AIAA JOURNAL
Vol. 46, No. 5, May 2008

Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output

John McFarland* and Sankaran Mahadevan†
Vanderbilt University, Nashville, Tennessee 37235
and
Vicente Romero‡ and Laura Swiler§
Sandia National Laboratories, Albuquerque New Mexico 87185

DOI: 10.2514/1.35288

Model calibration analysis is concerned with the estimation of unobservable modeling parameters using observations of system response. When the model being calibrated is an expensive computer simulation, special techniques such as surrogate modeling and Bayesian inference are often fruitful. In this paper, we show how the flexibility of the Bayesian calibration approach can be exploited to account for a wide variety of uncertainty sources in the calibration process. We propose a straightforward approach for simultaneously handling Gaussian and non-Gaussian errors, as well as a framework for studying the effects of prescribed uncertainty distributions for model inputs that are not treated as calibration parameters. Further, we discuss how Gaussian process surrogate models can be used effectively when simulator response may be a function of time and/or space (multivariate output). The proposed methods are illustrated through the calibration of a simulation of thermally decomposing foam.

Nomenclature

$\boldsymbol{d}$	=	experimentally observed values of the response
$G(\cdot)$	=	simulation-model operator
m	=	number of training points
n	=	number of experimental observations of the response
p	=	dimensionality of input
$\boldsymbol{s}$	=	scenario-descriptor inputs to the simulator
u	=	characterized observation, modeling uncertainty
$\boldsymbol{x}$	=	input variables
Y	=	response value
β	=	process mean
ε	=	random variable describing difference between predictions and observations
$\boldsymbol{\theta}$	=	calibration inputs to the simulator
λ	=	process variance
$\boldsymbol{\xi}$	=	simulation inputs with prescribed uncertainty
$\boldsymbol{\phi}$	=	parameters of the correlation function
σ^2	=	variance of ε

I. Introduction

THE importance of uncertainty in the modeling and simulation process is often overlooked. No model is a perfect representation of reality, and so it is important to ask how imperfect a model is before it is applied for prediction. The scientific community relies heavily on modeling and simulation tools for forecasting, parameter studies, design, and decision-making. However, these are all activities that can strongly benefit from meaningful representations of modeling uncertainty. For example, forecasts can contain error bars, designs can be made more robust, and decision-makers can be better-informed when modeling uncertainty is quantified to support these activities.

The set of activities that involve the quantification of uncertainty in the modeling and simulation process includes verification, validation, calibration, and uncertainty propagation. Verification focuses on the comparison of a computational implementation with a conceptual model, to verify the implementation and assess the amount of error introduced via numerical processes. Validation, on the other hand, is a process for comparing the computational implementation of a model against experimentally observed outcomes; this is another opportunity to quantify errors and uncertainties. Similarly, calibration involves comparing the implementation of a model with observations, but the objective is to use this comparison to make inferences about unknown parameters that govern the computational implementation. Uncertainty propagation is the process of estimating the uncertainty on the model output that is implied by uncertainty on the model inputs. This paper emphasizes that the model-calibration activity in conjunction with uncertainty propagation is an opportunity to quantify the effects on model predictions of a variety of uncertainty sources, including data uncertainty and modeling assumptions.

Calibration is a far-reaching term and can mean quite different things to different people. This paper deals only with a specific form of model calibration that is actually a special case of inverse problem analysis, in that the objective is to use observations of the simulator output to make inference about simulator inputs. This type of calibration analysis poses several problems in practice:

1) The simulation is often expensive, rendering an exhaustive exploration of the parameter space infeasible.

2) Various ranges and/or combinations of input parameters may yield simulator outputs that are comparable with the observed data.

3) The observed data contain some degree of error or uncertainty.

4) When the response quantity of interest is multivariate, the most appropriate measure of agreement between the simulator output and observed data is not obvious.

Previous work addressing the listed challenges is limited. Reference [1] gives an overview of various statistical methods that have been proposed for the calibration of computer simulations. One of the most straightforward approaches is to pose the calibration

Received 24 October 2007; revision received 24 January 2008; accepted for publication 26 January 2008. Copyright © 2008 by the American Institute of Aeronautics and Astronautics, Inc. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/08 $10.00 in correspondence with the CCC.

*Ph.D. Candidate, Department of Mechanical Engineering. Student Member AIAA.

†Professor, Department of Civil and Environmental Engineering and Mechanical Engineering. Member AIAA.

‡Senior Member, Technical Staff, Model Validation and Uncertainty Quantification Department. Senior Member AIAA.

§Principal Member, Technical Staff, Optimization and Uncertainty Estimation Department. Member AIAA.

problem in terms of nonlinear regression analysis [2]. The problem is then attacked using standard optimization techniques, for example, to minimize the sum of the squared errors between the predictions and observations. Reference [3] illustrates the use of such a method to obtain point estimates and various types of confidence intervals for a groundwater flow model.

Other methods that have been proposed include the generalized likelihood-uncertainty-estimation (GLUE) procedure [4], which is somewhat Bayesian, in that it attempts to characterize a predictive response distribution by weighting random parameter samples by their likelihoods. However, the GLUE method does not assume a particular distributional form for the errors, which prevents the application of rigorous probabilistic approaches, including maximum-likelihood estimation (MLE). Methods having their foundation in system identification and being related to the Kalman filter have also been proposed for model calibration and are particularly suited for situations in which new data become available over time [5,6].

Dealing with the first challenge previously listed, the fact that the simulation being calibrated is often expensive generally requires the use of an inexpensive approximation, or surrogate model. With the increasing complexity of computer simulations, there has been substantial interest in techniques for the design and analysis of computer experiments. The use of Gaussian process (GP) interpolation has been particularly popular [7–12]. Other approaches that have been considered include techniques for combining simulations with different levels of complexity [13].

One of the milestone papers for model calibration is the work of Kennedy and O'Hagan [10]. Not only does their formulation treat the computational simulation as a black box, replacing it by a GP surrogate, but they also purport to account for all of the uncertainties and variabilities that may be present. Toward this end, they formulate the calibration problem using a Bayesian framework, and both multiplicative and additive discrepancy terms are included to account for any deviations of the predictions from the experimental data that are not taken up in the simulation input parameters. Further, the additive discrepancy term is formulated as a GP indexed by the scenario variables (boundary conditions, initial conditions, etc.) that describe the system being modeled. In this regard, their formulation is particularly powerful for cases in which experimental data are available at a relatively large number of different scenarios, and predictions of interest are characterized by extrapolations (or interpolations) in this scenario space. Implementation of their complete framework is quite demanding and requires extensive use of numerical integration techniques such as quadrature or Markov chain Monte Carlo (MCMC) integration.

There have been few attempts in the literature to illustrate how calibration methodologies providing uncertainty representations should be applied to large-scale problems, in which simulation time is long, the number of parameters to be estimated may be high, the amount of experimental data is small, and the response quantity is multivariate. The example reported in [10] deals with a relatively large amount of experimental data, a small parameter space, and a scalar response quantity.

Furthermore, part of the power of the Bayesian approach is its flexibility, but there has been little previous work that shows how the Bayesian model-calibration approach can be extended to account for additional forms of uncertainty that are common to real-world modeling and simulation applications. Such extensions include the ability to handle measurement uncertainty characterized with bounds (as opposed to a Gaussian distribution) and model input parameters with prescribed uncertainty distributions.

The purpose of this paper is to advance the state of the art in Bayesian model calibration through the development and illustration of the extensions previously mentioned. Section II describes the use of GP interpolation as a surrogate-modeling technique, and Sec. II.B introduces our proposed approach for capturing simulator response that is highly multivariate, particularly response that is a function of temporal and/or spatial coordinates. Section III discusses the theory underlying the Bayesian calibration approach, including two extensions for uncertainty quantification described in Secs. III.A.1 and III.A.2. Finally, Sec. IV presents a case study based on the thermal simulation of decomposing foam to illustrate the entire Bayesian calibration methodology.

II. Gaussian Process Models

GP interpolation (which, in most cases, is equivalent to the family of methods that go by the name of *kriging predictors*) is a powerful technique based on spatial statistics. Not only can Gaussian process models be used to fit a wide variety of functional forms, but they also provide a direct estimate of the uncertainty associated with their predictions. Gaussian process models are increasingly being used as surrogates to expensive computer simulations for the purposes of optimization and uncertainty propagation [10,14–18].

The basic idea of the GP model is that the response values Y are modeled as a group of multivariate normal random variables [8,9]. A parametric covariance function is then constructed as a function of the inputs $\boldsymbol{x}$. The covariance function is based on the idea that when the inputs are close together, the correlation between the outputs will be high. As a result, the uncertainty associated with the model's predictions is small for input values that are close to the training points and it is large for input values that are not close to the training points. In addition, the mean function of the GP may capture large-scale variations, such as a linear or quadratic regression of the inputs (generally referred to as a *trend function* in the kriging literature [19]). The effect of the mean function on predictions that interpolate the training data tends to be small, but when the model is used for extrapolation, the predictions will follow the mean function very closely as soon as the correlations with the training data become negligible. Because the models used here are intended for data interpolation only, and for simplicity, we consider only Gaussian process models with a constant mean function.

Thus, we denote by Y a Gaussian process with mean and covariance given by

$$E[Y(\boldsymbol{x})] = \beta \tag{1}$$

and

$$\operatorname{cov}[Y(\boldsymbol{x}), Y(\boldsymbol{x}^*)] = \lambda c(\boldsymbol{x}, \boldsymbol{x}^* \mid \boldsymbol{\phi}) \tag{2}$$

where $c(\boldsymbol{x}, \boldsymbol{x}^* \mid \boldsymbol{\phi})$ is the correlation between $\boldsymbol{x}$ and $\boldsymbol{x}^*$, $\boldsymbol{\phi}$ is the vector of parameters governing the correlation function, and λ is the process variance.

Consider that we observed the process at m locations (the training or design points) $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ of a p-dimensional input variable, so that we have the resulting observed random vector

$$\boldsymbol{Y} = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_m))^T$$

By definition, the joint distribution of $\boldsymbol{Y}$ satisfies

$$\boldsymbol{Y} \sim \boldsymbol{N}_m(\beta \boldsymbol{1}, \lambda \boldsymbol{R}) \tag{3}$$

where $\boldsymbol{R}$ is the $m \times m$ matrix of correlations among the training points. Under the assumption that the parameters governing both the trend function and the covariance function are known, the expected value and variance (uncertainty) at any (possibly untested) location $\boldsymbol{x}$ are calculated as

$$E[Y(\boldsymbol{x}) \mid \boldsymbol{Y}] = \beta + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{Y} - \beta 1) \tag{4}$$

and

$$\operatorname{var}[Y(\boldsymbol{x}) \mid \boldsymbol{Y}] = \lambda(1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r}) \tag{5}$$

where $\boldsymbol{r}$ is the vector of correlations between $\boldsymbol{x}$ and each of the training points. Further, the full covariance matrix associated with a vector of predictions can be constructed using the following equation for the pairwise covariance elements:

$$\operatorname{cov}[Y(\boldsymbol{x}), Y(\boldsymbol{x}^*) \mid \boldsymbol{Y}] = \lambda[c(\boldsymbol{x}, \boldsymbol{x}^*) - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r}_*] \tag{6}$$

where $\boldsymbol{r}_*$ is the vector of correlations between $\boldsymbol{x}_*$ and each of the training points.

There are a variety of possible parameterizations of the correlation function [8,19]. Statisticians have traditionally recommended the Matérn correlation function [19,20], whereas engineers often use the squared-exponential formulation [9,21] for its ease of interpretation and because it results in a smooth, infinitely differentiable function [8]. This work uses the squared-exponential form, which is given by

$$c(\boldsymbol{x}, \boldsymbol{x}^*) = \exp\left[-\sum_{i=1}^{p} \phi_i (x_i - x_i^*)^2\right] \tag{7}$$

where p is the dimension of $\boldsymbol{x}$, and the p parameters ϕ_i must be nonnegative.

A. Parameter Estimation

Before applying the Gaussian process model for prediction, values of the parameters $\boldsymbol{\phi}$ and β must be chosen. Further, the value for λ also must be selected if Eq. (5) is to be used for uncertainty estimation. The most commonly used method for parameter estimation with Gaussian process models is the method of maximum-likelihood estimation [9,22].

MLE involves finding those parameters that maximize the likelihood function; the likelihood function describes the probability of observing the training data for a particular parameter set and is based on the multivariate normal distribution in this case. For computational reasons, the problem is typically formulated as a minimization of the negative log of the likelihood function:

$$-\log l(\boldsymbol{\phi}, \beta, \lambda) = m \log \lambda + \log |\boldsymbol{R}| + \lambda^{-1} (\boldsymbol{Y} - \beta \boldsymbol{1})^T \boldsymbol{R}^{-1} (\boldsymbol{Y} - \beta \boldsymbol{1}) \tag{8}$$

The numerical minimization of Eq. (8) can be an expensive task, because the $m \times m$ matrix $\boldsymbol{R}$ must be inverted for each evaluation. Fortunately, the gradients are available in analytic form [9,22]. Further, the optimal values of the process mean and variance, conditional on the correlation parameters $\boldsymbol{\phi}$, can be computed exactly. The optimal value of β is equivalent to the generalized least-squares estimator:

$$\hat{\beta} = (\boldsymbol{1}^T \boldsymbol{R}^{-1} \boldsymbol{1})^{-1} \boldsymbol{1}^T \boldsymbol{R}^{-1} \boldsymbol{Y} \tag{9}$$

However, Eq. (9) is highly susceptible to round-off error, particularly when $\boldsymbol{R}$ is ill-conditioned. We obtained better results by using the ordinary least-squares estimator, which is simply the mean of $\boldsymbol{Y}$ in this case. The conditional optimum for the process variance is given by

$$\hat{\lambda} = \frac{\boldsymbol{1}}{m} (\boldsymbol{Y} - \beta \boldsymbol{1})^T \boldsymbol{R}^{-1} (\boldsymbol{Y} - \beta \boldsymbol{1}) \tag{10}$$

B. Multivariate Output: Time and Space

In many cases, the computer simulation may output the response quantity of interest (e.g., temperature) at a large number of time instances and/or spatial locations. Such cases are sometimes termed *multivariate output*, because the response at each time or space instance can be thought of as a separate output variable.

Unfortunately, though, this introduces a considerable amount of additional complexity when we want to use a Gaussian process to model the code output. The simplest solution is probably to use a small number of features to represent the entire output. However, in many cases, we would like to take the entire output spectrum into account, to ensure agreement with the experimental data at all output locations.

If the dimensionality of the output spectrum is small (say, four or five outputs), we might consider building a separate, independent Gaussian process model for each output quantity. However, this approach becomes far too cumbersome when there are many time and/or space instances to consider. When we want to consider a large output spectrum, one possible approach is to treat those variables that index the output spectrum (e.g., time and location) as additional inputs to the surrogate. In this way, we deal with only one surrogate, and the output can be treated as a scalar quantity.

This approach, however, introduces its own difficulties. Consider a design of computer experiments based on 50 Latin hypercube (LH) samples for a computer simulation that outputs the response quantity at 1000 time instances. When time is parametrized as an input, this gives a total of 50,000 training points for the Gaussian process model. This will make the MLE process virtually impossible, because it will require the repeated inversion of a $50,000 \times 50,000$ correlation matrix. Further, if there is a significant degree of autocorrelation with time (which will almost certainly be the case, particularly if the code output uses small time intervals), this correlation matrix will be highly ill-conditioned and likely singular to numerical precision.

There are several possible methods for dealing with these issues. One approach that has been used in the past is a decomposition of the correlation matrix that is applicable when the training data form a grid design [16,23] (i.e., the output from each code run gives the response at the same time instances). The inverse of the correlation matrix is then computed based on a Kronecker product, and so instead of inverting a $50,000 \times 50,000$ matrix, two matrices are inverted, one of size 50×50 and one of size 1000×1000. However, this method is fairly complicated to implement, and it does not avoid the problem with ill-conditioned correlation matrices.

Most other solutions are based on the omission of a subset of the available points. Because the response is most likely strongly autocorrelated in time, many of the points are redundant anyway. The difficulty, though, is how to decide which points to throw away. Considering again the preceding example, even if the number of time instances is reduced from 1000 to 20, there are still 1000 training points (20 time instances times 50 samples) for the Gaussian process, which is likely too many.

To circumvent this problem, we propose an algorithm based on the *greedy-algorithm* concept, for selecting among a set of candidate training points. The underlying concept of a greedy algorithm is to follow a problem-solving procedure such that the locally optimal choice is made at each step [24]. We apply this concept to the problem of choosing among available surrogate-model training points by iteratively adding points one at a time, and the point added at each step is that point corresponding to the largest prediction error. This approach has several advantages:

1) The point-selection technique is easier to implement than the Kronecker product factorization of the correlation matrix.

2) It is not restricted to maintaining the grid design. That is, we may choose a subset of points such that code run 1 may be represented at time instance 1, but code run 2 may not get represented at time instance 1. Further, a nonuniform time spacing may be selected: perhaps there is more activity in the early time portion, and so more points are chosen in that region.

3) The amount of subjectivity associated with choosing which points to retain is strongly reduced. Instead of deciding on a new grid spacing, we can instead choose a desired total sample size or maximum error.

4) The one-at-a-time process of adding points to the model makes it easy to pinpoint exactly when numerical matrix singularity issues begin to come into play (if at all). This is particularly useful for very large data sets containing redundant information.

The greedy point-selection approach is outlined next. Let us denote the total number of available points by m_t, the set containing the selected points by Θ, the set containing the points not yet selected by Ω, and the size of Θ by m. Also, denote the maximum-allowed number of points as m^*, the desired cross-validation prediction error by δ^*, and the current vector of cross-validation prediction errors by $\boldsymbol{\delta}$.

1) Generate a very small ($\sim$5) initial subset Θ. Ideally, this is chosen randomly, because the original set of points is most likely structured.

2) Use MLE to compute the Gaussian process model parameters associated with the points in Θ.

3) Repeat until $m \geq m^*$ or $\max(\boldsymbol{\delta}) \leq \delta^*$:

a) Use the Gaussian process model built with the points in Θ to predict the $m_t - m$ points in the set Ω. Store the absolute values of these prediction errors in the vector $\boldsymbol{\delta}$.

b) Transfer the point with the maximum prediction error from Ω to Θ.

c) For the current subset Θ, estimate the Gaussian process model parameters using MLE.

As an example, we build a Gaussian process model for the two-dimensional Rosenbrock function

$$f(x_1, x_2) = (1 - x_1)^2 + 100\left(x_2 - x_1^2\right)^2$$

on the usual bounds $-2 \leq x_1 \leq 2$ and $-2 \leq x_2 \leq 2$. We randomly generate a set of 10,000 points within these bounds, and we use the greedy point-selection algorithm to choose a subset of $m = 35$. The resulting maximum prediction error is 1.58×10^{-2}, with a median prediction error of 2.87×10^{-3}. The five random initial points, along with the remaining selected points, are plotted in Fig. 1. The convergence of the maximum prediction error is plotted with a semilog scale in Fig. 2.

This example clearly shows the power of Gaussian process modeling for data interpolation. From Fig. 1, it is obvious that the point-selection algorithm tends to pick points on the boundary of the original set. This is expected and is because the Gaussian process model needs these points to maintain accuracy over the entire region. Only a relatively small number of points are needed at the interior, because of the interpolative accuracy of the model.

It is also interesting to note that the decrease in maximum prediction error is not strictly monotonic. Adding some points may actually worsen the predictive capability of the Gaussian process model in other regions of the parameter space. Nevertheless, we still expect the overall trend to show a decrease in maximum prediction error, at least until matrix ill-conditioning issues start coming into play.

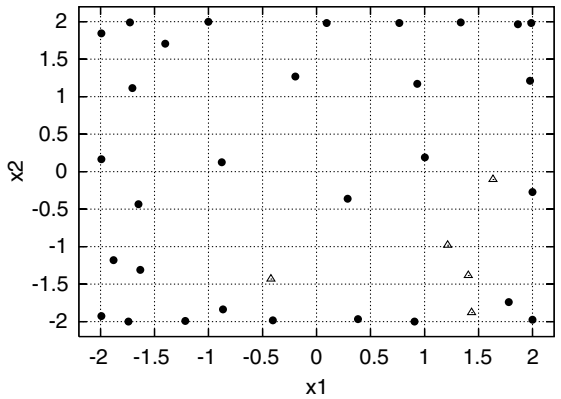

Fig. 1 Initial (triangles) and selected (circles) points chosen by the greedy algorithm with Rosenbrock's function.

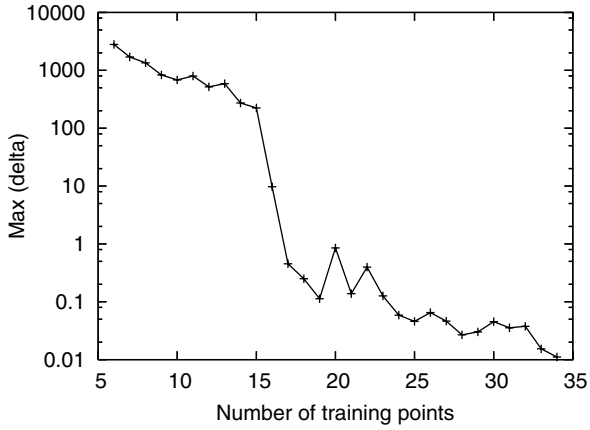

Fig. 2 Semilog plot of maximum prediction error versus m for Rosenbrock's function.

III. Bayesian Model Calibration

Model calibration is a particular type of inverse problem in which one is interested in finding values for a set of computer model inputs that result in outputs that agree well with observed data. As mentioned in Sec. I, one of the most straightforward approaches is to pose the calibration problem in terms of nonlinear least-squares optimization. However, this approach has several drawbacks:

1) Finding the set of model inputs that minimizes the sum of squares may require a large number of evaluations of the model (depending on the type of optimization algorithm being employed). When the model is very expensive to run, this approach may not even be feasible.

2) There may be a wide range of model inputs that provide fits comparable with the observed data (this is sometimes termed the *problem of uniqueness*) [25].

3) Small changes in some of the model inputs may cause drastic variations of the model output, resulting in an ill-posed optimization problem [25].

Further, approaching the calibration problem as a least-squares optimization problem will yield only one solution, and it can be difficult to construct meaningful information about the uncertainty associated with this solution (although some approaches have been attempted: for example, that of [3]). Thus, there would be a large amount of utility in any method that overcomes the difficulties associated with the nonlinear least-squares approach and that provides a more comprehensive treatment of the uncertainties present. Fortunately, the field of Bayesian analysis provides such a method.

The fundamental concept of Bayesian analysis is that unknown variables are treated as random variables. The power of this approach is that the established mathematical methods of probability theory can then be applied. Uncertain variables are given prior probability distribution functions, and these distribution functions are refined based on the available data, so that the resulting posterior distributions represent the new state of knowledge, in light of the observed data. Although the Bayesian approach can be computationally intensive in many situations, it is attractive because it provides a comprehensive treatment of uncertainty.

Bayesian analysis is founded on Bayes's theorem, which is a fundamental relationship among conditional probabilities. For continuous variables, Bayes's theorem is expressed as

$$f(\boldsymbol{\theta} \mid \boldsymbol{d}) = \frac{\pi(\boldsymbol{\theta}) f(\boldsymbol{d} \mid \boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta}) f(\boldsymbol{d} \mid \boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{11}$$

where $\boldsymbol{\theta}$ is the vector of unknowns, $\boldsymbol{d}$ contains the observations, $\pi(\boldsymbol{\theta})$ is the prior distribution, $f(\boldsymbol{d} \mid \boldsymbol{\theta})$ is the likelihood, and $f(\boldsymbol{\theta} \mid \boldsymbol{d})$ is the posterior distribution. Note that the likelihood is commonly written $L(\boldsymbol{\theta})$ because the data in $\boldsymbol{d}$ hold a fixed value once observed.

The primary computational difficulty in applying Bayesian analysis is the evaluation of the integral in the denominator of Eq. (11), particularly when dealing with multidimensional unknowns. When closed-form solutions are not available, computational sampling techniques such as MCMC sampling are often used. In particular, this work employs the componentwise scheme [26] of the Metropolis algorithm [27,28], as outlined in the Appendix.

A. Bayesian Analysis for Model Calibration

Consider that we are interested in making inference about a set of computer model inputs $\boldsymbol{\theta}$. Now let the simulation be represented by the forward-model operator $G(\boldsymbol{\theta}, \boldsymbol{s})$, where the vector of inputs $\boldsymbol{s}$ represents a set of scenario-descriptor inputs, which may typically represent boundary conditions, initial conditions, geometry, etc. Kennedy and O'Hagan [10] termed these inputs *variable inputs*, because they take on different values for different realizations of the

system. Thus, $y = G(\boldsymbol{\theta}, \boldsymbol{s})$ is the response quantity of interest associated with the simulation. Also, we assume that the value of the calibration inputs $\boldsymbol{\theta}$ should not depend on $\boldsymbol{s}$, the particular realization of the system being modeled [10].

Now consider a set of n experimental measurements $\boldsymbol{d} = d_1, \ldots, d_n$, which are to be used to calibrate the simulation. Note that each experimental measurement corresponds to a particular value of the scenario-descriptor inputs $\boldsymbol{s}$, and we assume that these values are known for each experiment. Thus, we are interested in finding those values of $\boldsymbol{\theta}$ for which the simulation outputs $G(\boldsymbol{\theta}, \boldsymbol{s}_1), \ldots, G(\boldsymbol{\theta}, \boldsymbol{s}_n)$ agree well with the observed data in $\boldsymbol{d}$. But, as previously mentioned, we are interested in more than simply a point estimate for $\boldsymbol{\theta}$: we would like a comprehensive assessment of the uncertainty associated with this estimate.

First, we define a probabilistic relationship between the model output $G(\boldsymbol{\theta}, \boldsymbol{s})$ and the observed data $\boldsymbol{d}$:

$$d_i = G(\boldsymbol{\theta}, \boldsymbol{s}_i) + \varepsilon_i \tag{12}$$

where ε_i is a random variable that can encompass both measurement errors on d_i and modeling errors associated with the simulation $G(\boldsymbol{\theta}, \boldsymbol{s})$. The most frequently used assumption for the ε_i is that they are independent and identically distributed (i.i.d.) $N(0, \sigma^2)$, which means that the ε_i are independent zero-mean Gaussian random variables with variance σ^2. Of course, more complex models may be applied: for instance, enforcing a parametric dependence structure among the errors [29].

The probabilistic model defined by Eq. (12), along with the i.i.d error model, results in a likelihood function for $\boldsymbol{\theta}$ that is the product of n normal probability density functions:

$$L(\boldsymbol{\theta}) = f(\boldsymbol{d} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(d_i - G(\boldsymbol{\theta}, \boldsymbol{s}_i))^2}{2\sigma^2} \right] \tag{13}$$

We can now apply Bayes's theorem (11) using the likelihood function (13), along with a prior distribution for $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$, to compute the posterior distribution $f(\boldsymbol{\theta} \mid \boldsymbol{d})$, which represents our belief about $\boldsymbol{\theta}$ in light of the data $\boldsymbol{d}$:

$$f(\boldsymbol{\theta} \mid \boldsymbol{d}) \propto \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}) \tag{14}$$

The posterior distribution for $\boldsymbol{\theta}$ represents the complete state of knowledge and may even include effects such as multiple modes, which would represent multiple competing hypotheses about the true (best-fitting) value of $\boldsymbol{\theta}$. Summary information can be extracted from the posterior distribution, including the mean (which is typically taken to be the best-guess point estimate) and standard deviation (a representation of the amount of residual uncertainty). We can also extract one- or two-dimensional marginal distributions, which simplify visualization of the features of the posterior distribution.

However, as previously mentioned, the posterior distribution cannot usually be constructed analytically, and this will almost certainly not be possible when a complex simulation model appears inside the likelihood function. MCMC sampling is considered here, but this requires hundreds of thousands of evaluations of the likelihood function, which equates to hundreds of thousands of evaluations of the computer model $G(\cdot, \cdot)$ in the case of model calibration. For most realistic models, this number of evaluations will not be feasible. In such situations, the analyst must usually resort to the use of a more inexpensive surrogate (also known as a response surface approximation) model. Such a surrogate might involve reduced-order modeling (e.g., a coarser mesh) or data-fit techniques such as Gaussian process (also known as kriging) modeling.

This work adopts the approach of using a Gaussian process surrogate to the true simulation. We find such an approach to be an attractive choice for use within the Bayesian calibration framework for several reasons:

1) The Gaussian process model is very flexible and can be used to fit data associated with a wide variety of functional forms.

2) The Gaussian process model is stochastic, thus providing both an estimated response value and an uncertainty associated with that estimate. Conveniently, the Bayesian framework allows us to take account of this uncertainty.

3) With regard to fit accuracy, the Gaussian process model has been shown to be competitive with most other modern data-fit methods, including Bayesian neural networks and multiple adaptive regression splines [22,30].

For model calibration with an expensive simulation, the uncertainty associated with the use of a Gaussian process surrogate can be accounted for through the likelihood function. There are a couple of possible approaches for doing so:

1) Treat the parameters governing the Gaussian process surrogate as objects of Bayesian inference, along with the calibration inputs. Thus, they are given a prior distribution and allowed to develop a posterior distribution based on both the observed simulator outputs and the experimental data.

2) Estimate the parameters of the Gaussian process surrogate a priori using the observed code runs. These parameters are then treated as constant known values for the remainder of the analysis. The direct-variance estimates provided by the Gaussian process model can still be incorporated into the calibration analysis.

The first, more complete, approach is outlined in detail by Kennedy and O'Hagan [10]. By treating the Gaussian process parameters as unknowns, the uncertainty that arises because these parameters must be estimated from the data is taken into account. However, this approach is computationally demanding, and it is often difficult to specify appropriate prior distributions for these parameters. For these reasons, Kennedy and O'Hagan [10] suggested that the second, simpler, approach should be used and that doing so does not have a significant effect on the resulting uncertainty analysis. For our work, we adopt the second approach, and the parameters are estimated a priori using the method of maximum likelihood, as discussed in Sec. II.A.

Through the assumptions used for Gaussian process modeling, the surrogate response that is conditional on a set of observed training points follows a multivariate normal distribution. For a discrete set of new inputs, this response is characterized by a mean vector and a covariance matrix [see Eqs. (4) and (6)]. Let us denote the mean vector and covariance matrix corresponding to the inputs $(\boldsymbol{\theta}, \boldsymbol{s}_1), \ldots, (\boldsymbol{\theta}, \boldsymbol{s}_n)$ as $\boldsymbol{\mu}_{\mathrm{GP}}$ and $\boldsymbol{\Sigma}_{\mathrm{GP}}$, respectively. It is easy to show that the likelihood function for $\boldsymbol{\theta}$ is then given by a multivariate normal probability density function (note that the likelihood function of Eq. (13) can also be expressed as a multivariate normal probability density, with $\boldsymbol{\Sigma}$ diagonal):

$$L(\boldsymbol{\theta}) = (2\pi)^{-n/2} |\boldsymbol{\Sigma}|^{-1/2} \exp[-\tfrac{1}{2}(\boldsymbol{d} - \boldsymbol{\mu}_{\mathrm{GP}})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{d} - \boldsymbol{\mu}_{\mathrm{GP}})] \tag{15}$$

where $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I} + \boldsymbol{\Sigma}_{\mathrm{GP}}$, so that both $\boldsymbol{\mu}_{\mathrm{GP}}$ and $\boldsymbol{\Sigma}$ depend on $\boldsymbol{\theta}$.

Simply put, because the uncertainty associated with the surrogate model is independent of the modeling and observation uncertainty captured by the ε_i, the covariance of the Gaussian process predictions ($\boldsymbol{\Sigma}_{\mathrm{GP}}$) simply adds to the covariance of the error terms ($\sigma^2 \boldsymbol{I}$). As mentioned before, if a more complicated error model is desired (i.e., one that does not assume the errors to be independent of each other), we can replace $\sigma^2 \boldsymbol{I}$ by a full covariance matrix.

Also, in some cases in which there is a large amount of experimental data available, we may even want to model different segments of the output (e.g., different spatial locations or different time intervals) using separate, independent, Gaussian process surrogates. In such a case, the likelihood function is a product of multivariate normal densities, in which each density contains a particular partition of $\boldsymbol{d}$ and the corresponding surrogate predictions $\boldsymbol{\mu}_{\mathrm{GP}}$ and $\boldsymbol{\Sigma}_{\mathrm{GP}}$. Such a formulation may improve the accuracy of and decrease the uncertainty in the surrogates because they are more localized, but the implementation is somewhat more cumbersome.

1. *Prescribed Input Uncertainties*

In some cases, it may be of interest to study how the results of a calibration analysis are affected when additional simulator inputs are subject to uncertainty. In most cases, we would do so in the Bayesian setting by augmenting the set of calibration parameters $\boldsymbol{\theta}$ with the additional uncertain model inputs. If the data $\boldsymbol{d}$ do not provide any

information about these additional uncertain inputs, then they will essentially be sampled over their prior distribution, potentially resulting in an increase in the uncertainty in the original calibration parameters. On the other hand, if the data $\boldsymbol{d}$ do provide information about the additional inputs, then their posterior distribution will reflect less uncertainty than their prior distribution. However, if we are strictly interested in the effect of additional prescribed input uncertainties, such inputs cannot be treated as calibration inputs, because their posterior distribution may not match the prescribed distribution of interest. Thus, this section presents a method that allows the Bayesian calibration analysis to take into account the prescribed uncertainties for additional model inputs.

Let us denote by $\boldsymbol{\xi}$ those inputs to the simulation $G(\cdot)$ that have prescribed probability distributions. Thus, our simulation model is now a function of the calibration inputs, the scenario-descriptor inputs, and the inputs with prescribed distributions: $y = G(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{\xi})$. Denote the probability density function associated with $\boldsymbol{\xi}$ by $f(\boldsymbol{\xi})$. To develop the posterior distribution for $(\boldsymbol{\theta}, \boldsymbol{\xi})$ in which the distribution of $\boldsymbol{\xi}$ is not refined by $\boldsymbol{d}$, we must assume artificially that the data $\boldsymbol{d}$ are statistically independent of $\boldsymbol{\xi}$. Whether or not this is true in reality can be checked by treating $\boldsymbol{\xi}$ as a calibration parameter in $\boldsymbol{\theta}$; but by artificially enforcing the assumption, the parameters $\boldsymbol{\xi}$ are held to the prescribed distribution $f(\boldsymbol{\xi})$.

By assuming that $\boldsymbol{\xi}$ is independent of $\boldsymbol{d}$, we have

$$f(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{d}) \propto \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}) f(\boldsymbol{\xi})$$

Because the simulation output is a function of $\boldsymbol{\xi}$, $L(\boldsymbol{\theta})$ is as well, and so for clarity, we write $L(\boldsymbol{\theta}; \boldsymbol{\xi})$,[¶] which yields

$$f(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{d}) \propto \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \boldsymbol{\xi}) f(\boldsymbol{\xi}) \tag{16}$$

Ultimately, though, we are interested in the posterior distribution of $\boldsymbol{\theta}$ after marginalizing over the nuisance variable $\boldsymbol{\xi}$, and so we want

$$f(\boldsymbol{\theta} \mid \boldsymbol{d}) \propto \int \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \boldsymbol{\xi}) f(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{17}$$

This marginalization is trivial if $f(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{d})$ is constructed using Markov chain Monte Carlo sampling. One possibility for constructing $f(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{d})$ is to use a componentwise scheme to sequentially sample each component of $(\boldsymbol{\theta}, \boldsymbol{\xi})$ from its respective full conditional distribution. Each component of $\boldsymbol{\theta}$ can be sampled using the Metropolis algorithm (see the Appendix), by sampling the ith component from its full conditional distribution:

$$f(\theta_i \mid \boldsymbol{\theta}_{-i}, \boldsymbol{\xi}, \boldsymbol{d}) \propto \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \boldsymbol{\xi}) \tag{18}$$

where $\boldsymbol{\theta}_{-i}$ contains all components of $\boldsymbol{\theta}$ except for θ_i. Notice that $f(\boldsymbol{\xi})$ does not appear in Eq. (18) because it does not depend on $\boldsymbol{\theta}$.

Further, if the joint distribution of $\boldsymbol{\xi}$ is such that it can be sampled (in particular, if the components of $\boldsymbol{\xi}$ are independent and can be sampled), the vector $\boldsymbol{\xi}$ can be directly sampled at each iteration. This is because the full conditional distribution of $\boldsymbol{\xi}$ is equal to $f(\boldsymbol{\xi})$, so at each iteration we draw a sample of $\boldsymbol{\xi}$ from $f(\boldsymbol{\xi})$, which is its full conditional distribution.

In short, the process of accounting for prescribed input uncertainties within the Bayesian calibration analysis is very simple, given that Markov chain Monte Carlo is used to construct the posterior distribution for $\boldsymbol{\theta}$. To account for the additional total uncertainty introduced by the inputs $\boldsymbol{\xi}$ having prescribed uncertainties, we simply sample a random realization of $\boldsymbol{\xi}$ at each iteration of the MCMC sampler.

2. *Characterized Observation and Modeling Uncertainty*

In the probabilistic error model of Eq. (12), ε is a random variable that encompasses both observation uncertainty in the data $\boldsymbol{d}$ and modeling error: together, these effects result in a difference between the observations and the predictions. In most cases, the overall magnitude of this net effect (represented by the variance of ε, σ^2) is not known, and σ^2 is treated as an object of Bayesian inference, along with the calibration inputs $\boldsymbol{\theta}$. However, in some cases, the experimental instrumentation may be understood well enough that the error associated with the observed data $\boldsymbol{d}$ can be characterized using a parametric probability distribution. For example, the experimenter might claim that the errors in the measurements $\boldsymbol{d}$ follow a Gaussian distribution with zero-mean and standard deviation equal to 10% of the measured value.

Similarly, the error associated with the analysis code $G(\cdot)$ might also be characterized as a random quantity in some cases. For example, based on a mesh convergence study, an analyst may be able to quantitatively characterize the magnitude of the error associated with the output of $G(\cdot)$, which might, for example, be used to derive a probabilistic representation for that error.

When the error/uncertainty associated with the observations and/or analysis code can be characterized, we would like to include it in our probabilistic model. In most cases, we would still want to retain a separate ε term, which would represent all other sources of error that lead to a difference between the predictions and observations. Thus, we might formulate a new probabilistic model as

$$d_i = G(\boldsymbol{\theta}, \boldsymbol{s}_i) + \varepsilon_i + u_i \tag{19}$$

where the random variable u_i represents the characterized uncertainty associated with either the observation d_i or the analysis code output $G(\boldsymbol{\theta}, \boldsymbol{s}_i)$. Note that the additive-error model of Eq. (19) is used here only as an illustration, and this formulation is not necessarily a requirement.

For simple cases in which both ε and u are Gaussian random variables, we can replace both of them with one random variable that is their sum, and it will be Gaussian as well. However, although ε is most often taken to be Gaussian, other distributions might be chosen for u. For example, the experimenter might characterize the measurement uncertainty with bounds, in which case it would be most appropriate to use a uniform probability distribution for u. In such cases, it is difficult to analytically express the probability distribution of the sum $\varepsilon + u$, and alternative methods may be more prudent.

One possibility is to use the same approach that was taken in Sec. III.A.1 and sample $\boldsymbol{u}$ along with $\boldsymbol{\theta}$. First, let us denote the joint probability density function for $\boldsymbol{u} = (u_1, \ldots, u_n)$ by $f(\boldsymbol{u})$. Then, analogously to Eq. (16), we have

$$f(\boldsymbol{\theta}, \boldsymbol{u} \mid \boldsymbol{d}) \propto \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}; \boldsymbol{u}) f(\boldsymbol{u}) \tag{20}$$

If we consider the case in which $\boldsymbol{u}$ represents characterized observation uncertainty, we see that the likelihood function for $\boldsymbol{\theta}$ depends on $\boldsymbol{u}$ in the sense that after subtracting the effect of u_i, the observation is actually given by $d_i - u_i$. That is, the likelihood function of Eq. (13) would become

$$L(\boldsymbol{\theta}; \boldsymbol{u}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(d_i - u_i - G(\boldsymbol{\theta}, \boldsymbol{s}_i))^2}{2\sigma^2} \right] \tag{21}$$

Thus, as outlined in Sec. III.A.1, the approach is to sample a random realization of $\boldsymbol{u}$ from $f(\boldsymbol{u})$ at each iteration of the MCMC sampler. Then, before computing $L(\boldsymbol{\theta})$, we artificially perturb the observed data as $\boldsymbol{d} - \boldsymbol{u}$.

B. Summary of the Proposed Approach

In summary, a variety of new ideas were proposed in Secs. II and III to enhance the Bayesian framework for model calibration under uncertainty. First, we presented a simple point-selection algorithm applicable to Gaussian process surrogate modeling that is particularly useful for modeling code output that is a function of time or space. We also outlined two methods for expanding on the uncertainty-estimation capabilities of the Bayesian calibration framework. The first method, discussed in Sec. III.A.1, illustrates the

[¶]Although it is tempting to write $L(\boldsymbol{\theta}, \boldsymbol{\xi})$, we avoid doing so, because this is really $f(\boldsymbol{d} \mid \boldsymbol{\theta}, \boldsymbol{\xi})$; because $\boldsymbol{\xi}$ is (assumed to be) statistically independent of $\boldsymbol{d}$, this would reduce to $f(\boldsymbol{d} \mid \boldsymbol{\theta}) = L(\boldsymbol{\theta})$. Thus, we write $L(\boldsymbol{\theta}; \boldsymbol{\xi})$ to emphasize that it is a function of $\boldsymbol{\xi}$, but there is no statistical relationship between $\boldsymbol{\xi}$ and $\boldsymbol{d}$.

procedure through which one can study the effects on the calibration parameters of prescribed uncertainties on additional model inputs. Finally, Sec. III.A.2 illustrates a procedure for accounting for characterized observation of modeling uncertainty. This characterized uncertainty may be accounted for, in addition to other uncharacterized effects that produce a difference between predictions and observations, even when different distributional forms are desired for the characterized and uncharacterized effects.

In what follows, Sec. IV presents a case study that illustrates all of the proposed ideas. The case study involves the Bayesian calibration of a computer simulation that models the thermal response of a decomposing foam.

IV. Case Study: Thermal Simulation of Decomposing Foam

A series of experiments have been conducted at Sandia National Laboratories in an effort to support the physical characterization and modeling of thermally decomposing foam [31]. An associated thermal model is described in [32]. The system considered here, often referred to as the *foam-in-a-can system*, consists of a canister containing a mock weapons component encapsulated by a foam insulation. Several illustrations of this setup are shown in Figs. 3 and 4.

The simulation model is a finite element model (FEM) developed for simulating heat transfer through decomposing foam. The model contains roughly 81,000 hexahedral elements and has been verified to give spatially and temporally converged temperature predictions. The heat-transfer model is implemented using the massively parallel code CALORE [33], which was developed at Sandia National Laboratories under the advanced simulation and computing program of the National Nuclear Security Administration.

The simulator was configured to model the foam-in-a-can experiment, but several of the input parameters are still unknowns (either not measured or not measurable). In particular, we consider five calibration parameters: q_2, q_3, q_4, q_5, and the final foam pore diameter FPD. The parameters q_2 through q_5 describe the applied heat-flux boundary condition, which is not well-characterized in the experiments. The last calibration parameter, FPD, is the parameter of most interest, because it will play a role in the ultimate modeling and prediction process. We want to consider the calibration of the simulator for the temperature response up to 2200 s at nine different locations on the structure (six external and three internal).

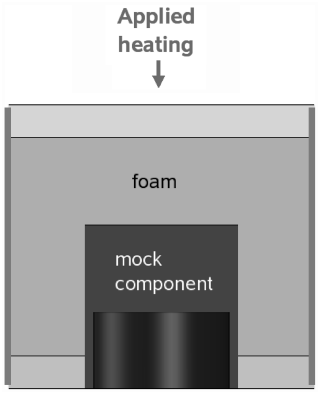

Fig. 3 Schematic of the foam-in-a-can system.

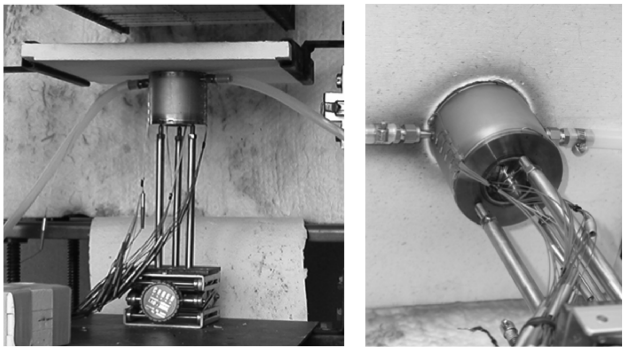

Fig. 4 Experimental setup.

A. Preliminary Analysis

The first step is to collect a database of simulator runs for different values of the calibration parameters, from which the surrogate model will be constructed. Ideally, we would like our design of computer experiments to provide good coverage for the posterior distribution of the calibration inputs. However, because we do not know the form of the posterior distribution beforehand, we have to begin with an initial guess for the appropriate bounds. Fortunately, the Bayesian method provides feedback, and so if our original bounds are not adequate, they can be revised appropriately. This type of sequential approach has previously been used for Bayesian model calibration and other studies [10,34–36].

We make use of the Design Analysis Kit for Optimization and Terrascale Applications (DAKOTA) [37] software package for our design and collection of computer experiments. DAKOTA is an object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis that can be configured to interface with the thermal simulator via external file input/output and a driver script. For our initial design, we use the DAKOTA software package to generate an LH sample of size 50 using the variable bounds listed in Table 1.

The Bayesian calibration using these bounds illustrates that some adjustment to the bounds would be useful, because the resulting posterior distribution directly indicates which regions of the parameter space are feasible, including whether the parameter space should be expanded in the subsequent design. Thus, we construct a new LH sample of size 50 using the revised design described in Table 2. The revised bounds are chosen so that they will cover the entire range of the posterior distribution for the calibration inputs.

Using the results from the simulation runs, we can compare the ensemble of predicted time histories against the experimental time histories to see if the experimental data are enveloped by the simulation data. Figures 5 and 6 compare the envelope of simulator outputs against the experimental data for locations 1 and 9, respectively. In general, the experimental observations are enveloped by the simulator outputs, although at locations 5 and 6, the experimental response exceeds the maximum of the simulator outputs for $t < 800$ s, as seen in Fig. 7.

B. Bayesian Calibration Analysis: Nominal Case

Here, we consider the nominal Bayesian calibration of the CALORE simulator using data from all nine locations of interest. Some of these locations (for example, location 1) are averages of

Table 1 Original design of computer experiments

Variable	Lower bound	Upper bound
FPD	2.0×10^{-3}	15.0×10^{-3}
q_2	25,000	150,000
q_3	100,000	220,000
q_4	150,000	300,000
q_5	50,000	220,000

Table 2 Revised design of computer experiments

Variable	Lower bound	Upper bound
FPD	4.0×10^{-3}	6.0×10^{-3}
q_2	25,000	150,000
q_3	0	200,000
q_4	100,000	400,000
q_5	120,000	160,000

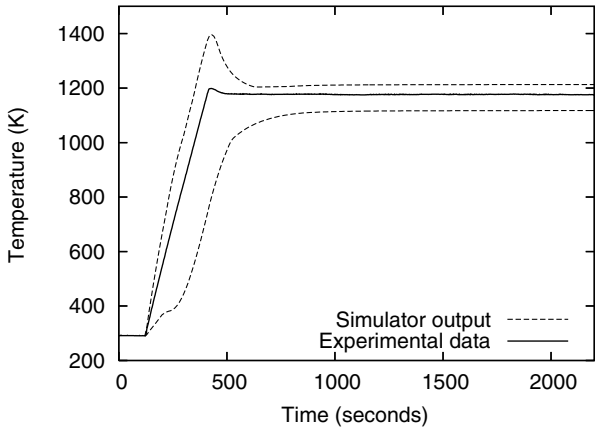

Fig. 5 Temperature-response comparison for envelope of 50 simulator outputs with observed data for location 1 (average lid temperature).

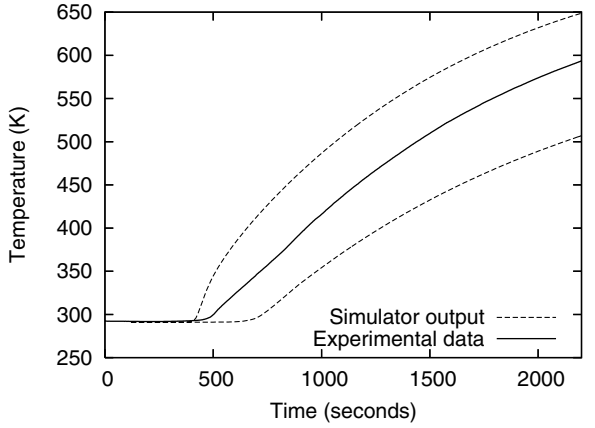

Fig. 6 Temperature-response comparison for envelope of 50 simulator outputs with observed data for location 9 (internal thermocouple).

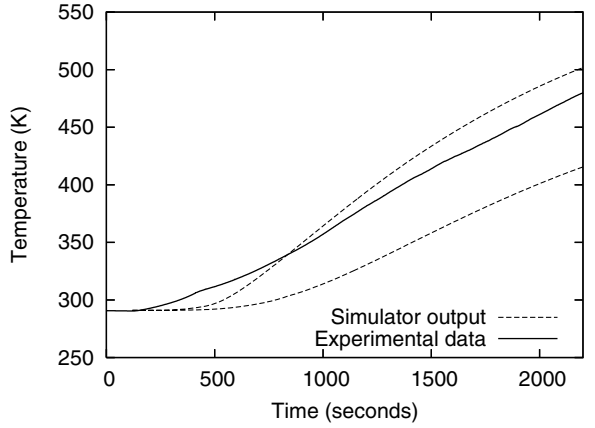

Fig. 7 Temperature-response comparison for envelope of 50 simulator outputs with observed data for location 6 (average of thermocouples 13 through 16).

multiple thermocouple readings and others represent single thermocouple readings. The application of the Bayesian calibration extensions discussed in Secs. III.A.1 and III.A.2 will be presented in Secs. IV.C and IV.D.

The variance of ε in Eq. (12), σ^2, is not considered as a function of time or location. It would be straightforward to incorporate a parametric dependence for the variance on temporal or spatial coordinates if such a formulation were desired. We do, however, treat σ^2 as an object of Bayesian inference, making use of the standard reference prior distribution [38]:

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2} \tag{22}$$

For the prior distributions of the calibration parameters, we choose independent uniform distributions based on the bounds given in Table 1 for the initial analysis, and after revising the design of computer experiments, the prior bounds are adjusted to reflect those listed in Table 2.

Each of the nine locations are modeled separately, with two independent surrogates representing the response before and after 500 s, which results in a total of 18 surrogate models for the simulator output. We employ the use of multiple Gaussian process surrogate models because a single stationary Gaussian process representation of the response at all locations and time instances does not seem to be appropriate. Our choice of dividing the surrogates at 500 s is admittedly subjective (and a more comprehensive approach might choose different time divisions for different locations), but on average for the different locations, there is a significant change in the response behavior around 500 s: for example, the process variance increases (see Figs. 5–7).

For each surrogate, we employ the point-selection algorithm discussed in Sec. II.B to select an optimal subset of points with which to build the surrogate. At each location, the first surrogate is based on 75 points chosen optimally from the 1950 available points (39 time instances times 50 samples), and the second is based on 100 points chosen optimally from 8550 points. We emphasize that the process for constructing these surrogate models is not trivial: for each of 18 separate surrogate models, we employ the iterative MLE process described in Sec. II.B. This results in approximately 3000 numerical MLE optimization problems in six dimensions, which is why an efficient MLE scheme is critical, and the use of gradient information, as discussed in Sec. II.A, can be very important.

For the experimental data, we use 21 points evenly spaced at time intervals of 100 s for each of the nine locations. The MCMC simulation is adjusted appropriately and run for 100,000 iterations. The resulting marginal posterior distributions for the two parameters of most interest, FPD and q_5, are shown in Figs. 8 and 9, in which the plotting ranges are representative of the bounds of the prior

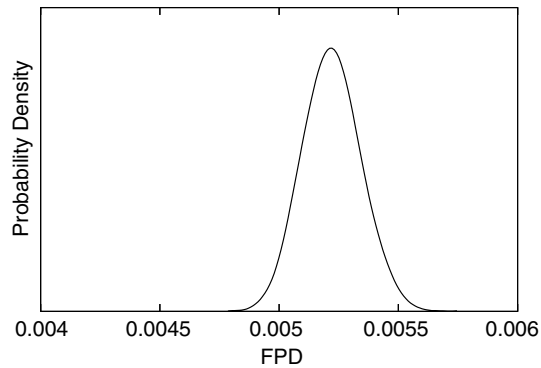

Fig. 8 Posterior distribution of FPD (x range represents prior bounds).

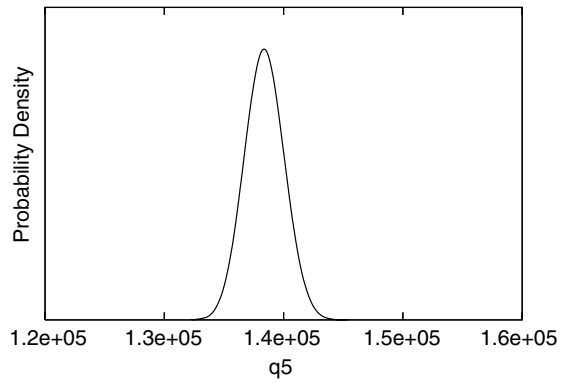

Fig. 9 Posterior distribution of q_5 (x range represents prior bounds).

Table 3 Posterior statistics based on the nominal calibration analysis

Variable	Mean	Std. Dev.
FPD	5.22×10^{-3}	1.17×10^{-4}
q_2	88,546	16,977
q_3	113,100	11,307
q_4	246,270	11,652
q_5	138,390	1,565

Table 4 Pairwise correlation coefficients within the posterior distribution for nominal analysis

	FPD	q_2	q_3	q_4	q_5
FPD	1.00	0.02	0.02	−0.25	−0.67
q_2	0.02	1.00	−0.80	0.18	−0.02
q_3	0.02	−0.80	1.00	−0.58	−0.01
q_4	−0.25	0.18	−0.58	1.00	0.00
q_5	−0.67	−0.02	−0.01	0.00	1.00

distribution. Recall that the prior distribution for $\boldsymbol{\theta}$ is independently uniform over the ranges listed in Table 2.

The statistics of the marginal posteriors are given in Table 3, and the pairwise correlation coefficients are given in Table 4. The correlation coefficients indicate a strong negative relationship between q_2 and q_3, as well as moderate negative relationships between FPD and q_5 and between q_3 and q_4. For a more visual interpretation of these relationships, we can use kernel-density estimation [39] to visualize the two-dimensional density functions. For example, Fig. 10 plots the 95% plausibility region for FPD and q_5 based on a kernel-density estimate to the two-dimensional posterior distribution of these two variables.

It is also possible to define a simple error measure that allows for the quantification of the agreement of the simulator output to the experimental data. An intuitive error measure is the sum of the squared errors or, equivalently, the square root of the mean of the squared errors (rms, used here because it has the same units as the response quantity). The rms error between the experiments and the predictions is defined as

$$\text{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - G(\boldsymbol{\theta}, \boldsymbol{s}_i))^2} \tag{23}$$

To consider the accuracy of the Bayesian estimate when only a small number of simulator runs are available, we consider the posterior mean based on the analysis with the original bounds on the calibration parameters (Table 1), which corresponded to only 50 runs of the simulator. The rms agreement with the experimental data for this case is 19.4 K.

We now derive an alternative estimate to the calibration parameters, which we use as a baseline for comparison with the Bayesian mean estimate. For the baseline estimate, we adopt an admittedly ad hoc nonlinear least-squares approach. We simply construct a deterministic optimization problem in which the design variables are the calibration inputs $\boldsymbol{\theta}$, and the objective function is the rms error measure, as defined in Eq. (23). We apply the dividing-rectangles (DIRECT) [40] global optimization algorithm using convergence criteria that limit the number of objective function evaluations (equivalently, runs of the CALORE simulation) to a number comparable with that used in the Bayesian calibration analysis: 50. To keep the comparison fair, we provide the DIRECT algorithm with the same variable bounds that were available to the Bayesian analysis (the prior bounds, listed in Table 1).

After 65 function evaluations, the DIRECT algorithm reduces the rms error to 32.3 K. What we notice is that although the Bayesian approach provides a comprehensive framework for representing uncertainty in the parameter estimates, this framework is still capable of providing an efficient (in terms of the number of simulator runs) means for obtaining accurate point estimates to the calibration parameters, compared with the alternative nonlinear least-squares optimization approach. We acknowledge that a surrogate-based optimization approach might be preferred to interfacing directly with the expensive simulator, but we do not make such a comparison because we feel it would be too similar (in terms of the resulting point estimates) to the Bayesian approach itself, which has made use of Gaussian process surrogates.

Finally, as a check on the surrogate models, we compute the total rms difference (over all nine locations, with 5-s time increments) between the surrogate output and the true simulator output for the posterior mean of the calibration inputs. This rms difference is found to be only 2.4 K, which suggests that the surrogate has accurately captured the relationship between the simulator inputs and outputs. Figure 11 illustrates how the surrogate compares with the actual simulator output at location 9. The discrepancy is visually almost indistinguishable. The experimental observations were plotted as well, for illustration.

C. Accounting for Characterized Measurement Uncertainty

In this section, we adopt the approach described in Sec. III.A.2 to account for characterized measurement uncertainty associated with the thermocouple readings. We expect this addition to be reflected by a broadening of the posterior distribution of the calibration inputs. In addition, because some of the thermocouples are biased, we also expect to see a shift in the posterior distribution, which accounts for this bias.

The thermocouple reading uncertainty is characterized by bounds, and so we use uniform random variables to represent this uncertainty

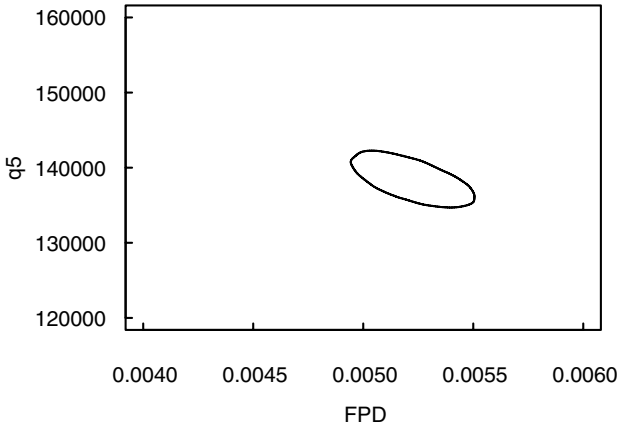

Fig. 10 The 95% plausibility region for FPD and q_5; plotting bounds represent prior bounds.

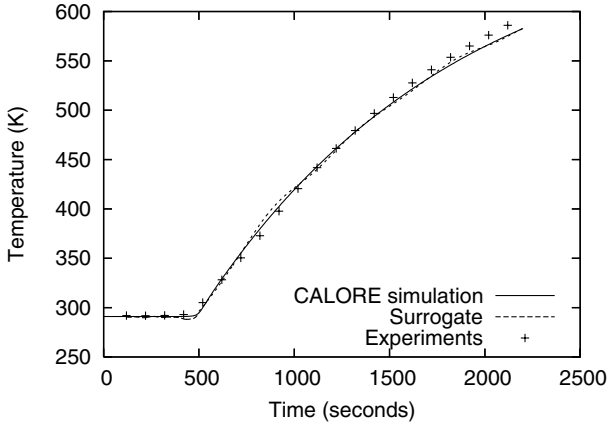

Fig. 11 Comparison of surrogate-model output to actual CALORE output for location 9, based on the posterior mean of the calibration inputs.

(not to be confused with a prior distribution, because the thermocouple error is not an object of Bayesian inference). For the thermocouples on the sides and bottom of the structure (corresponding to locations 2–6), the uncertainty is characterized with bounds $u_i \sim \text{uniform}(-0.02 \times d_i, 0)$, which is a time-dependent percentage of the measured temperature, d_i. As is apparent from Eq. (19), negative values of u correspond to measurements that underestimate the actual value.

For the remaining thermocouples (corresponding to locations 1, 7, 8, and 9), the FEM simulation itself is used to estimate the measurement uncertainty. This is possible because these thermocouples are explicitly modeled in the FEM simulation, along with an associated contact parameter, which represents the amount of contact between the thermocouple and the structure. By varying the contact parameter, we are able to use the simulator to estimate the magnitude of the effect that imperfect contact might have on the thermocouple reading. As a result, the uncertainty for these thermocouples is characterized as $u_i \sim \text{uniform}(-\delta_i, \delta_i)$, where δ_i is the difference between the simulator output for minimum and maximum contact. Note that δ_i varies over the time instances and thermocouple locations.

Also, we note that several of the locations are averages of multiple thermocouple readings. For example, location 1 is the average of four thermocouples mounted on the lid. In these cases, the thermocouple measurement errors average as well, and the generation of random realizations from such averages is handled using simulation.

The resulting statistics of the posterior distribution for this case (based on 100,000 MCMC samples) are reported in Table 5, in which we notice small shifts in the means and small increases in the variance. This is illustrated graphically for FPD and q_5 in Fig. 12, which compares a contour of the posterior density with and without the effect of characterized measurement uncertainty.

The shift in the means for both q_5 and FPD is explainable in terms of the thermocouple uncertainty. Because q_5 represents applied heat flux, it is positively related to temperature response. Similarly, the negative correlation between q_5 and FPD suggests that the final foam pore diameter is also positively related to temperature response. Because the external thermocouples are known to provide readings that underestimate the actual temperature response, we expect that accounting for this bias will result in an increase in the estimates for q_5 and FPD, and this is in fact what we see.

D. Adding Prescribed Input Uncertainties

In this section, we extend the nominal analysis to include additional modeling uncertainties, as discussed in Sec. III.A.1. Although we have considered the calibration of five model inputs thus far, there are actually many additional inputs to the simulator that are subject to uncertainty or lack of knowledge. Here, we study the effect on the calibration results when we treat 13 additional model inputs as having prescribed uncertainties (in this case, simply feasible bounds, represented by uniform probability density functions).

Although it is possible to treat these additional model inputs as calibration parameters, along with the original five, the primary reason for holding their uncertainties fixed is simply because there is an interest in knowing what effect this will have on the results. On the other hand, if they are treated as additional calibration parameters, their prior uncertainties may be reduced in light of the data $\boldsymbol{d}$, which would not give a picture of the effect of the prescribed uncertainties.

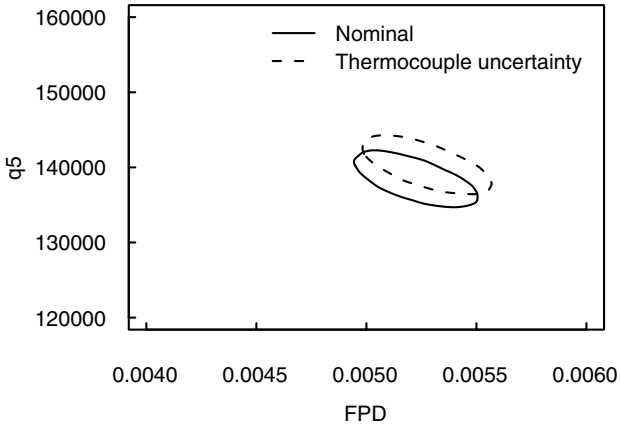

Fig. 12 Comparisons of joint posterior distribution for FPD and q_5 with and without characterized thermocouple uncertainty (95% plausibility regions).

Nevertheless, we conduct each of these analyses, as well as one control analysis, for comparison:

1) To make a fair comparison, we first conduct the analysis while holding the additional uncertain inputs fixed at their mean values. Although conceptually the same as the analysis discussed in Sec. IV.C, the surrogate models are based on a different set of training data, and the surrogates must now model the relationship between the additional 13 inputs and the response, which we expect to result in additional overall uncertainty.

2) Using the method outlined in Sec. III.A.1, we perform the analysis while allowing the additional inputs to vary according to their prescribed uncertainty distributions.

3) For comparison, we also perform the analysis in which the additional 13 inputs are treated as calibration parameters, along with the original five.

The first step is to collect a new set of simulator data, which is necessary because the Gaussian process surrogates must now model the relationship between the temperature response and the 13 new inputs, in addition to the five original calibration inputs. This results in a design of computer experiments over 18 variables and surrogates that are based on 19 inputs (because time is an input to the surrogates). We use a random LH sample of size 50, with the bounds for the original parameters shown in Table 6 (for brevity, the information on the 13 additional parameters is not shown). Generous bounds are used for the calibration parameters, because it is not known how much extra uncertainty will be introduced by the additional uncertain inputs.

With the new code runs, we use the same structure for our surrogates as before: two surrogates (for response before and after 500 s) are used at each of nine locations on the structure, for a total of 18 surrogate models. We emphasize that the surrogates capture the temperature response as a function of time, the five original calibration inputs, and the 13 additional uncertain inputs. We again employ the point-selection process discussed in Sec. II.B, and this time, between 40 and 128 points are used for each surrogate, depending on the complexity of the response.

Each of the three analyses previously described are then conducted. For each case, we use 50,000 MCMC samples to construct the posterior distribution. We note that these analyses are

Table 5 Posterior statistics based on the calibration analysis with characterized measurement uncertainty

Variable	Mean	Std. Dev.
FPD	5.27×10^{-3}	1.24×10^{-4}
q_2	87,477	16,723
q_3	116,900	12,223
q_4	242,680	12,864
q_5	140,290	1,647

Table 6 Design of computer experiments for study with additional prescribed input uncertainties (specifications for additional 13 inputs not listed)

Variable	Lower bound	Upper bound
FPD	2.0×10^{-3}	10.0×10^{-3}
q_2	0	200,000
q_3	0	200,000
q_4	100,000	400,000
q_5	50,000	200,000

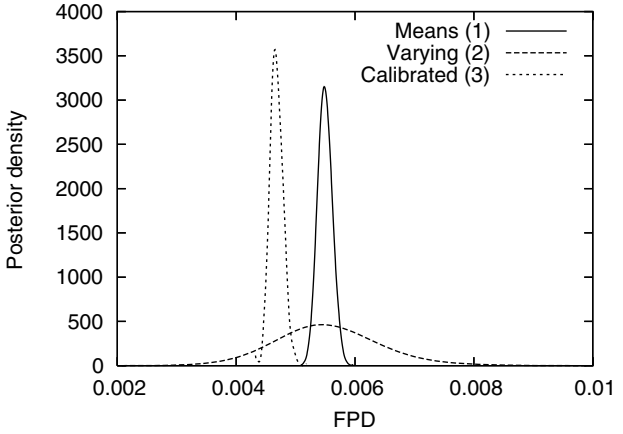

Fig. 13 Comparison of posterior distribution of FPD for each of three approaches for treating the 13 additional uncertain model inputs.

considerably more expensive than those described in Secs. IV.B and IV.C. Of the three, the most expensive is the third case, in which the new inputs are treated as calibration inputs; the computational cost here is high because the MCMC sampler must evaluate the likelihood ratio [see Eq. (15)] once per iteration for each calibration input. Running on a Linux machine with a 64-bit, 2.4-GHz processor, the third analysis took approximately 30 h, and the first two took on the order of 10 h each.

Because the calibration parameter FPD is of most interest for the thermal simulation, we illustrate its marginal posterior distribution in Fig. 13, comparing each of the three analyses previously listed. As expected, the posterior distribution for analysis 1 (holding the additional uncertain parameters fixed to their nominal mean values) is basically the same posterior distribution that was obtained in the nominal analysis described in Sec. IV.B. We also see that allowing the additional parameters to vary on their prescribed uncertainty bounds results in a significant increase in the posterior uncertainty for FPD. Finally, the least amount of uncertainty in FPD is obtained when the additional uncertain parameters are treated as calibration parameters, and this is to be expected as well, because that analysis effectively increases the number of degrees of freedom in the calibration.

The preceding analyses were also redone using 100 LH samples over the 18 variables, to assess whether the results differ significantly from those found using 50 LH samples. It is determined that increasing the number of simulator runs to 100 does not significantly alter the posterior distribution for this analysis.

V. Conclusions

The important role that computational models play in prediction, design, and decision-making necessitates appropriate methods for assessing the uncertainty in such predictions. This work explored the use of Bayesian model calibration as a tool for calibrating a computational simulation with experimental observations while accounting for the uncertainty that is introduced in the process.

One emphasis was on the use of Gaussian process surrogate models. In particular, we proposed an iterative point-selection process that allows one to build efficient Gaussian process surrogates for an analysis code that may have highly multivariate output (for example, time-history response).

Further, we showed how a variety of uncertainties associated with the calibration process can be accounted for in the resulting estimates. This includes uncertainty associated with the use of surrogate models, both characterized and uncharacterized observation and modeling errors, and prescribed input parameter uncertainties.

We applied this methodology to an expensive thermal simulation of a foam-in-a-can system with a database of time-dependent experimental observations. The results illustrate the ability of the Bayesian method to provide a comprehensive representation of the uncertainty present in the resulting parameter estimates, but we also showed that the point estimates obtained from our analysis are not only very efficient (in terms of the number of runs of the FEM simulation), but also very accurate and competitive with other methods that do not provide an uncertainty representation.

There are still many opportunities for further research in this area. Any work that extends the capabilities of surrogate-modeling techniques would be helpful. In particular, practical methods for Gaussian process interpolation with nonstationary covariance would be of interest, as would additional approaches for modeling time-history output. Future work might also more closely consider the transfer of posterior uncertainty to new predictions, particularly when not all of the calibration inputs appear in the simulation for the new configuration.

Appendix: Markov Chain Monte Carlo Sampling

MCMC simulation is a numerical simulation technique that is often used in Bayesian analysis to construct the posterior distribution when no analytical expression is available. The posterior distribution is approximated by generating a list of random samples that form a Markov chain for which the stationary distribution equals the posterior distribution.

The particular MCMC implementation used in this work is known as the Metropolis algorithm and it is a form of rejection sampling. The algorithm is fairly simple to implement and it can be used to generate samples from both univariate and multivariate densities. Consider that one wants to generate samples from a univariate density $f(x)$ that can be evaluated up to a proportionality constant, such that $\tilde{f}(x)$ is known, where $\tilde{f}(x) \propto f(x)$ [in Bayesian inference, $\tilde{f}(x) = \pi(x)L(x)$, as in Eq. (14)]. In this case, the Metropolis algorithm can be implemented as follows:

1) Set $i = 0$ and choose a starting value, x_0.

2) Initialize the list of samples: $\boldsymbol{X} = \{x_0\}$.

3) Repeat the following steps many times:

a) Sample a candidate x^* from the proposal density function $q(x^* \mid x_i)$.

b) Calculate the acceptance ratio

$$\alpha = \min\left[1, \frac{\tilde{f}(x^*)}{\tilde{f}(x_i)}\right]$$

c) Generate a random number u from the uniform distribution on [0,1].

d) If $u < \alpha$, set $x_{i+1} = x^*$, otherwise set $x_{i+1} = x_i$.

e) Augment the list of sampled values $\boldsymbol{X}$ by x_{i+1}.

f) Increment i.

4) After convergence is reached, the list of samples $\boldsymbol{X}$ can be used to construct an approximation to the target density $f(x)$.

The proposal density $q(x^* \mid x_i)$ defines a probability density that generates random moves x^* based on the current point x_i. For the Metropolis algorithm, the only restriction on the choice of proposal density $q(\cdot \mid \cdot)$ is that it be symmetric with respect to its arguments; that is, the probability of going from x_i to x^* is the same as that of going from x^* to x_i. An extension of this algorithm, known as the Metropolis–Hastings algorithm, allows the proposal density to have any form.

Convergence of the chain is generally achieved fairly quickly. However, poor choices for the starting value x_0 may cause the chain to take many samples to reach its stationary distribution. A simple method for assessing convergence is to look at a trace plot of the samples.

When implementing the Metropolis algorithm, the user must only specify the starting value and the proposal density $q(\cdot \mid \cdot)$. Unfortunately, the performance of the algorithm can be sensitive to both of these choices, particularly, the choice of proposal density. The most commonly used proposal density is the random walk density, in which the candidate point is given by $x^* = x_i + \eta$, where η is a random variable chosen to be symmetric about the origin (Gaussian perturbations are employed for this work). The choice of the variance of η is critical to the performance of the algorithm. If the

moves are very small and the acceptance probability is very high, most moves will be accepted, but the chain will take a large number of iterations to converge. If the moves are large, they are likely to fall in the tails of the posterior distribution and result in a low value of the acceptance ratio. One wants to cover the parameter space in a computationally efficient fashion. Some studies have been done on optimal acceptance rates, and the results seem to indicate that 0.45–0.5 is the optimal acceptance rate for one-dimensional problems, whereas 0.23–0.25 is the optimal acceptance rate for high-dimensional problems [41].

When the target density is multivariate, one possible extension of the preceding algorithm is the componentwise scheme discussed by [26,28]. With the componentwise scheme, the components of $\boldsymbol{x}$ are sampled sequentially from their full conditional densities, such that the acceptance ratio for the jth component is computed as

$$\alpha_j = \min\left[1, \frac{f(x_j^* \mid \boldsymbol{x}_{-j})}{f(x_j \mid \boldsymbol{x}_{-j})}\right]$$

where $\boldsymbol{x}_{-j}$ contains all elements of $\boldsymbol{x}$ except for x_j. By cancellation of the marginal density for $\boldsymbol{x}_{-j}$, the acceptance ratio is expressed more conveniently in terms of a joint density proportional to the target density:

$$\alpha_j = \min\left[1, \frac{\tilde{f}(x_j^*, \boldsymbol{x}_{-j})}{\tilde{f}(x_j, \boldsymbol{x}_{-j})}\right]$$

where $\tilde{f}(\cdot)$ may take the form of expression (14).

The nature of MCMC sampling is such that the samples obtained in this fashion will almost always show a strong degree of serial correlation. For this reason, one generally makes inference about the posterior distribution using a very large number of samples (typically, on the order of 10,000 or more). The nature of the resulting Markov chain should also be kept in mind if one wants to generate a small random sample from the posterior distribution. For example, if the posterior distribution is simulated using 20,000 MCMC samples, and 100 random samples from the posterior distribution are needed, it would not be appropriate to take 100 consecutive samples from the chain. Instead, one might either choose the samples from the chain at evenly spaced intervals of 200 or randomly from the 20,000 available.

Acknowledgments

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL8500. This study was partly supported by funds from the National Science Foundation, through the Integrative Graduate Education and Research Traineeship (IGERT) multidisciplinary doctoral program in risk and reliability engineering at Vanderbilt University and partly by funds from Sandia National Laboratories through summer internship for the first author. The first author would also like to acknowledge helpful discussions with Youssef Marzouk regarding the theory of Bayesian inference for inverse problems, and valuable assistance from Tarn Duong for contour plotting of bivariate density estimates.

References

[1] Campbell, K., "Statistical Calibration of Computer Simulations," *Reliability Engineering and System Safety*, Vol. 91, Nos. 10–11, 2006, pp. 1358–1363.

[2] Trucano, T., Swiler, L., Igusa, T., Oberkampf, W., and Pilch, M., "Calibration, Validation, and Sensitivity Analysis: What's What," *Reliability Engineering and System Safety*, Vol. 91, Nos. 10–11, 2006, pp. 1331–1357.

[3] Vecchia, A., and Cooley, R., "Simultaneous Confidence and Prediction Intervals for Nonlinear Regression Models, with Application to a Groundwater Flow Model," *Water Resources Research*, Vol. 23, No. 7, 1987, pp. 1237–1250. doi:10.1029/WR023i007p01237

[4] Beven, K., and Binley, A., "The Future of Distributed Models: Model Calibration and Uncertainty Prediction," *Hydrological Processes*, Vol. 6, 1992, pp. 279–298. doi:10.1002/hyp.3360060305

[5] Stigter, J., and Beck, M., "A New Approach to the Identification of Model Structure," *Environmetrics*, Vol. 5, No. 3, 1994, pp. 315–333. doi:10.1002/env.3170050310

[6] Banks, H., "Remarks on Uncertainty Assessment and Management in Modeling and Computation," *Mathematical and Computer Modelling*, Vol. 33, Nos. 1–3, 2001, pp. 39–47. doi:10.1016/S0895-7177(00)00227-2

[7] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435. doi:10.1214/ss/1177012413

[8] Santner, T. J., Williams, B. J., and Noltz, W. I., *The Design and Analysis of Computer Experiments*, Springer–Verlag, New York, 2003.

[9] Martin, J., and Simpson, T., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, 2005, pp. 853–863.

[10] Kennedy, M. C., and O'Hagan, A., "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society Series B (Methodological)*, Vol. 63, No. 3, 2001, pp. 425–464. doi:10.1111/1467-9868.00294

[11] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D., "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, Vol. 86, No. 416, 1991, pp. 953–963. doi:10.2307/2290511

[12] Morris, M. D., Mitchell, T. J., and Ylvisaker, D., "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, Vol. 35, No. 3, 1993, pp. 243–255. doi:10.2307/1269517

[13] Kennedy, M. C., and O'Hagan, A., "Predicting the Output from a Complex Computer Code When Fast Approximations are Available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13. doi:10.1093/biomet/87.1.1

[14] Bichon, B. J., Eldred, M. S., Swiler, L. P., Mahadevan, S., and McFarland, J. M., "Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions," *AIAA Journal* (submitted for publication).

[15] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. doi:10.1023/A:1008306431147

[16] 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., "A Framework for Validation of Computer Models," National Inst. of Statistical Sciences, TR 128, Research Triangle Park, NC, 2002.

[17] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241.

[18] Kaymaz, I., "Application of Kriging Method to Structural Reliability Problems," *Structural Safety*, Vol. 27, No. 2, 2005, pp. 133–151. doi:10.1016/j.strusafe.2004.09.001

[19] Ripley, B., *Spatial Statistics*, Wiley, New York, 1981.

[20] Stein, M. L., *Interpolation of Spatial Data: Some Theory for Kriging*, Springer Series in Statistics, Springer–Verlag, New York, 1999.

[21] Simpson, T., Peplinski, J., Koch, P., and Allen, J., "Metamodels in Computer-Based Engineering Deisgn: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150. doi:10.1007/PL00007198

[22] Rasmussen, C., *Evaluation of Gaussian Processes and Other Methods for Nonlinear Regression*, Ph.D. Thesis, Univ. of Toronto, Toronto, Ontario, Canada, 1996.

[23] Kennedy, M. C., and O'Hagan, A., "Supplementary Details on Bayesian Calibration of Computer Codes," Dept. of Probability and Statistics, Univ. of Sheffield, Sheffield, England, U.K., 2000.

[24] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[25] Marzouk, Y., Najm, H., and Rahn, L., "Stochastic Spectral Methods for Efficient Bayesian Solution of Inverse Problems," *Journal of Computational Physics*, Vol. 224, No. 2, June 2007, pp. 560–568. doi:10.1016/j.jcp.2006.10.010

[26] Hastings, W. K., "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, Vol. 57, No. 1, 1970,

pp. 97–109.
doi:10.1093/biomet/57.1.97

[27] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equations of State Calculations by Fast-Computing Machines," *Journal of Chemical Physics*, Vol. 21, No. 6, 1953, pp. 1087–1092. doi:10.1063/1.1699114

[28] Chib, S., and Greenberg, E., "Understanding the Metropolis–Hastings Algorithm," *American Statistician*, Vol. 49, No. 4, 1995, pp. 327–335. doi:10.2307/2684568

[29] Seber, G. A. F., and Wild, C. J., *Nonlinear Regression*, Wiley, New York, 2003

[30] Giunta, A. A., McFarland, J. M., Swiler, L. P., and Eldred, M. S., "The Promise and Peril of Uncertainty Quantification Using Response Surface Approximations," *Structure and Infrastructure Engineering*, Vol. 2, Nos. 3–4, 2006, pp. 175–189. doi:10.1080/15732470600590507

[31] Erickson, K. L., Trujillo, S. M., Thompson, K. R., Sun, A. C., Hobbs, M. L., and Dowding, K. J., "Liquefaction and Flow Behavior of a Thermally Decomposing Removable Epoxy Foam," *Computational Methods in Materials Characterisation*, edited by A. A. Mammoli and C. A. Brebbia, WIT Press, Boston, 2004, pp. 217–242.

[32] Romero, V. J., Shelton, J. W., and Sherman, M. P., "Modeling Boundary Conditions and Thermocouple Response in a Thermal Experiment," 2006 International Mechanical Engineering Congress and Exposition, American Society for Mechanical Engineers, Paper IMECE2006-15046, Nov. 2006.

[33] CALORE, Software Package, Ver. 4.1, Sandia National Labs., Albuquerque, NM, 2005.

[34] Bernardo, M. C., Buck, R. J., Liu, L., Nazaret, W. A., Sacks, J., and Welch, W. J., "Integrated Circuit Design Optimization Using a Sequential Strategy," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 11, No. 3, 1992, pp. 361–372. doi:10.1109/43.124423

[35] Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A., "Bayes Linear Strategies for Matching Hydrocarbon Reservoir History," *Bayesian Statistics 5*, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Oxford Univ. Press, Oxford, 1996, pp. 69–95.

[36] Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J., and Welch, W. J., "Circuit Optimization Via Sequential Computer Experiments: Design of an Output Buffer," *Journal of the Royal Statistical Society Series C (Applied Statistics)*, Vol. 47, No. 1, 1998, pp. 31–48. doi:10.1111/1467-9876.00096

[37] Eldred, M. S., Brown, S. L., Adams, B. M., Dunlavy, D. M., Gay, D. M., Swiler, L. P., Giunta, A. A., Hart, W. E., Watson, J. P., Eddy, J. P., Griffin, J. D., Hough, P. D., Kolda, T. G., Martinez-Canales, M. L., and Williams, P. J., "DAKOTA: A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 4.0 Reference Manual," Sandia National Labs., TR SAND2006-4055, Albuquerque, NM, Oct. 2006.

[38] Lee, P., *Bayesian Statistics: An Introduction*, Oxford Univ. Press, New York, 2004.

[39] Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, CRC Press, Boca Raton, FL, 1986.

[40] Jones, D., Perttunen, C., and Stuckman, B., "Lipschitzian Optimization Without the Lipschitz Constant," *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, 1993, pp. 157–181. doi:10.1007/BF00941892

[41] Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., *Markov Chain Monte Carlo in Practice*, CRC Press, Boca Raton, FL, 1996.

N. Alexandrov
Associate Editor