(D^M)]^{-1} \times C^M(D^M, D_\mathbf{u}^F)$,

$$f(s_F^2 | \lambda^F) = \lambda^F \chi^2 \left(\lambda^F s_F^2 \middle| \sum_{i=1}^{n} (n_i - 1) \right), \tag{A.2}$$

$$f(\bar{\mathbf{y}}^F | \mathbf{b}, \mathbf{y}_\star^M, \lambda^F) = \mathrm{N}\left(\bar{\mathbf{y}}^F \middle| \mathbf{y}_\star^M + \mathbf{b}, \frac{1}{\lambda^F} (\operatorname{diag} \mathbf{n})^{-1} \right), \tag{A.3}$$

$$f(\mathbf{b} | \boldsymbol{\theta}^b, \mu^b) = \mathrm{N}\left(\mathbf{b} \middle| \mu^b(D^F), \frac{1}{\lambda^b} C^b(D^F) \right), \tag{A.4}$$

$$f(\mathbf{y}_\star^M | \mathbf{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M, \mathbf{u}) = \mathrm{N}(\mathbf{y}_\star^M | \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{A.5}$$

and

$$f(\mathbf{y}^M | \boldsymbol{\theta}^L, \boldsymbol{\theta}^M) = \mathrm{N}(\mathbf{y}^M | \mu^M(D^M), C^M(D^M)). \tag{A.6}$$

Note that we can analytically integrate out $\mathbf{b}$ and $\mathbf{y}_\star^M$ in (A.1) to obtain

$$\begin{aligned} f(\bar{\mathbf{y}}^F, s_F^2, \mathbf{y}^M | \boldsymbol{\theta}^L, \boldsymbol{\theta}^M, \mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u}) \\ = f(s_F^2 | \lambda^F) \times \mathrm{N}\Bigg(\bar{\mathbf{y}}^F \Bigg| \boldsymbol{\mu} + \mu^b(D^F), \boldsymbol{\Sigma} + \frac{1}{\lambda^F} (\operatorname{diag} \mathbf{n})^{-1} \\ + \frac{1}{\lambda^b} C^b(D^F) \Bigg) \times f(\mathbf{y}^M | \boldsymbol{\theta}^L, \boldsymbol{\theta}^M). \end{aligned} \tag{A.7}$$

A.2 Modularization

Here we describe the approximate Bayesian analysis, which we refer to as the modular approach. The basic idea is to first do the analysis of all the model data, ignoring the contribution of the field data in estimating GASP model approximation parameters (including $\boldsymbol{\theta}^L$), then treat the model parameters (other than tuning parameters) as specified by the resulting posterior distribution—or possibly by their MLE—and incorporate the field data through a separate Bayesian analysis. Formally, this is a partial likelihood approach, treating (A.6) as the only part of the likelihood used to determine the model GASP parameters.

This approach is implemented as follows:

Stage 1. Analyze the model data in isolation to obtain the posterior density $p(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M | \mathbf{y}^M)$, using (A.6) together with the prior density $p(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$ specified in Section A.3. This will typically be represented by an MCMC cloud of realizations of points $(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$. Alternatively, if the MLE plug-in approach is used, then simply use $(\hat{\boldsymbol{\theta}}^L, \hat{\boldsymbol{\theta}}^M)$ in what follows.

Stage 2. To incorporate the field data $\mathbf{y}^F$, find the marginal posterior [defining $\boldsymbol{\theta} = (\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$]

$$\begin{aligned} p(\mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u} | \mathbf{y}^F, \mathbf{y}^M, \text{ stage 1}) \\ = \int p(\mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u} | \mathbf{y}^F, \mathbf{y}^M, \boldsymbol{\theta})\, p(\boldsymbol{\theta} | \mathbf{y}^M)\, d\boldsymbol{\theta}, \end{aligned}$$

or use $p(\mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u} | \mathbf{y}^F, \mathbf{y}^M, \hat{\boldsymbol{\theta}})$ if the MLE plug-in approach is used. This step is implemented by drawing a point from the stage 1 cloud [or using $(\hat{\boldsymbol{\theta}}^L, \hat{\boldsymbol{\theta}}^M)$]; generating $\mu^b, \boldsymbol{\theta}^b, \lambda^F, \mathbf{u}$, and perhaps also $\mathbf{b}$ and $\mathbf{y}_\star^M$; and repeating. Note that in generating from $p(\mu^b, \boldsymbol{\theta}^b, \lambda^F | \mathbf{y}^F, \mathbf{y}^M, \boldsymbol{\theta}, \mathbf{u})$, the full likelihood [(A.7) or (A.1)] must be used, together with the prior density $p(\mu^b, \boldsymbol{\theta}^b, \lambda^F)\, p(\mathbf{u})$.

The motivation and advantages of the modular approach are as follows:

1. Field data can affect the GASP model approximation parameters (the α's, β's, and λ's) in undesirable ways, allowing them to do some of the "tuning" of the model, instead of limiting the tuning effect of the field data to $\mathbf{u}$. Indeed, this was observed in the spot welding example, where $\mathbf{u}$ was shifted to the edge of its domain and the GASP parameters played the role of model "tuners." The modular approach prevents this from happening.
2. This easily generalizes to systems with several model components, M_i, each of which has separate model run data. Dealing first with the separate model run data in setting up the GASP model approximations and incorporating the field data only at the tuning/validation stage makes for an easier-to-understand and computationally much more efficient process.
3. Computations are simplified considerably, because the overall posterior factors into lower-dimensional blocks.

A.3 Prior Distributions

Paulo (2005) specifically addressed the problem of specifying the prior $p(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M)$ and sampling from the corresponding posterior $p(\boldsymbol{\theta}^L, \boldsymbol{\theta}^M | \mathbf{y}^M)$. In that article, several priors are derived and compared on the basis of their frequentist properties.

However, as already mentioned, for computational reasons, we recommend simply computing the MLEs of $\boldsymbol{\theta}^L$ and $\boldsymbol{\theta}^M$ based on model data alone, and considering those parameters as fixed in the second stage of the modular approach.

To carry out the analysis of the second stage, we must specify the prior on μ^b, $\boldsymbol{\theta}^b = (\boldsymbol{\beta}^b, \lambda^b)$, λ^F, and $\mathbf{u}$. The prior on the calibration parameter $\mathbf{u}$ is the one specified in the I/U map. Choosing default priors for the other bias GASP parameters is actually quite challenging, because of the typically limited data available, and the fact that no direct data about the bias are available. Also, as with the model GASP parameters, we noticed considerable confounding between the parameters, and thus opted for a method (described later) that fixes the $\boldsymbol{\beta}^b$ parameters at reasonable values and allows only λ^b (and possibly μ^b) to vary.

It then remains to choose priors for λ^b and λ^F. As long as replications are available, using a standard prior such as $1/\lambda^F$ should be fine for the error precision, but replications are not always available. Other problems are that the likelihood for λ^b can be quite flat, and λ^b can be highly confounded with $\mathbf{u}$. This leads us to advocate the use of data-dependent priors, centered at estimates of λ^b and λ^F.

Any of these choices can be criticized from a strict Bayesian standpoint, but we feel that there are compelling practical reasons to make them. First, a great deal of confounding is occurring here; we want the flexibility of GASPs in approximating the model and representing the bias, but they have too many parameters. Proper subjective priors for these parameters are simply not going to be available, and the principled objective priors of Paulo (2005) are computationally too intensive. Because the methodology is being designed for use by nonexperts, it also is not feasible to use more standard default priors with the advice to "watch out for convergence or stability issues." Finally, even with the rather ad hoc methods that we use to determine the GASP parameters (and center some of the priors), the variability of the resulting predictions seems to be similar to that from a full (careful) Bayesian analysis. Hence we feel that we are capturing the major uncertainties of the problem, while using a blend of techniques that results in a reliable and stable methodology.

Here are the details of the proposed implementation:

1. Using the first-stage approximation to the computer model, produce the pure-model prediction at the points in the field design space D^F augmented with a reasonable guess, $\tilde{\mathbf{u}}$, of the calibration parameter (e.g., the MLE or simply the a priori mean). Recall that we denote this augmented design space by $D^F_{\tilde{\mathbf{u}}}$. Denote the vector of resulting model predictions by $\tilde{\mathbf{y}}^M$.
2. Treat the vector $\mathbf{y}^F - \tilde{\mathbf{y}}^M$ as a realization from a Gaussian process with a nugget, namely as a realization from a multivariate normal with constant mean μ^b and covariance matrix $C^b(D^F)/\lambda^b + \mathbf{I}/\lambda^F$. Using the GASP software of W. Welch, we can then obtain an (MLE) estimate of $\boldsymbol{\beta}^b$, which will be the fixed value used in the analysis. Note that if the model and field design points were the same, then there would be no need to use the model approximation to determine the vector $\tilde{\mathbf{y}}^M$.
3. The GASP software also will yield MLE estimates, $\hat{\lambda}^b$ and $\hat{\lambda}^F$, but it is important to allow λ^b and λ^F to vary in the Bayesian analysis. For these parameters, we choose independent exponential priors with means equal to a modest multiple (e.g., 5) of the MLEs. In line with Paulo (2005), experience has shown that the final predictions are relatively insensitive to the choice of the multiplying factor.
4. If a nonzero mean, μ^b, is used for the bias, then we suggest simply using the usual constant prior (which can be shown to yield a proper posterior). We typically do not use a mean for the bias; we usually set $\mu^b = 0$.

APPENDIX B: THE MCMC METHOD FOR POSTERIOR INFERENCE

Here we present the details of the MCMC method for posterior inference under the modular MLE approach that we recommend for routine implementation of the methodology. (When performing a full Bayesian analysis, algorithms described in Paulo 2005 and Bayarri et al. 2002 work well, although they may require monitoring and tuning.)

As detailed in Section A.3, the only parameters that have not been fixed are the calibration parameter $\mathbf{u}$, the precisions λ^F and λ^b, and possibly the bias mean μ^b. These are sampled in the MCMC; given the current state of the chain $\mathbf{y}^M_{\star\text{old}}, \mathbf{b}_{\text{old}}, \lambda^F_{\text{old}}, \lambda^b_{\text{old}}, \mathbf{u}_{\text{old}}$, we determine the next state as follows:

1. Generate $(\mathbf{y}^M_{\star\text{new}}, \mathbf{b}_{\text{new}})$ directly from its full conditional, which is a multivariate normal whose parameters are determined using the fact that, conditional on all other parameters, the distribution of $(\mathbf{y}^M_\star, \mathbf{b}, \mathbf{y}^M, \bar{\mathbf{y}}^F)$ is multivariate normal with readily computed mean vector and covariance matrix.
2. Generate λ^F_{new} from its full conditional, which is $\Gamma(\lambda^F | a_1, a_2)$, where $a_1 = \sum_{i=1}^n n_i/2 + \alpha_F$ and $a_2 = r_F + s_F^2/2 + (\bar{\mathbf{y}}^F - \mathbf{b}_{\text{new}} - \mathbf{y}^M_{\star\text{new}})' \operatorname{diag} \mathbf{n}\, (\bar{\mathbf{y}}^F - \mathbf{b}_{\text{new}} - \mathbf{y}^M_{\star\text{new}})/2$ if, a priori, $\lambda^F \sim \Gamma(\alpha_F, r_F)$ (In Sec. A.3 we recommended $\alpha_F = 1$ and $r_F = 5\hat{\lambda}^F$, but the MCMC works in this more general setting as well.)
3. Generate λ^b_{new} directly from its full conditional, which is $\Gamma(\lambda^b | a_1, a_2)$, where $a_1 = n/2 + \alpha_b$ and $a_2 = r_F + \mathbf{b}'_{\text{new}}\, [C^b(D^F)]^{-1}\, \mathbf{b}_{\text{new}}/2$, if, a priori, $\lambda^b \sim \Gamma(\alpha_b, r_b)$. (In Sec. A.3 we recommended $\alpha_b = 1$ and $r_b = 5\hat{\lambda}^b$, but the MCMC works in this more general setting as well.)

   If a nonzero bias mean is used in the analysis, then we also must sample μ^b_{new} directly from its full conditional. This is a normal distribution with mean $\mathbf{1}'\, [C^b(D^F)]^{-1} \times \mathbf{b}_{\text{new}} / \mathbf{1}'[C^b(D^F)]^{-1}\mathbf{1}$ and precision given by $\lambda^b_{\text{new}} \times \mathbf{1}'[C^b(D^F)]^{-1}\mathbf{1}$.
4. Generate $\mathbf{u}_{\text{new}}$ using a Metropolis–Hastings step (e.g., Robert and Casella 1999). We have had success with the strategy of choosing with probability Q (e.g., .5) to propose a draw from the prior on $\mathbf{u}$, $p(\mathbf{u})$, and with probability $1 - Q$ to propose a locally perturbed version of the current value of the chain, that is, a random vector drawn

from the product of uniform distributions on the intervals $(u_{i,\text{old}} - \epsilon_i, u_{i,\text{old}} + \epsilon_i)$, where $u_{i,\text{old}}$ is the ith component of $\mathbf{u}_{\text{old}}$ and the ϵ_i are chosen as, say, a fixed proportion of the range of u_i in the prior.

APPENDIX C: PREDICTIONS

For prediction, it is necessary to sample from the posterior predictive distribution of the real process evaluated at a set D^F_{NEW} of new design points, namely

$$\int p(\{y^M(\mathbf{x},\mathbf{u}), b(\mathbf{x}) : \mathbf{x} \in D^F_{\text{NEW}}\} | \bar{\mathbf{y}}^F, s^2_F, \mathbf{y}^M, \boldsymbol{\theta}) \times p(\boldsymbol{\theta} | \bar{\mathbf{y}}^F, s^2_F, \mathbf{y}^M)\, d\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ represents the vector of parameters that have not been fixed at some value. To obtain these draws, we proceed as follows: For each element of a sample from the posterior distribution of $\boldsymbol{\theta}$, $p(\boldsymbol{\theta} | \bar{\mathbf{y}}^F, s^2_F, \mathbf{y}^M)$, say $\boldsymbol{\theta}^{(i)}$, we must generate a realization from $p(\{y^M(\mathbf{x},\mathbf{u}), b(\mathbf{x}) : \mathbf{x} \in D^F_{\text{NEW}}\} | \bar{\mathbf{y}}^F, s^2_F, \mathbf{y}^M, \boldsymbol{\theta}^{(i)})$. This distribution is multivariate normal with parameters readily computed using standard Kalman filter formulas.

If we decided to collect more computer model data to aid prediction, then formally we should rerun the MCMC to update the posterior of the unknown parameters given this additional information. That is rarely practical, even if we are following a modular approach, so we recommend adding the additional code data to the vector $\mathbf{y}^M$ but leaving all other aspects of the posterior unchanged.

[Received April 2005. Revised December 2006.]

REFERENCES

Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J., and Welch, W. J. (1998), "Circuit Optimization via Sequential Computer Experiments: Design of an Output Buffer," *Applied Statistics*, 47, 31–48.

Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P. (1996), "Experimental Design and Observation for Large Systems," *Journal of the Royal Statistical Society*, Ser. B, 58, 77–94.

Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., and Walsh, D. (2006a), "Computer Model Validation With Functional Output," Technical Report 165, National Institute of Statistical Sciences.

Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2002), "A Framework for Validation of Computer Models," Technical Report 128, National Institute of Statistical Sciences.

Bayarri, M. J., Berger, J. O., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Lin, C. H., and Tu, J. (2006b), "Bayesian Validation of a Computer Model for Vehicle Crashworthiness," Technical Report 163, National Institute of Statistical Sciences.

Berk, R., Bickel, P., Campbell, K., Fovell, R., Keller-McNulty, S., Kelly, E., Linn, R., Park, B., Perelson, A., Rouphail, N., Sacks, J., and Schoenberg, F. (2002), "Workshop on Statistical Approaches for the Evaluation of Complex Computer Models," *Statistical Science*, 17, 173–192.

Cafeo, J., and Cavendish, J. (2001), "A Framework for Verification and Validation of Computer Models and Simulations," internal document, General Motors, Research & Development Center.

Caswell, H. (1976), "The Validation Problem," in *Systems Analysis and Simulation in Ecology*, Vol. IV, ed. B. Patten, New York: Academic Press, pp. 313–325.

Chapman, W., Welch, W., Bowman, K., Sacks, J., and Walsh, J. (1994), "Arctic Sea Ice Variability: Model Sensitivities and a Multidecadal Simulation," *Journal of Geophysical Research*, 99, 919–935.

Conti, S., Anderson, C., O'Hagan, A., and Kennedy, M. (2005), "Bayesian Analysis of Complex Dynamic Computer Models," in *Sensitivity Analysis of Model Output*, eds. K. Hanson and F. Hemez, Los Alamos National Laboratory, pp. 147–156; available at *http://library.lanl.gov/ccw/samo2004/*.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001), "Bayesian Forecasting for Complex Systems Using Computer Simulators," *Journal of the American Statistical Association*, 96, 717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1997), "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. III, eds. C. Gatsonis, J. S. Hodges, R. E. Kass, R. McCulloch, P. Rossi, and N. D. Singpurwalla, New York: Springer, pp. 36–93.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, 86, 953–963.

Easterling, R. G. (2001), "Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations," Technical Report SAND2001-0243, Sandia National Laboratories.

Goldstein, M., and Rougier, J. C. (2003), "Calibrated Bayesian Forecasting Using Large Computer Simulators," technical report, University of Durham, Statistics and Probability Group.

——— (2004), "Probabilistic Formulations for Transferring Inferences From Mathematical Models to Physical Systems," technical report, University of Durham, Statistics and Probability Group.

Gough, W., and Welch, W. (1994), "Parameter Space Exploration of an Ocean General Circulation Model Using an Isopycnal Mixing Parametrization," *Journal of Marine Research*, 52, 773–796.

Gramacy, R., Lee, H., and Macready, W. (2004), "Parameter Space Exploration With Gaussian Process Trees," in *Proceedings of the International Conference on Machine Learning*, pp. 353–360.

Gustafson, P. (2005), "On Model Expansion, Model Contraction, Identifiability and Prior Information: Two Illustrative Scenarios Involving Mismeasured Variables," *Statistical Science*, 20, 111–140.

——— (2006), "Sample Size Implications When Biases Are Modeled Rather Than Ignored," *Journal of the Royal Statistical Society*, Ser. A, 169, 1–17.

Higdon, D. (2002), "Space and Space-Time Modeling Using Process Convolutions," in *Quantitative Methods for Current Environmental Issues*, eds. C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, London: Springer-Verlag, pp. 37–56.

Higdon, D., Gattiker, J., and Williams, B. (2005), "Computer Model Calibration Using High Dimensional Output," Technical Report LAUR-05-6410, Los Alamos National Laboratories.

Higdon, D., Kennedy, M. C., Cavendish, J., Cafeo, J., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Kennedy, M. C., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models" (with discussion), *Journal of the Royal Statistical Society*, Ser. B., 63, 425–464.

Kennedy, M. C., O'Hagan, A., and Higgins, N. (2002), "Bayesian Analysis of Computer Code Outputs," in *Quantitative Methods for Current Environmental Issues*, eds. C. W. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi, London: Springer-Verlag, pp. 227–243.

Lee, H., Higdon, D., Bi, Z., Ferreira, M., and West, M. (2002), "Markov Random Field Models for High-Dimensional Parameters in Simulations of Fluid Flow in Porous Media," *Technometrics*, 44, 230–241.

Lee, H., Higdon, D., Calder, C., and Holloman, C. (2005), "Efficient Models for Correlated Data via Convolutions of Intrinsic Processes," *Statistical Modelling*, 5, 53–74.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.

Oakley, J. (2004), "Estimating Percentiles of Computer Code Outputs," *Applied Statistics*, 53, 83–93.

Oakley, J., and O'Hagan, A. (2002), "Bayesian Inference for the Uncertainty Distribution of Computer Model Outputs," *Biometrika*, 89, 769–784.

——— (2004), "Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach," *Journal of the Royal Statistical Society*, Ser. B, 66, 751–769.

Oberkampf, W., and Trucano, T. (2000), "Validation Methodology in Computational Fluid Dynamics," Technical Report 2000-2549, American Institute of Aeronautics and Astronautics.

Oreskes, N., Shrader-Frechette, K., and Belitz, K. (1994), "Verification, Validation and Confirmation of Numerical Models in the Earth Sciences," *Science*, 263, 641–646.

Paulo, R. (2005), "Default Priors for Gaussian Processes," *The Annals of Statistics*, 33, 556–582.

Pilch, M., Trucano, T., Moya, J. L., Froehlich, G., Hodges, A., and Peercy, D. (2001), "Guidelines for Sandia ASCI Verification and Validation Plans: Content and Format, Version 2.0," Technical Report SAND 2001-3101, Sandia National Laboratories.

Roache, P. (1998), *Verification and Validation in Computational Science and Engineering*, Albuquerquenm: Hermosa Publishers.

Robert, C., and Casella, G. (1999), *Monte Carlo Statistical Methods*, New York: Springer-Verlag.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423.

Saltelli, A., Chan, K., and Scott, M. (eds.) (2000), *Sensitivity Analysis*, Chichester, U.K.: Wiley.

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

Schmidt, A., and O'Hagan, A. (2003), "Bayesian Inference for Nonstationary Spatial Covariance Structure via Spatial Deformatations," *Journal of the Royal Statistical Society*, Ser. B, 65, 745–758.

Sellke, T., Bayarri, M., and Berger, J. (2001), "Calibration of p Values for Testing Precise Null Hypotheses," *The American Statistician*, 55, 62–71.

Stainforth, D. A., Aina, T., Christensen, C., Collins, M., Faull, N., Frame, D. J., Kettleborough, J. A., Knight, S., Martin, A., Murphy, J. M., Piani, C., Sexton, D., Smith, L. A., Spicer, R. A., Thorpe, A. J., and Allen, M. R. (2005), "Uncertainty in Predictions of the Climate Response to Rising Levels of Greenhouse Gases," *Nature*, 433, 403–406.

Trucano, T., Pilch, M., and Oberkampf, W. O. (2002), "General Concepts for Experimental Validation of ASCII Code Applications," Technical Report SAND 2002-0341, Sandia National Laboratories.

Wang, P., and Hayden, D. (1999), "Computational Modeling of Resistance Spot Welding of Aluminum," Research Report R&D-9152, GM Research & Development Center.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.