Reliability Engineering and System Safety 138 (2015) 194–209

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems

Shankar Sankararaman [a,*], Sankaran Mahadevan [b]

[a] SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, United States
[b] Vanderbilt University, Department of Civil and Environmental Engineering, Nashville, TN 37235, United States

ARTICLE INFO

Article history:
Received 2 June 2014
Received in revised form
21 January 2015
Accepted 24 January 2015
Available online 2 February 2015

Keywords:
Multi-level system
Uncertainty quantification
Bayesian network
Calibration
Validation
Verification

ABSTRACT

This paper proposes a Bayesian methodology to integrate model verification, validation, and calibration activities for the purpose of overall uncertainty quantification in different types of engineering systems. The methodology is first developed for single-level models, and then extended to systems that are studied using multi-level models that interact with each other. Two types of interactions amongst multi-level models are considered: (1) Type-I, where the output of a lower-level model (component and/or subsystem) becomes an input to a higher level system model, and (2) Type-II, where parameters of the system model are inferred using lower-level models and tests (that describe simplified components and/or isolated physics). The various models, their inputs, parameters, and outputs, experimental data, and various sources of model error are connected through a Bayesian network. The results of calibration, verification, and validation with respect to each individual model are integrated using the principles of conditional probability and total probability, and propagated through the Bayesian network in order to quantify the overall system-level prediction uncertainty. The proposed methodology is illustrated with numerical examples that deal with heat conduction and structural dynamics.

1. Motivation

1.1. Introduction

Computational models are widely used for the analysis, design, performance prediction and life cycle management of engineering systems. The process of model development needs to ensure that the models accurately represent the underlying scientific phenomenon. There are several activities in the development of a model [1], and these activities can be grouped into five steps, as shown in Fig. 1. Note that these steps are not necessarily in a fixed sequence; different sequences might be suitable for different problems and sometimes, iterations might be required between some of the steps. Also, note that some of the activities separately delineated by Alvin et al. [1] are collected together in order to facilitate the objectives of the present paper.

The first step is to develop a conceptual model and construct a mathematical equation (for e.g. a partial differential equation) that represents the model output (y) as a function of inputs ($\boldsymbol{x}$) and model parameters ($\boldsymbol{\theta}$) as $y = G(\boldsymbol{x};\boldsymbol{\theta})$. In the second step, a numerical solution procedure is developed to solve the mathematical equation, and this solution procedure is implemented using a computer code. The output of this computer code is the model prediction ($y_c = G_c(\boldsymbol{x};\boldsymbol{\theta})$); this y_c may be different from y, the true solution of the mathematical equation.

The third step is the process of model verification [2,3], which includes both code verification (identification of programming errors and debugging) and solution verification (convergence studies, identifying and computing solution approximation errors). Methods for code verification [4–9] and estimation of solution approximation error [7,9–16] have been investigated by several researchers. It is desirable to *perform verification before calibration and validation* so that the solution approximation errors are accounted for during calibration and validation. Solution approximation errors in finite element analysis have been estimated using a variety of techniques, such as convergence analysis [17], a posteriori error estimation [18], and Richardson extrapolation [16,19,20]. Another type of solution approximation error arises when the underlying model is replaced with a surrogate model for fast uncertainty propagation and/or model calibration. Many surrogate modeling techniques have been developed, such as regression models [21], polynomial chaos expansions [22], radial basis functions [23] or Gaussian processes [24]. The quantification of this surrogate model error is different for different types of

* Corresponding author. Tel.: +1 650 604 0552.
E-mail address: shankar.sankararaman@nasa.gov (S. Sankararaman).

http://dx.doi.org/10.1016/j.ress.2015.01.023
0951-8320/

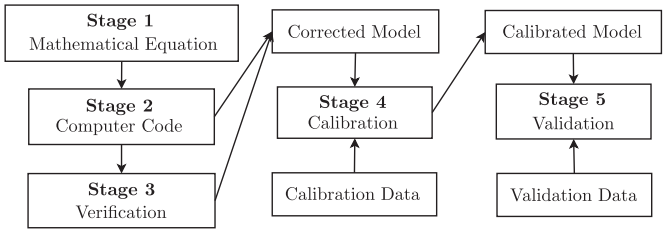

Fig. 1. Stages in model development.

surrogate models and the methods are well-established in the literature.

The fourth step is model parameter estimation or model calibration. The mathematical equation developed in the first step contains some parameters, denoted by $\boldsymbol{\theta}$ (for example, damping coefficient in a differential equation governing plate deflection under dynamic loading) and the values of these parameters for a particular system may need to be estimated based on observed input–output data. Least squares [25], likelihood-based [26,27], and Bayesian [28–34] methods are available for model parameter estimation. In classical statistics, the fundamental assumption is that the parameter is a deterministic unknown quantity and it is not meaningful to discuss the probability distribution of the parameter; therefore, the uncertainty about the value of the parameter is expressed in terms of confidence intervals. On the other hand, the Bayesian approach attributes a probability distribution (prior and posterior) to the model parameters, and this uncertainty is representative of the analyst's uncertainty about the model parameter.

Having calibrated the model, the fifth step is model validation which refers to the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended use of the model [4,35]. In this regard, researchers have been developing different types of validation metrics that express the accuracy of a computational model through comparison with experimental data, and determine whether the model is adequate for its intended use (sometimes, referred to as qualification [7]). Coleman and Stern [36] and Oberkampf and Trucano [7] discussed several philosophical and practical aspects of model validation, and provided guidelines for conducting validation experiments and developing validation metrics. Available approaches for quantitative model validation are based on statistical confidence intervals [37], computing distance between the model prediction and experimental data by computing the area metric [9,38], normalizing residuals [39], classical statistics-based hypothesis testing [40], Bayesian hypothesis testing [41–45], and reliability analysis-based techniques [46–48]. Liu et al. [49] and Ling and Mahadevan [50] investigated several of these validation approaches in detail, and discussed their practical implications in engineering. While some of these approaches compute validation metrics, some other approaches focus on directly estimating the so-called model form error [14,30] as the difference between the model prediction and the underlying physical phenomenon the model seeks to represent. The present manuscript mostly focuses on computing validation metrics and does not explicitly compute the model-form error, *while performing validation*. (However, the model form error can be computed through the use of a discrepancy function in the Kennedy O'Hagan framework for model calibration, but such an analysis would have to be performed during the previous task of calibration and not during validation.)

Another important issue related to model validation is the topic of extrapolating the model to application conditions under which experiments may not have been performed. Typically, there are two types of extrapolation. The first type is where the model is validated at certain input values, but prediction needs to be performed at other input values that are not contained in the validation domain. The second type of extrapolation is where validation is performed using a simplified system (with restricted features, physics, etc.) and the desired prediction is of the original system. While regression-based techniques have been developed for the first type of extrapolation [9], model extrapolation, in general, is still a challenging issue and researchers are currently studying this problem. This paper does not focus on the first type of model extrapolation and primarily focuses on the integration of results from verification, validation, and calibration activities; in the process, some aspects of the second type of extrapolation are discussed later in this paper.

1.2. Need for integration

While individual methods for calibration, verification, and validation have been developed as mentioned above, it is not clear how these activities can be integrated for the purpose of overall uncertainty quantification in the model prediction. This is not trivial because of several reasons. First, the solution approximation errors calculated as a result of the verification process need to be accounted for during calibration, validation, and prediction. Second, the result of validation may lead to a binary result, i.e., the model is accepted or rejected; however, even when the model is accepted, it is not completely valid/correct. Hence, it is necessary to account for the degree of correctness of the model, during prediction and uncertainty quantification. Third, calibration and validation are performed using independent data sets and it is not straightforward to compute their combined effect on the overall uncertainty in the system-level response.

The issue gets further complicated when the behavior of complex engineering systems is studied using multiple component-level and subsystem-level models that integrate to form the overall multi-level system model. In each level, there is a computational model with inputs, parameters, and outputs, experimental data (hopefully available for calibration and validation separately), and several sources of uncertainty – physical variability, data uncertainty (sparse or imprecise data, measurement errors), and model uncertainty (parameter uncertainty, solution approximation errors and model form error). In such a multi-level system, the first task would be to connect all the available models and associated sources of uncertainty.

Recent studies by the authors and coworkers [51,52] have demonstrated that the Bayesian network methodology provides an efficient and powerful tool to integrate multiple levels of models, associated sources of uncertainty and error, and available data at multiple levels. While the Bayesian approach can be used to perform calibration and validation individually for each model in the multi-level system, it is not straightforward to integrate the information from these activities in order to compute the overall uncertainty in the system-level prediction. This paper extends the Bayesian approach to integrate and propagate information from verification, calibration, and validation activities in order to quantify the margins and uncertainties in the overall system-level prediction. In Bayesian calibration, the goal is to estimate the probability distributions of the underlying model parameters, using the data available for calibration. Once the model is calibrated, it is validated using an independent set of input–output data. There are several advantages in using a Bayesian methodology for both calibration and validation:

1. Both calibration and validation involve comparing model prediction against experimental data; the Bayesian approach not only allows the comparison of entire distributions of model prediction and experimental data, but also provides a

systematic approach to include the various types of uncertainty – physical variability, data uncertainty, and model uncertainty/errors – through the Bayesian network.
2. The Bayesian approach can systematically handle epistemic uncertainty due to sparse, imprecise, and unpaired input–output data, as demonstrated by the authors for calibration [53] as well as validation [45,50,54].
3. Model validation in this paper is through the use of the Bayes factor metric [55], which is the ratio of the likelihoods that the model is valid and that the model is invalid. The Bayes factor can be used to directly calculate the probability that the model is valid. Further, the threshold Bayes factor for model acceptance can be derived based on a risk versus cost trade-off, thereby aiding in robust, meaningful decision-making, as shown by Jiang and Mahadevan [56]. Alternatively, the model reliability metric [47] can also be used to compute "the probability that the model is valid".

While Bayesian methods can be used for calibration as well as validation, the two procedures are different and should not be confounded. The distinction will be clearly maintained in this paper; in fact, this paper considers separate data sets for calibration and validation. While the Bayesian approach offers several advantages, there are some practical challenges in implementing Bayesian methods; these challenges are still being addressed by several researchers. Some researchers view the need to assume prior probability distributions while other researchers find the method attractive to incorporate prior knowledge (if available) and use non-informative prior probability distributions [55] when no prior knowledge is available. More importantly, Bayesian techniques involve the computation of high-dimensional integrals that are often solved through Markov Chain Monte Carlo (MCMC) sampling techniques that require extensive computational effort. With the advent of high performance computing techniques, it has become easier to implement such computationally intensive methods for practical applications, and therefore, Bayesian methods are being increasingly applied in engineering disciplines, during the past 20 years.

The above methods for calibration and validation have been demonstrated only for individual models with calibration and validation data. What happens when there is flow of information across multiple levels of models that are used to study a multi-level system? Since the Bayesian approach represents the various sources of uncertainty across multiple levels through probability distributions, the problem reduces to propagating these probability distributions through component and subsystems, in order to compute the uncertainty in the system-level prediction. Both solution approximation errors and model form errors can be included in the Bayesian network as additional nodes [52,57,58]. The resultant Bayesian network can be used for both the forward problem of uncertainty propagation [57] and inverse problem of calibration [58,59]. The results of calibration and validation activities are expressed in terms of probability distributions for the model parameters, and the probability that each model is valid respectively. The Bayesian approach is thus able to provide a unified framework for integrating information from verification, calibration, and validation at multiple levels to calculate the overall system-level prediction uncertainty. Using the principles of conditional probability and total probability, this paper develops a computational approach for such integration. The proposed methodology is first developed for single-level models and then extended to multi-level system models.

1.3. Multi-level system models

Typically, a multi-level system is studied using different types of models; each model may represent a particular component, a particular subsystem, or an isolated set of features/physics of the original system. The interaction between any two models depends on what features of the original system they represent, and it is necessary to translate the dependency between the features into mathematical relationship between the models. In order to facilitate such translation, and hence, the objectives of this paper, two types of interactions (designated as Type-I and Type-II in this paper), are considered in detail. In both of these types, the quantity of interest is an overall system-level response, but there is a significant difference in how this quantity is calculated using information from lower-level models, verification, validation, and calibration.

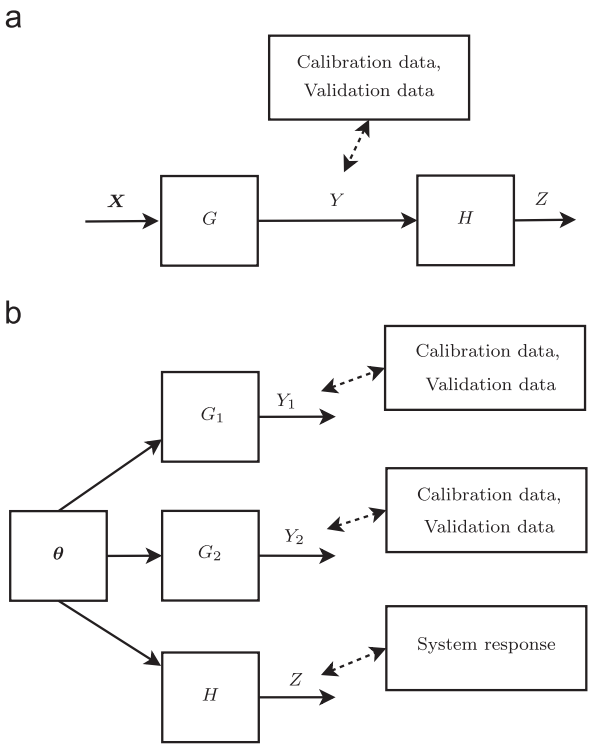

Fig. 2. Two types of interactions between models: (a) Type-I interaction and (b) Type-II interaction.

The interaction between two models (denoted by G and H in Fig. 2(a)) is considered to be "Type-I", when the lower-level model G represents the behavior of a particular component/subsystem whose output (Y) becomes an input to a higher-level subsystem/component whose behavior is represented by the higher-level model H. Each model has its own set of model parameters (not indicated in Fig. 2(a)); there may or may not be any model parameter common between two models. For example, the rise in the temperature (Y, computed using G) of a conducting wire leads to a change in its resistance, and hence, in its current carrying capacity (Z, which is computed using H).

The interaction between two models is said to be "Type-II", when the lower-level model represents a simplification of the overall system. Typically, it may not be possible to study certain features of the system (i.e., the system parameters) since extensive testing of the overall system may be prohibitory. Therefore, a simplified configuration that consists of an isolated set of features and/or an isolated set of physics of the multi-level system is often considered for testing. The response of the simplified subsystem is not directly related to the response of the overall system; however, there are some parameters ($\boldsymbol{\theta}$) of the system model (H) that can be inferred using lower-level models and experiments. Further, multiple test options may be available (two in Fig. 2(b), where the model G_2 may describe more features than the model G_1, in terms of complexity). The model of the highest complexity (H)

represents the system of interest and the system-level response (Z) needs to be calculated. The model parameters ($\boldsymbol{\theta}$) are calibrated using models and experiments of reduced complexity (e.g. isolated components or physics), and then propagated through the system model to compute the desired response.

For example, consider a coupon subjected to axial testing (say, the axial deflection is modeled using G_1), a cantilever beam subjected to point loading (say, the deflection is modeled using G_2) and a plate subjected to bending (say, the bending stress is modeled using H), where the coupon, the beam, and the plate are made of the same material. While both the axial deflection of the coupon (Y_1) and the deflection of the beam (Y_2) are not directly related to the bending stress in the plate (Z), some material properties (like the elastic modulus, denoted by $\boldsymbol{\theta}$) of the plate can be studied using tests performed on the coupon and the beam. Note that there is no interaction between models G_1 and G_2; there is "Type-II" interaction between models G_1 and H, and between models G_2 and H. Urbina et al. [51] and Sankararaman et al. [34] discuss practical multi-level systems with Type-II interaction; while the former considers multiple lower-level models of increasing complexity, the latter considers lower-level models of increasing complexity and physics, i.e., only a few aspects of physics are captured at the lowest-level model and more aspects are increasingly captured in subsequent higher levels. As mentioned earlier, in some cases, an extrapolation problem can be described using two models with "Type-II" interaction. When the application conditions are physically different from validation conditions (for example, validating using a beam but extrapolating to a plate), model G_2 may correspond to the validation conditions and model H may correspond to the extrapolation conditions. (The problem of extrapolating the model to untested input conditions cannot be described using this approach.)

Note that there is a third type of interaction between two models, that is commonly observed in multi-disciplinary systems. The two models represent different physics, but the output of each model is an input to the other. This is feedback coupling, and it is necessary to perform iterative analysis between these two models in order to compute the system-level response. The authors developed a likelihood-based method [60] to mathematically transform two-way coupling to one-way coupling; as a result, the method proposed in this paper for Type-I interaction can also be applied to models with feedback coupling. Further, there may be other types of interactions in multi-level models, but this paper studies only these two types of interactions (Type-I and Type-II) in detail. The primary goal of this paper is to develop a framework for the integration of verification, validation, and calibration activities in order to facilitate system-level uncertainty quantification, by considering multi-level models that exhibit Type-I or Type-II interaction. This is accomplished by computing the probability density function (PDF) or the cumulative distribution function (CDF) of the system-level response quantity of interest, and this PDF or CDF needs to incorporate the results of verification, validation, and calibration.

The proposed integration methodology is different for system models with Type-I and Type-II interactions. In the former case, the linking variables between two models are the outputs of the lower-level models that become inputs to the higher level models, whereas in the latter case, the linking variables are the common model parameters. With the focus on the linking variables, and using the principles of conditional probability and total probability, this paper develops a Bayesian network-based methodology to integrate the results of verification, validation, and calibration activities, and to compute the uncertainty in the overall system-level prediction.

1.4. Organization of the paper

The rest of the paper is organized as follows. Section 2 discusses the proposed methodology for the integration of calibration, verification, and validation in a single-level model. Sections 3 and 4 extend the proposed methodology for models with Type-I and Type-II interactions respectively. Though the methodology is individually developed for each type of interaction, it is straightforward to extend it to the generic multi-level models that may contain both types of interactions. Finally, the proposed methods are illustrated using two numerical examples, by first considering a single-level model (Section 5), and then by considering multi-level system models with Type-I interaction (Section 6) and Type-II interaction (Section 7). The first numerical example studies the behavior of an electric wire under heat conduction, and the second numerical example deals with a structural dynamics challenge problem developed at Sandia National Laboratories [61]. Finally, concluding remarks are presented in Section 8.

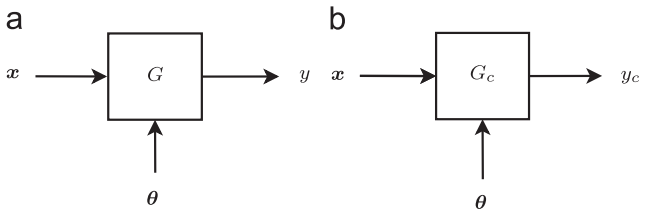

Fig. 3. A single-level model: (a) mathematical equation and (b) computer code.

2. Integration of verification, validation, and calibration

Consider a single-level model as shown in Fig. 3. The inputs are $\boldsymbol{x}$, the model parameters are $\boldsymbol{\theta}$, the true solution of the mathematical equation is y, and the code output is y_c. Both y_c and y are deterministic functions of inputs ($\boldsymbol{x}$) and model parameters ($\boldsymbol{\theta}$).

This section proposes methods to integrate the results of calibration, verification, and validation of the model. Since the process of verification is not related to data, it needs to be performed first; both calibration and validation must include the results of verification analysis (i.e., solution error quantification). Then, the results of verification, calibration, and validation are integrated to compute the overall uncertainty in the response quantity.

2.1. Verification

The process of verification checks how close the code output is to the true solution of the mathematical equation. As stated earlier in Section 1, it is not only sufficient to verify that the two solutions are sufficiently close, but also essential to quantify the solution approximation error, i.e., the difference between the code output and true solution, in order to quantify the uncertainty in the prediction. Once the solution approximation error is computed, the true solution of mathematical equation can be computed as a function of the model inputs and parameters as $y(\boldsymbol{x};\boldsymbol{\theta}) = y_c(\boldsymbol{x};\boldsymbol{\theta}) + G_{se}(\boldsymbol{x};\boldsymbol{\theta})$, where $G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ denotes the solution approximation error. Two types of solution approximation errors – discretization error and surrogate modeling error – are considered in this paper. However, the integration methodology is general enough to accommodate other types of solution approximation errors, once such errors can be quantified.

In general, solution approximation errors ($G_{se}(\boldsymbol{x};\boldsymbol{\theta})$) are deterministic quantities; however, sometimes, when probabilistic approaches are used to quantify them, it becomes necessary to represent solution approximation errors using probability distributions. For example, the discretization error in finite element analysis is a deterministic quantity. When Richardson extrapolation [20] is used to quantify this error, $G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ can be calculated deterministically. When a probabilistic method, such as Gaussian process modeling [16], is used, both the mean and variance of $G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ are predicted; this variance is simply dependent on the training points used to train the Gaussian process model and on

how far it is necessary to extrapolate, and it does not imply that $G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ is a physically random quantity. Similarly, when a high fidelity simulation (such as a computationally intensive finite element analysis) is replaced by an inexpensive surrogate model, the surrogate model error (i.e., the difference between the prediction of the original high fidelity simulation and the surrogate model) is a deterministic quantity. However, the solution of the underlying high fidelity simulation is only available at a few input settings. The surrogate model prediction and the error (at untrained input locations) are therefore expressed using a probability distribution; this probability distribution only indicates the analyst's uncertainty regarding the surrogate model error and does not imply that the surrogate model error is random. This uncertainty is epistemic, and reduces as the number of training points increases. The Bayesian framework represents such epistemic uncertainty through probability distributions.

In this paper, Richardson extrapolation is used to compute discretization error, and Gaussian process surrogate models are used to replace high fidelity simulations. While the former leads to deterministic error estimates ($G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ is point-valued), the latter leads to stochastic error representation ($G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ is a probability distribution). In practical engineering problems, both discretization and surrogate modeling may be necessary, and therefore, the overall solution approximation error is composed of both deterministic and stochastic terms. In the context of uncertainty propagation, deterministic errors are addressed by correcting the bias, whenever they occur, and the corrected solutions are used to train the surrogate model; the stochastic errors of the surrogate model are accounted for through sampling based on their estimated distributions. As a result, the overall solution approximation error $G_{se}(\boldsymbol{x};\boldsymbol{\theta})$ is also stochastic, i.e., y is stochastic even for given values of $\boldsymbol{x}$ and $\boldsymbol{\theta}$, and this stochasticity, without physical randomness, also needs to be interpreted subjectively. The remainder of this subsection briefly reviews the estimation of discretization error and surrogate model uncertainty.

Recall from Fig. 1 that verification is performed before calibration and validation. Once the solution approximation errors are quantified, the model predictions are corrected as explained above, the corrected solution (y) and not the code output (y_c) is used for calibration and validation. Such an approach integrates the result of verification into calibration, validation, and all subsequent uncertainty quantification.

2.1.1. Discretization error

Several methods are available in the literature [18,62,63] to estimate discretization error in finite element analysis, but many of them only quantify a surrogate measure of error to facilitate adaptive mesh refinement. The Richardson extrapolation (RE) method has been found to come closest to quantifying the actual discretization error [14,20]. This technique has been commonly applied to quantifying discretization error in finite element analysis by several researchers [6,7,10,52].

Consider a polynomial model $y = y_c + Ah^p$, where y_c is the solution corresponding to mesh size h, and y corresponds to the "true" solution of the mathematical equation which is obtained as h tends to zero. Three different mesh sizes ($h_1 < h_2 < h_3$) are considered and the corresponding finite element solutions ($y_c(h_1) = \Psi_1,\ y_c(h_2) = \Psi_2,\ y_c(h_3) = \Psi_3$) are calculated. Using the aforementioned polynomial model, y can be estimated by solving three simultaneous equations in three variables. Closed form solutions are available in some special cases; for example, if $r = h_3/h_2 = h_2/h_1$, then the discretization error (ϵ_h) and the true solution can be calculated as:

$$y = \Psi_1 - \epsilon_h$$
$$\Psi_2 - \Psi_1 = \epsilon_h(r^p - 1)$$

$$p\log(r) = \log\left(\frac{\Psi_3 - \Psi_2}{\Psi_2 - \Psi_1}\right) \tag{1}$$

The solutions Ψ_1, Ψ_2, Ψ_3 are dependent on both $\boldsymbol{x}$ and $\boldsymbol{\theta}$ and hence the error estimate ϵ_h and the true solution y are also functions of both $\boldsymbol{x}$ and $\boldsymbol{\theta}$. Since the discretization error is a deterministic quantity, it needs to be corrected for, in the context of uncertainty propagation.

Recently, Rangavajhala et al. [16] extended the Richardson extrapolation methodology from a polynomial relation to a more flexible Gaussian process extrapolation. This approach expresses the discretization error as a probability distribution, and therefore, the training points (in particular, the output values) for the surrogate model are themselves stochastic. Rasmussen [64–66] discusses constructing GP models when the training point values are themselves stochastic.

2.1.2. Surrogate model uncertainty

This section considers the case where the original computer code is replaced with a Gaussian process surrogate model. The basic idea of the GP model is that the response values Y evaluated at different values of the input variables, $\boldsymbol{X}$, are modeled as a Gaussian random field, with a mean and covariance function. Suppose that there are m training points, x_1, x_2, x_3 $\ldots x_m$ of a d-dimensional input variable vector, yielding the output values $Y(x_1)$, $Y(x_2)$, $Y(x_3)$ $\ldots Y(x_m)$. The training points can be compactly written as x_T versus y_T where the former is a $m \times d$ matrix and the latter is a $m \times 1$ vector. Suppose that it is desired to predict the response (output values y_P) corresponding to the input x_P, where x_P is $p \times d$ matrix; in other words, it is desired to predict the output at p input combinations simultaneously. Then, the joint density of the output values y_P can be calculated as:

$$p(y_P|x_P, x_T, y_T; \Theta) \sim N(m, S) \tag{2}$$

where Θ refers to the hyperparameters of the Gaussian process, which need to be estimated based on the training data. The prediction mean and covariance matrix (m and S respectively) can be calculated as:

$$m = K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} y_T$$
$$S = K_{PP} - K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} K_{TP} \tag{3}$$

In Eq. (3), K_{TT} is the covariance function matrix (size $m \times m$) amongst the input training points (x_T), and K_{PT} is the covariance function matrix (size $p \times m$) between the input prediction point (x_P) and the input training points (x_T). These covariance matrices are constructed using the chosen covariance function (squared exponential is chosen, in this paper), which are functions of the training points terms and the hyperparameters (Θ): a multiplicative term (θ), the length scale in all dimensions (l_q, $q=1$ to d), and the noise standard deviation (σ_n). These hyperparameters are estimated based on the training data by maximizing the following log-likelihood function:

$$\log p(y_T|x_T; \Theta) = -\frac{y_T^T}{2}(K_{TT} + \sigma_n^2 I)^{-1} y_T - \frac{1}{2}\log|(K_{TT} + \sigma_n^2 I)| + \frac{d}{2}\log(2\pi) \tag{4}$$

Once the hyperparameters are estimated, then the Gaussian process model can be used for predictions using Eq. (3). For details of this method, refer to [64,66–72].

Note that the variance in Eq. (3) is representative of the uncertainty due to the use of the GP surrogate model, and this uncertainty needs to be accounted for, while computing the overall prediction uncertainty.

2.2. Calibration

The next step is to estimate the model parameters ($\boldsymbol{\theta}$) using input–output ($\boldsymbol{x}$ versus y) data collected for calibration (D^C), using Bayes' theorem as:

$$f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C) = \frac{L(\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \tag{5}$$

In Eq. (5), $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ is the prior PDF and $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)$ is the posterior PDF; the calibration procedure uses the model form G and hence the posterior is conditioned on G. The function $L(\boldsymbol{\theta})$ is the likelihood of $\boldsymbol{\theta}$ defined as being proportional to the probability of observing the data D^C (given as $\boldsymbol{x_i}$ versus y_i; $i=1$ to n) conditioned on the parameters $\boldsymbol{\theta}$. This likelihood function is evaluated as:

$$L(\boldsymbol{\theta}) \propto \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{(2\pi)}} \exp - \left(\frac{(y_i - G(\boldsymbol{x_i}, \boldsymbol{\theta}))^2}{2\sigma^2} \right), \tag{6}$$

where σ is the standard deviation of $\epsilon_{obs} = y - G(\boldsymbol{x}, \boldsymbol{\theta})$. Note that the likelihood is constructed using the actual solution of the mathematical equation (y), i.e., after correcting the raw code prediction (y_c) with solution approximation errors (ϵ_{soln}). Sometimes, σ is also inferred along with $\boldsymbol{\theta}$, by constructing the joint likelihood $L(\boldsymbol{\theta}, \sigma)$. Note that Eq. (6) assumes that the experimental observations are unbiased. If the predictions are biased due to modeling errors, then the output can be modeled as $y = G(\boldsymbol{x}, \boldsymbol{\theta}) + \delta + \epsilon_{obs}$, where δ represents the modeling error, and is inferred along with $\boldsymbol{\theta}$. This approach was further extended by Kennedy and O'Hagan [30] by modeling the output as $y = G(\boldsymbol{x}, \boldsymbol{\theta}) + \delta(\boldsymbol{x}) + \epsilon_{obs}$, where $\delta(\boldsymbol{x})$ (sometimes, referred to as the model inadequacy function) is represented using a Gaussian process (GP) whose "hyperparameters" are also inferred along with $\boldsymbol{\theta}$. A challenge with the Kennedy O'Hagan (KOH) framework is that it is necessary to (1) simultaneously calibrate both the model parameters and the GP hyperparameters; and (2) possess good prior knowledge regarding both the model parameters and the GP hyperparameters. This is a challenging issue and still being addressed by several researchers [73,74]. Using the KOH framework, all the available data have been used [30] to simultaneously estimate both the model parameters and the model form error (expressed through the model discrepancy function) in the calibration step. An alternative approach is to consider separate sets of data for calibration and validation; first, the model parameters are estimated without considering the model discrepancy term in the calibration step, and then, the effect of model form error is inferred in the validation step by calculating a validation metric. The latter route is pursued in this paper, in an effort to distinctly separate the two steps of calibration and validation, and use different sets of data for these two activities.

Note that the model "G" is used for calibration in Eq. (5); recall that "G" refers to the model corrected for solution approximation errors. Hence, the results of verification are included in the calibration procedure. Deterministic discretization errors are corrected for before training the surrogate model, and the surrogate model uncertainty is included in the likelihood function as demonstrated by Kennedy and O'Hagan [30] and McFarland [69]. In addition, the construction of the likelihood function can also include additional uncertainty in inputs and parameters, and account for imprecise and unpaired data. Refer to [26,52,58,75] for further details on the construction of the likelihood function in different types of situations.

The posterior PDFs of the model parameters can be calculated using direct integration of the denominator in Eq. (5), if the number of calibration parameters is small. Alternatively, Markov Chain Monte Carlo sampling [76] methods such as Metropolis algorithm [77], Gibbs algorithm [78] or slice sampling [79] can be used to generate samples of the posterior distributions of the parameters. The method of slice sampling is used in this paper.

2.3. Validation

Assume that an independent set of validation data (D^V) is available. The prediction of the verified and calibrated model is compared against the validation data. The model prediction can be computed as a function of input as:

$$f_Y(y|\boldsymbol{x}, G, D^C) = \int f_Y(y|\boldsymbol{x}, \boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)\, d\boldsymbol{\theta} \tag{7}$$

In the case of partially characterized validation data (e.g. field data), the input $\boldsymbol{x}$ may not be measured, in which case the model prediction must include the uncertainty in the input as:

$$f_Y(y|G, D^C) = \int f_Y(y|\boldsymbol{x}, \boldsymbol{\theta}) f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)\, d\boldsymbol{\theta}\, d\boldsymbol{x} \tag{8}$$

The above equations simply refer to uncertainty propagation through the model and hence the model prediction is conditioned on the event that the mathematical model is correct, and written as $f_Y(y|G, D^C)$. The results of verification are included while computing y, and the results of calibration are included by using the posterior PDF of the model parameter ($f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)$) in the prediction.

The model prediction is then compared with the validation data using Bayesian hypothesis testing in this section. Let $P(G)$ and $P(G')$ denote the probabilities that the model is valid (null hypothesis) and that the model is invalid (alternate hypothesis) respectively. Prior to validation, if no information is available, $P(G) = P(G') = 0.5$. Using Bayesian hypothesis testing, these probabilities can be updated using the validation data (D^V), and the likelihood ratio, referred to as Bayes factor, is defined as:

$$B = \frac{P(D^V|G)}{P(D^V|G')} \tag{9}$$

The likelihoods $P(D^V|G)$ and $P(D^V|G')$ are denoted as $L(G)$ and $L(G')$ respectively. The numerator $P(D^V|G)$ can be calculated using $f_Y(y|G)$ as:

$$L(G) \propto P(D^V|G) \propto \int f(D^V|y) f_Y(y|\, G, D^C)\, dy \tag{10}$$

In Eq. (10), the term $f(D^V|\, y)$ is calculated based on the measurement error $\epsilon_{obs} \sim N(0, \sigma^2)$, as:

$$f(D^V|y) = \frac{1}{\sigma\sqrt{(2\pi)}} \exp - \left(\frac{(D^V - y)^2}{2\sigma^2} \right) \tag{11}$$

In order to compute $P(D^V|G')$, it is necessary to assume the alternate PDF $f_Y(y|G')$, i.e., the PDF of Y when the model is wrong. Expert opinion may be used to construct this PDF, or a uniform PDF may be used if no additional information is available. Then $L(G')$ is calculated similar to Eq. (10) by replacing $f_Y(y|G, D^C)$ with $f_Y(y|G')$. Using Bayes' theorem and assuming that G and G' are equally likely before collecting data, the probability that the model is correct, i.e., $P(G|D^V)$ can be calculated as $B/(B+1)$ [14].

The Bayes factor gives a probabilistic measure of model validity. While the Bayesian hypothesis testing is one approach to calculate the probability that the model is correct, the model reliability metric [46,47] provides an alternative methodology. Similar to the Bayesian hypothesis testing method, the reliability-based method can also account for the different types of uncertainty, and compute the probability that the model is valid, i.e., $P(G|D^V)$. The assumption of the alternate PDF $f_Y(y|G')$, though not necessary for the computation of $P(G|D^V)$, is still necessary for the purpose of integration of verification, validation, and calibration as explained in the following subsection.

2.4. Integration for overall uncertainty quantification

The calibration procedure in Section 2.2 assumed that the model form G is valid and estimated the model parameters $\boldsymbol{\theta}$. In contrast, the validation procedure in Section 2.3 calculated the probability that the model G is valid by assuming the uncertainty in the model parameters $\boldsymbol{\theta}$. The two results can be combined to calculate the overall uncertainty in the model prediction, using the theorem of total probability as:

$$f_Y(y|D^C, D^V) = P(G|D^V) f_Y(y|G, D^C) + P(G') f_Y(y|G') \tag{12}$$

In Eq. (12), $P(G|D^V)$ can also be calculated using the model reliability metric instead of Bayesian hypothesis testing. Note that the result of verification, i.e., solution approximation error, was already included in both calibration and validation. Thus, the PDF $f_Y(y|D^C, D^V)$ includes the results of verification, calibration, and validation activities.

3. Multi-level models with Type-I interaction

Consider a system which is studied using multiple levels of models with Type-I interaction, i.e., the output of a lower-level model becomes an input to the higher level model, and hence is the linking variable between the two models. While the methods of verification, validation, and calibration can be applied to each of the individual models, the challenge is to integrate the results from these activities performed at multiple levels. This section proposes a methodology for the integration of verification, validation and calibration across multiple levels of modeling with Type-I interaction. The proposed methodology is illustrated for two levels of models, as shown in Fig. 4, and Eq. (13), but can be extended to any number of levels of modeling without loss of generality:

$$\begin{aligned} Y &= G(\boldsymbol{X}; \boldsymbol{\theta}) \\ Z &= H(Y, \boldsymbol{W}; \boldsymbol{\alpha}) \end{aligned} \tag{13}$$

Assume that data are not available at the system-level, i.e., it is not possible to validate/calibrate model H. Let D^C and D^V denote the data available on Y for calibration (of $\boldsymbol{\theta}$) and validation (of G) respectively. Let $\epsilon_{obs} \sim N(0, \sigma^2_{obs})$ denote the measurement errors in the data.

The first step is to connect the various sources of uncertainty using a Bayesian network, as shown in Fig. 5.

This Bayesian network indicates that two sets of data are available for calibration and validation; the Bayesian methods for calibration and validation can be applied to these sets. (If the KOH framework is pursued for calibration, then both the parameters and the model inadequacy function can be included in the Bayesian network.) While Bayesian updating is used for model calibration, either Bayesian hypothesis testing or the model reliability metric can be pursued for model validation.

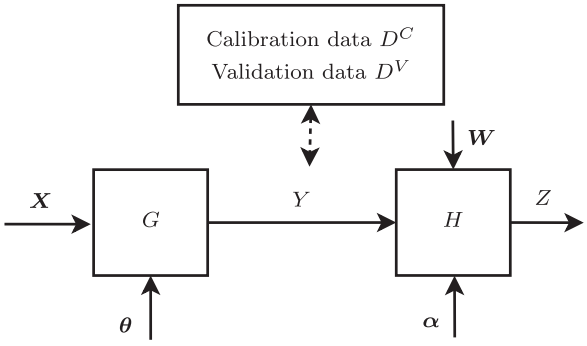

Fig. 4. Type-I interaction: two levels of models.

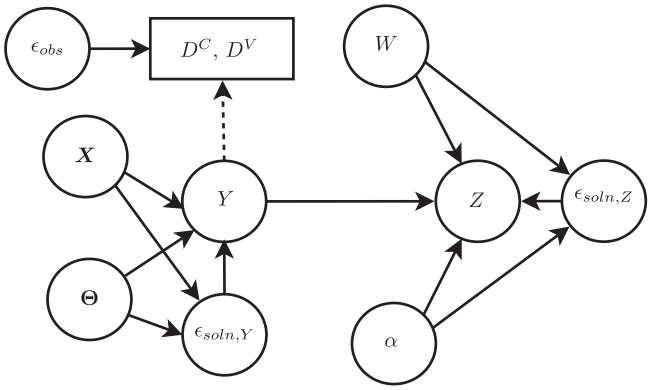

Fig. 5. Bayesian network: Type-I interaction between two models.

The task is to compute the overall uncertainty in Z by using lower-level data; this uncertainty must include the effect of verification, calibration, and validation activities.

3.1. Verification, calibration, and validation

Both the models G and H can be verified since experimental data are not required for verification. During the process of verification, the solution approximation error (ϵ_{soln}) is quantified for both the models G and H. Note that the solution approximation error is a function of the inputs and the model parameters. Note that these solution approximation errors (ϵ_{soln} for both G and H) account for the combined effect of both deterministic and stochastic errors, as discussed in Section 2.1. Now the Bayesian network includes quantification of solution approximation error and it can now be used for calibration, validation and system-level prediction.

The next step is to calibrate the model parameters. Suppose that the PDFs of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are assumed to be $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ and $f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ respectively before any testing; these are the prior PDFs. Since no data are available on Z, it is not possible to update the PDF of $\boldsymbol{\alpha}$. The data on Y, i.e., D^C, is used to calibrate the parameters $\boldsymbol{\theta}$, using Bayesian inference, as in Section 2.2. The calibration procedure uses the data and assumes that the model is correct, and hence the posterior PDF of $\boldsymbol{\theta}$ is denoted by $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)$. During the calibration procedure, for every realization of $\boldsymbol{\theta}$, the corresponding solution approximation error is estimated and therefore, calibration is based on comparing y against experimental data, rather than y_c, thereby accounting for the results of verification during calibration.

Additional independent data (D^V) is assumed to be available for the purpose of validating the model G. The alternate hypothesis PDF $f_Y(y|G')$ is assumed and the posterior probability of model being correct, i.e., $P(G|D^V)$ is calculated as explained in Section 2.3; alternatively, the model reliability metric $P(M)$ can also be used instead of $P(G|D^V)$.

3.2. Integration for overall uncertainty quantification

The Bayesian network can be used for forward propagation of uncertainty using the principles of conditional probability and total probability. Prior to the collection of any data, the uncertainty in $\boldsymbol{x}$, $\boldsymbol{\theta}$, and $\boldsymbol{\alpha}$ can be propagated through the models as:

$$\begin{aligned} f_Z(z|H) &= \int f_Z(z|\boldsymbol{w}, \boldsymbol{\alpha}, y, H) f_{\boldsymbol{W}}(\boldsymbol{w}) f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) f_Y(y|G)\, d\boldsymbol{w}\, d\boldsymbol{\alpha}\, dy \\ f_Y(y|G) &= \int f_Y(y|\boldsymbol{x}, \boldsymbol{\theta}, G) f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, d\boldsymbol{x}\, d\boldsymbol{\theta} \end{aligned} \tag{14}$$

However, this procedure assumes that (1) the PDFs of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ and $f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ respectively; and (2) the models G and H are correct. These two issues were addressed in calibration and

validation respectively. While the PDF of $\boldsymbol{\alpha}$ did not change, the PDF of $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ was updated to $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G, D^C)$. Further, the probability that G is correct, i.e., $P(G|D^V)$, was evaluated. These two quantities can now be used to calculate the overall uncertainty in Z. First, if the calibration data alone was used, then the PDFs of Y and Z are given by:

$$f_Z(z|G,H,D^C) = \int f_Z(z|\boldsymbol{w},\boldsymbol{\alpha},y,H) f_{\boldsymbol{W}}(\boldsymbol{w}) f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) f_Y(y|G,D^C)\, d\boldsymbol{w}\, d\boldsymbol{\alpha}\, dy$$
$$f_Y(y|G,D^C) = \int f_Y(y|\boldsymbol{x},\boldsymbol{\theta},G) f_{\boldsymbol{X}}(\boldsymbol{x}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G,D^C)\, d\boldsymbol{x}\, d\boldsymbol{\theta} \quad (15)$$

The theorem of total probability can then be used to include the result of validation. The PDF of Y is modified as;

$$f_Y(y|D^C,D^V) = P(G|D^V) f_Y(y|G,D^C) + P(G'|D^V) f_Y(y|G') \quad (16)$$

The overall uncertainty in Z, which includes the results of verification, calibration, and validation, can be calculated as:

$$f_Z(z|H,D^C,D^V) = P(G|D^V) f_Z(z|G,H,D^C) + P(G'|D^V) f_Z(z|G',H)$$
$$f_Z(z|G',H) = \int f_Z(z|\boldsymbol{w},\boldsymbol{\alpha},y,H) f_{\boldsymbol{W}}(\boldsymbol{w}) f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) f_Y(y|G')\, d\boldsymbol{w}\, d\boldsymbol{\alpha}\, dy \quad (17)$$

The PDF of Z is still conditioned on H because it is assumed that the model H is correct and it is not possible to calibrate/validate this model. In fact, Eq. (17) is equivalent to simply propagating the PDF $f_Y(y|D^C,D^V)$ (in Eq. (16)) through the model H. Note that the model H has been verified; therefore, during uncertainty propagation, it is necessary to estimate and account for the solution approximation error, thereby including the result of verification of H during calibration, validation, and response computation. Thus, the PDF of the linking variable can be used to compute the uncertainty in the system-level response, thereby integrating the results of verification, validation, and calibration activities at a lower-level.

The principles of conditional probability and total probability can also be extended to multiple models that exhibit Type-I interaction, as explained below.

3.3. Extension to multiple models

Until now, only the first model G was considered for verification, validation, and calibration. However, the proposed methodology is general and can be extended to multiple models. For example, consider the case where there are two models whose individual outputs become inputs for the system model. For example, consider the equations:

$$Y_1 = G_1(X_1,\theta_1)$$
$$Y_2 = G_2(X_2,\theta_2)$$
$$Z = H(Y_1,Y_2) \quad (18)$$

The inputs to the models G_1 and G_2 are X_1 and X_2 respectively; the corresponding parameters are θ_1 and θ_2 respectively. The Bayesian network for this multi-level system is shown in Fig. 6.

Assume that there is no data at the system-level Z, but data are available for calibration and validation of lower-level models G_1 and G_2, as shown in the Bayesian network in Fig. 6. Using the calibration data, the PDFs $f(\theta_1|G_1,D_1^C)$, $f(\theta_2|G_2,D_2^C)$, $f(y_1|G_1,D_1^C)$, and $f(y_2|G_2,D_2^C)$ are calculated. Using the validation data, the probability $P(G_1|D_1^V)$ and $P(G_2|D_2^V)$ are calculated; further $P(G_1'|D_1^V) = 1 - P(G_1|D_1^V)$ and $P(G_2'|D_2^V) = 1 - P(G_2|D_2^V)$. As explained earlier, the probabilities that G_1 and G_2 are correct can also be calculated using the reliability-based metric.

The unconditional PDF of Z needs to be calculated by considering four quantities:

1. $P(G_1 \cap G_2|D_1^V,D_2^V) = P(G_1|D_1^V)P(G_2|D_2^V)$
2. $P(G_1 \cap G_2'|D_1^V,D_2^V) = P(G_1|D_1^V)P(G_2'|D_2^V)$
3. $P(G_1' \cap G_2|D_1^V,D_2^V) = P(G_1'|D_1^V)P(G_2|D_2^V)$
4. $P(G_1' \cap G_2'|D_1^V,D_2^V) = P(G_1'|D_1^V)P(G_2'|D_2^V)$

Note the assumption that the two models G_1 and G_2 are independent. If the dependence is known, then it can be included in the calculation of the joint probabilities. Then, the unconditional PDF of Z is written as:

$$\begin{aligned} f_Z(z|D_1^C,D_1^V,D_2^C,D_2^V,H) = {} & P(G_1|D_1^V)P(G_2|D_2^V) f_Z(z|G_1,G_2,H) \\ & + P(G_1'|D_1^V))P(G_2|D_2^V) f_Z(z|G_1',G_2,H) \\ & + P(G_1|D_1^V)P(G_2'|D_2^V) f_Z(z|G_1,G_2',H) \\ & + P(G_1'|D_1^V)P(G_2'|D_2^V) f_Z(z|G_1',G_2',H) \end{aligned} \quad (19)$$

In Eq. (19), $f_Z(z|G_1,G_2,H)$ is calculated by propagating the posteriors of Y_1 and Y_2 through H, since both the models are correct; $f_Z(zG_1',G_2,H)$ is calculated by propagating the alternate PDF of Y_1 and the posterior of Y_2 through H, since only G_2 is

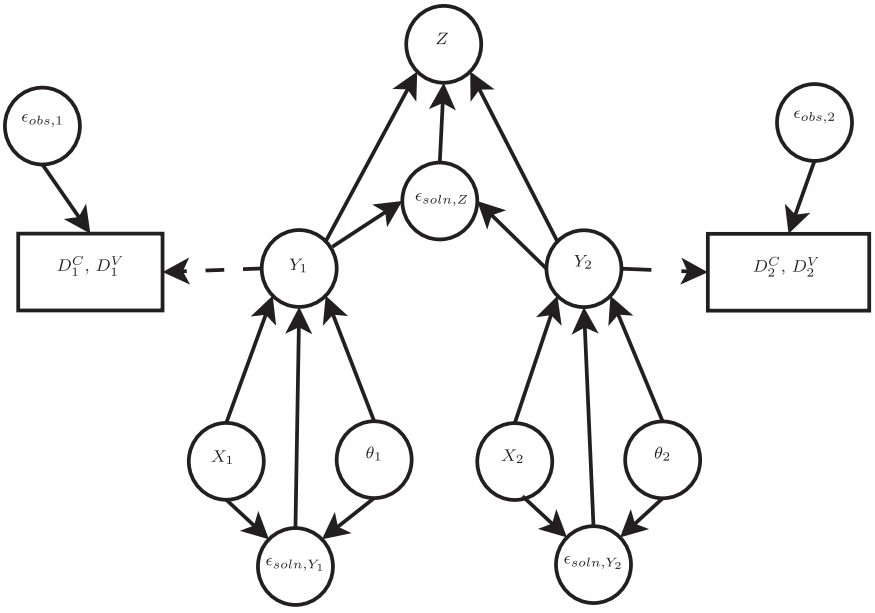

Fig. 6. Bayesian network: Type-I interaction between multiple models.

correct; similarly, $f_Z(z|G_1, G_2', H)$ is calculated by propagating the posterior of Y_1 and alternate PDF of Y_2, and $f_Z(z|G_1', G_2', H)$ is calculated by propagating the alternate PDFs of Y_1 and Y_2. If there are more than two lower-level models (G_1, G_2, G_3 and so on), the number of terms on the right-hand side of Eq. (19) increases exponentially, since the number of terms will be equal to 2^{n_m} where n_m is the number of models. Each of these terms indicate which subset of the models is correct. For example, in the case of three models, the event "$G_1 \cap G_2' \cap G_3'$" indicates that the model G_1 is correct whereas the models G_2 and G_3 are not. Similarly, the event "$G_1 \cap G_2 \cap G_3'$" indicates that the model G_2 is correct whereas the models G_1 and G_3 are not. Though Eq. (19) clearly distinguishes between all such possibilities, the computation of the right-hand side of this equation may be cumbersome due to the exponential number of terms. Such computational complexity can be easily avoided by first computing the unconditional PDFs of the lower-level outputs ($f(y_1|D_1^C, D_1^V)$ and $f(y_2|D_2^C, D_2^V)$ in this case) similar to Eq. (16), and then propagating these PDFs through the model H. Both the approaches will yield the same resultant PDF of Z, which accounts for the results of verification, validation, and calibration activities in both the models G_1 and G_2.

4. Multi-level models with Type-II interaction

Sometimes, a system model is developed using progressively complex models and corresponding experiments (isolated features, isolated physics, simplified geometry, scaled models, etc.). The experiments of lowest complexity (simplest geometry or single physics) have been referred to as unit-level experiments [39]. A higher-level experiment could include an assembly of units or combined physics.

A typical example of such a system is discussed in [51], where material level tests (lowermost level), performance of a single joint, and performance of three joints are used to calibrate underlying material and model parameters that are used in the overall system-level model. Usually, in such a system, the complexity increases going from the lower-level to the higher level (more physics, features, components, etc.). The response of a lower-level experiment may not be directly related to the system-level response. However, there are some system-level parameters that can be inferred using lower-level experiments.

Assume that a generic system-level model is given by:

$$Z = H(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\Psi}) \tag{20}$$

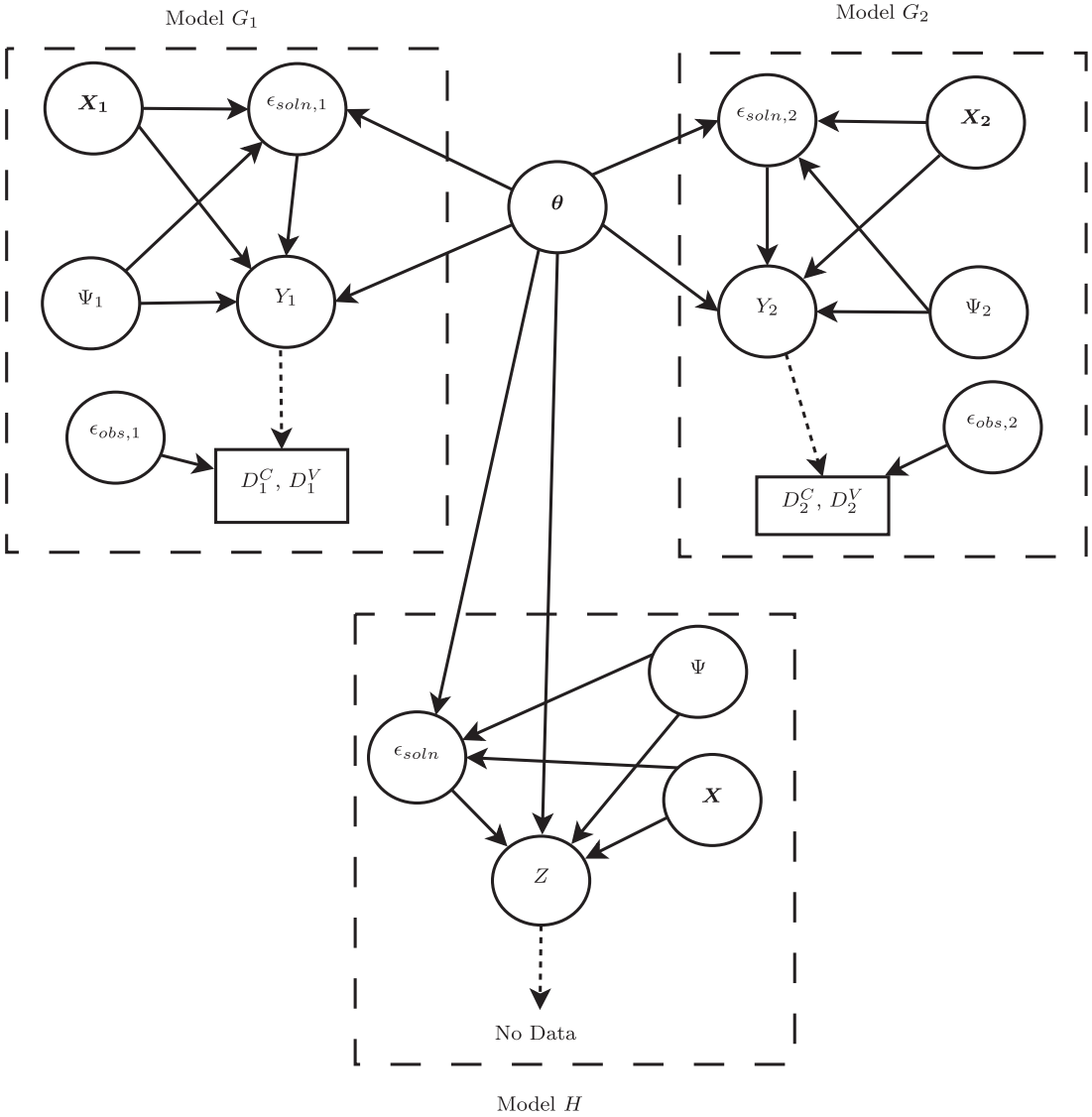

Fig. 7. Bayesian network: Type-II interaction between models.

In Eq. (20), Z is the system-level prediction, $\boldsymbol{\theta}$ is the set of model parameters which are calibrated based on lower-level models and tests, $\boldsymbol{\Psi}$ is the set of additional model parameters at the system-level, and $\boldsymbol{X}$ are the inputs.

Consider two lower-level models – G_1 and G_2. Both these models have common model parameters $\boldsymbol{\theta}$, but they have their own inputs ($\boldsymbol{X_1}$ and $\boldsymbol{X_2}$) and outputs (Y_1 and Y_2); in addition, they may have additional lower-level model parameters ($\boldsymbol{\Psi_1}$ and $\boldsymbol{\Psi_2}$).

$$Y_1 = G_1(\boldsymbol{\theta}, \boldsymbol{X_1}, \boldsymbol{\Psi_1})$$
$$Y_2 = G_2(\boldsymbol{\theta}, \boldsymbol{X_2}, \boldsymbol{\Psi_2}) \quad (21)$$

Assume that separate sets of data are available for calibration (D_1^C and D_2^C for levels 1 and 2 respectively) and validation (D_1^V and D_2^V for levels 1 and 2 respectively). Full system testing is not possible, i.e., no test data are available at the system-level (Z) and it is required to quantify the uncertainty in the system-level prediction using the data at the lower-levels (Y_1 and Y_2). The inputs, model parameters, outputs, and data at all levels are connected through a Bayesian network, as shown in Fig. 7.

4.1. Verification, calibration, and validation

The steps of verification, calibration, and validation in each model are similar to the previous sections, however the procedure for integration of these activities is different.

If $\boldsymbol{\theta}$ is estimated using each individual model (G_1 or G_2) and the corresponding calibration data (D_1^C or D_2^C), then the corresponding PDFs of the model parameter $\boldsymbol{\theta}$ are $f(\boldsymbol{\theta}|D_1^C, G_1)$ or $f(\boldsymbol{\theta}|D_2^C, G_2)$ respectively. The Bayesian network facilitates the simultaneous use of both models and the corresponding data to calibrate $\boldsymbol{\theta}$ and obtain the PDF $f(\boldsymbol{\theta}|D_1^C, D_2^C, G_1, G_2)$. This step of simultaneous calibration using multiple data sets from experiments of differing complexity is different from the calibration considered in Sections 2 and 3, where only one model and the corresponding calibration data were used to estimate $\boldsymbol{\theta}$. In order to integrate the results of verification, validation, and calibration in Section 4.2, all the PDFs, i.e., those calibrated using individual data sets ($f(\boldsymbol{\theta}|D_1^C, G_1)$ and $f(\boldsymbol{\theta}|D_2^C, G_2)$), as well as those calibrated using multiple data sets ($f(\boldsymbol{\theta}|D_1^C, D_2^C, G_1, G_2)$) are necessary.

The use of validation data is identical to the procedure in Sections 2 and 3. The quantities $P(G_1|D_1^V)$ and $P(G_2|D_2^V)$ are calculated using the Bayes factor metric; further $P(G_1'|D_1^V) = 1 - P(G_1|D_1^V)$ and $P(G_2'|D_2^V) = 1 - P(G_2|D_2^V)$. Alternatively, the reliability-based method can also be used to calculate this probability. Since the two models are assumed independent, $P(G_1 \cap G_2|D_1^V, D_2^V) = P(G_1|D_1^V)P(G_2|D_2^V)$.

4.2. Integration for overall uncertainty quantification

The proposed method for overall uncertainty quantification and integration of the above activities is different from Section 3 because the linking variables in this case are the model parameters whereas the linking variables in Section 3 were the outputs of lower-level models. While the unconditional PDF of the lower-level output was calculated in Section 3, it is now necessary to calculate the *unconditional PDF of the model parameter*, that also accounts for validation results. This is accomplished using the total probability theorem as:

$$\begin{aligned} f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|D_1^C, D_2^C, D_1^V, D_2^V) = {} & f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1 \cap G_2)P(G_1 \cap G_2|D_1^V, D_2^V) \\ & + f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1' \cap G_2)P(G_1' \cap G_2|D_1^V, D_2^V) \\ & + f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1 \cap G_2')P(G_1 \cap G_2'|D_1^V, D_2^V) \\ & + f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1' \cap G_2')P(G_1' \cap G_2'|D_1^V, D_2^V) \end{aligned} \quad (22)$$

Eq. (22) is expressed as the sum of four terms; the first term $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1 \cap G_2)$ is calculated for the case when both models are valid. This conditioning (i.e., the distribution of $\boldsymbol{\theta}$ being conditioned on G_1 and G_2) can be enforced by using both the models G_1 and G_2 for calibration using the data sets D_1^C and D_2^C. The resultant PDF is $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|D_1^C, D_2^C, G_1, G_2)$. If both the models have *high* probabilities of being valid (i.e., if $P(G_1 \cap G_2|D_1^V, D_2^V)$ is high), then this PDF is weighed correspondingly *high*, in Eq. (22). The second term $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1' \cap G_2)$ means that the PDF of $\boldsymbol{\theta}$ is calculated for the case when model G_1 is not valid but model G_2 is valid. To enforce this condition, only G_2 and D_2^V are used for calibration, and the resulting posterior PDF of $\boldsymbol{\theta}$ is denoted by $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|D_2^C, G_2)$. Similarly, the third term $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1 \cap G_2')$ is equal to $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|D_1^C, G_1)$, and is calculated by using only G_1 and D_1^V for calibration. Finally, the fourth term $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1' \cap G_2')$ is calculated under the condition that both G_1 and G_2 are invalid. In this case, $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G_1' \cap G_2')$ is simply equal to the prior PDF $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. If both the models have *high* probabilities of being invalid (i.e., if $P(G_1' \cap G_2'|D_1^V, D_2^V)$ is high), then the prior information is weighed accordingly *high*, because the posterior PDFs calculated using the models are unreliable (since the models themselves have low probabilities of being valid). On the other hand, if both the models have high probabilities of being valid, then a small weight is assigned to the joint prior PDF and a large weight is assigned to the joint posterior PDF because the posterior has been calculated using the models that are *highly* reliable (since the models themselves have high probabilities of being valid). Thus, the proposed approach provides a systematic way to weigh the results of uncertainty quantification before and after calibration, by using the results of model validation, and in the process, computes the PDF of $\boldsymbol{\theta}$ on the left-hand side of Eq. (22).

(In general, if there are n_m models, then the right-hand side of Eq. (22) has 2^{n_m} terms. Each of these terms, except the one corresponding to the case where all the models are not correct, needs to be computed through a Bayesian inference procedure. Hence, this may be a computational challenge in larger applications. In such scenarios, the use of an inexpensive surrogate model is recommended for such repeated Bayesian inference calculations, as performed in this paper).

The PDF $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|D_1^C, D_2^C, D_1^V, D_2^V)$ calculated in Eq. (22) accounts for the verification, calibration, and validation activities with respect to each of the lower-level models. This unconditional PDF is propagated through the system model $H(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\Psi})$, in order to quantify the uncertainty in the system-level response Z. Thus, similar to Section 3, it can be seen that the tools of conditional probability and total probability are directly useful for integrating verification, validation, and calibration, and thereby, aid in the quantification of the system-level prediction uncertainty.

5. Numerical example: single-level model

This section discusses a numerical example, where a single-level model is subject to verification, validation and calibration. The results of these activities are integrated to calculate the overall uncertainty in the response quantity.

5.1. Description of the problem

Consider steady state heat transfer in a thin wire of length L, with thermal conductivity k, and convective heat coefficient β. Assume that the heat source is $Q(x) = 25(2x - L)^2$, where x is measured along the length of the wire. For the sake of illustration, it is assumed that this problem is essentially one dimensional and that the solution can be obtained from the following boundary value problem [14]:

$$-k\frac{\partial^2 T}{\partial T^2} + \beta T = Q(x)$$

$$T(0) = T_0$$
$$T(L) = T_L \quad (23)$$

The length of the wire is assumed to be deterministic (L=4 m). The boundary conditions, i.e., the temperatures at the ends of the wire ($T(0)$ and $T(L)$), are assumed to be normally distributed with statistics N(0, 1). The thermal conductivity of the wire (k) is assumed to be normally distributed N(5, 0.2) with units W m^{-1}/°C. The convective heat coefficient (β) is an unknown parameter which needs to be estimated using calibration data (D^C); this quantity is assumed to have a normally distributed prior as N(0.5, 0.05). The goal of the model is to predict the temperature (Y) at the mid-point of the wire.

5.2. Verification, validation, and calibration

First, the differential equation in Eq. (23) is solved using a finite difference code. Three different discretization sizes are considered, and Richardson extrapolation [20] is used to calculate the solution approximation error which is used to correct the model prediction every time this differential equation is solved. Since there are four uncertain quantities (T_0, T_L, k, and the model parameter being updated, i.e., β), it is necessary to quantify the solution approximation error as a function of these four quantities. Every time the model prediction Y needs to be computed as a function of these four quantities, a mesh refinement study is conducted, i.e., three different mesh sizes are considered (with sizes 0.01, 0.005, and 0.0025) and Eq. (1) is used to compute the solution approximation error and hence, the corrected prediction.

It may be noted that, for this particular numerical example, linear diffusion dominates and therefore, the solution approximation error is not very sensitive to k. However, this behavior is not explicitly considered during verification. This is because of two reasons. First, it may be recalled that, in order to integrate the results of verification into calibration and validation, the solution approximation errors are computed and corrected whenever needed while performing calibration (update β using data $D^C = \{22; 23; 25; 26.1; 25.4\}$, in °C) and validation. In other words, during calibration, samples of the aforementioned four uncertain quantities are generated through slice sampling; for each generated sample the solution approximation error is quantified through mesh refinement. Since it is necessary to repeat this procedure for every generated sample, it does not matter whether the nominal value of k or the actually sampled value of k is used, from a computational point of view. The second reason is that the underlying physics behavior and the choice of numerical values are not exploited during the implementation of the proposed methodology. The code used to solve the differential equation in Eq. (23) is treated as if it were a black box model, in an effort to keep the illustration as general as possible. In specific problems, special features may be exploited to reduce computational effort.

The prior ($f_\Theta(\theta)$) and posterior ($f_\Theta(\theta|G, D^C)$) PDFs of β are shown in Fig. 8. Additional validation data ($D^V = \{24; 24.5; 24.6; 23.8\}$, in °C) is used to compute the probability that the temperature prediction model is correct, i.e., $P(G)$=0.84.

5.3. Integration and overall uncertainty quantification

The method developed in Section 2.4 is used to calculate the unconditional PDF of temperature using the principle of total probability, as shown in Fig. 9. This PDF integrates the results of verification, validation, and calibration to compute the overall uncertainty in the temperature at the mid-point of the wire.

Fig. 9 indicates three PDFs: (i) $f_Y(y|G, D^C)$ denotes the model prediction, (ii) $f_Y(y|G')$ denotes the prediction under the alternate hypothesis (assumed uniform; due to sampling errors and use of kernel density estimation for plotting, the PDF is not perfectly horizontal in Fig. 9), and (iii) $f_Y(y|G, D^C, D^V)$ which represents the PDF that integrates the validation result with the previous calibration and verification activities. The third PDF is referred to as the unconditional PDF of the temperature response, since it is not conditioned on the model form. Conventionally, the model prediction is used for performance prediction, failure analysis, and reliability analysis. Since failures are, generally, events with low probabilities of occurrence, it is important to be able to accurately capture tail probabilities in order to predict failures. For example, if the component is assumed to fail when the temperature is greater than 25 °C, then the model prediction PDF gives the failure probability as 0.0135, whereas the unconditional PDF gives the failure probability as 0.0390. Thus, it is clear that, using the raw model prediction (i.e., by simply considering the calibrated model, without accounting for the result of validation) underestimates the failure probability; whereas the proposed approach systematically includes the effect of model uncertainty in reliability analysis by integrating verification, validation, and calibration during system-level uncertainty quantification.

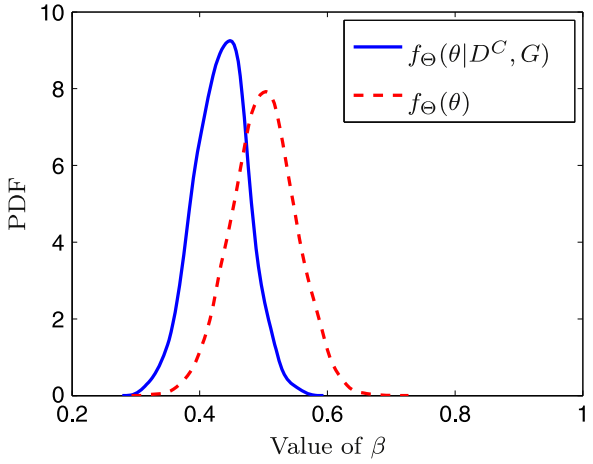

Fig. 8. PDF of convective heat coefficient (β).

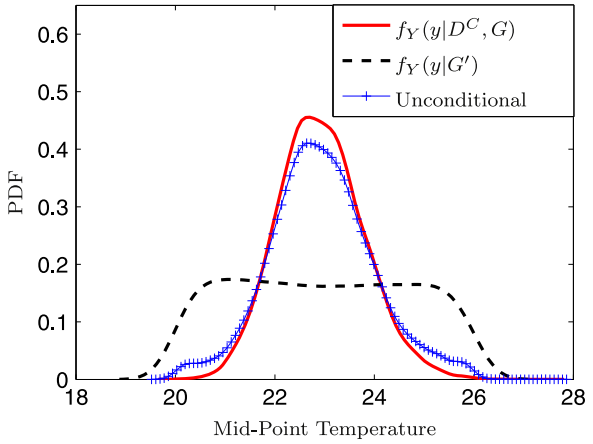

Fig. 9. PDF of mid-point temperature.

6. Numerical example: two models with Type-I interaction

This section discusses two models that represent thermal and electrical analyses, with Type-I interaction. This example is an extension of the heat conduction problem in Section 5; the

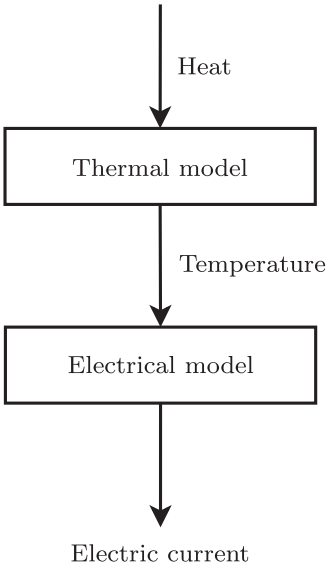

Fig. 10. Thermal electric analysis.

temperature rise in the wire causes change in the electrical resistance. The goal is to predict the system response, which is the electric current in the wire. Hence, the output of the lower-level model (temperature predictor in Eq. (23)), i.e., temperature, becomes an input to a higher level model (current predictor), as shown in Fig. 10.

Consider the same wire as in Section 5. Before application of the heat, the resistance of the wire is given in terms of the resistivity (ρ), the cross section area (A), and length (L) as:

$$R_{old} = \rho \frac{L}{A} \tag{24}$$

After steady state is reached, the mid-point temperature (Y) computed in Eq. (23) causes an increase in the resistance of the wire; this increase is evaluated using the coefficient of resistivity (α). The current through the wire when a 10 V voltage is applied is calculated as:

$$I = \frac{10}{R_{old}(1+\alpha Y)} \tag{25}$$

Assume that there is no electrical performance test data for the wire, and it is required to predict the uncertainty in the electrical current, by including the results of verification, validation, and calibration in the lower-level model. The two models and the associated sources of uncertainty are connected through a Bayesian network as shown in Fig. 11. The four uncertain quantities (T_0, T_L, k, β) are used to predict the mid-point temperature (Y) which is then used to compute the resistance (R) as a function of higher-level model parameters (A, α, I, ρ). Notice that the lower-level solution approximation error is actually a function of the aforementioned four uncertain quantities (T_0, T_L, k, β). Data (D^C, D^V) is available for comparison against Y, and ϵ_m is the measurement error associated with such data.

Since the thermal model used for temperature prediction has already been verified, calibrated, and validated, the unconditional PDF of the temperature is simply propagated through the current-predictor model to calculate the current in the wire. For the purpose of illustration, and to see the effect of uncertainty in Y on the uncertainty in electrical current (I), the other parameters of the current-predictor model (α, A, ρ) are chosen to be deterministic. The PDF of the current of the wire is shown in Fig. 12, for three cases.

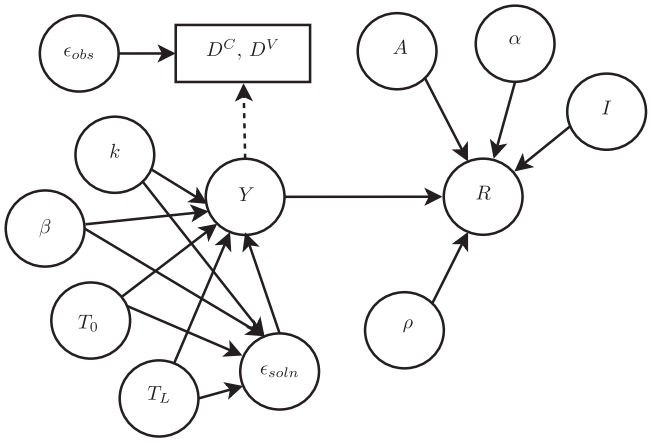

Fig. 11. Bayesian network: thermal electric analysis.

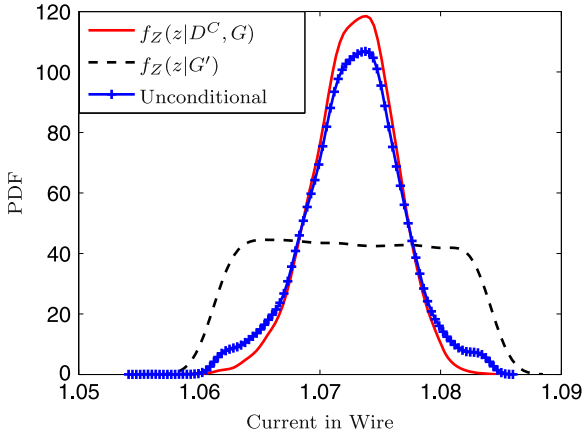

Fig. 12. PDF of current: system response.

The PDF $f_Z(z|G, D^C)$ is obtained by propagating the model prediction of the thermal model through the electrical model, and the PDF $f_Z(z|G')$ is obtained by propagating the alternate PDF of temperature ($f_Y(y|G')$) through the electrical model. The unconditional PDF ($f_Z(z|D^C, D^V)$) represents the current response by integrating verification, validation, and calibration activities for the lower-level heat conduction model. Similar to the previous example, the difference between $f_Z(z|G, D^C)$ and the unconditional $f_Z(z|D^C, D^V)$ can be quantified; for example, $1 - F_Z(z = 1.08|G) = 0.0086$ whereas $1 - F_Z(z = 1.08) = 0.0400$.

7. Numerical example: models with Type-II interaction

This section illustrates the methodology for an engineering system studied using multi-level models that exhibit Type-II interaction, through a numerical example which consists of a three-level structural dynamics problem, as shown in Fig. 13. This numerical example was developed at Sandia National Laboratories [61], as a model validation challenge problem.

7.1. Description of the problem

In the first-level, three spring–mass–dampers are integrated to form a subsystem. In the second-level, the integrated spring–mass–damper subsystem is mounted on a beam to form the overall system. The overall objective is to compute the system-level output (R) which is defined to be the maximum acceleration of mass m_3, under a given realization of random process loading [61] on the beam. The model to compute this system-level output is provided by Red-Horse and Paez

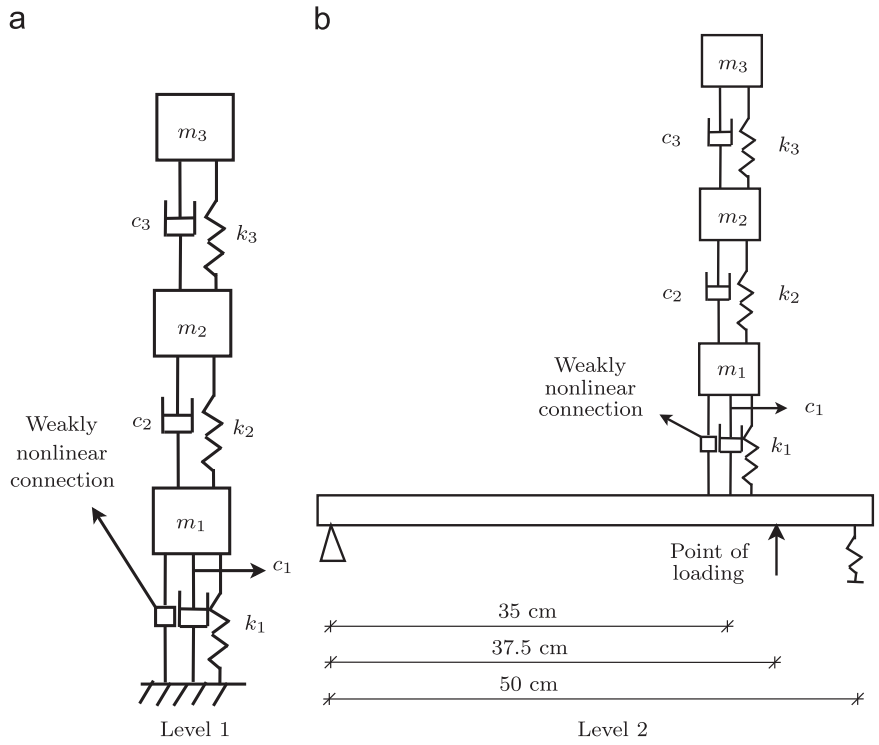

Fig. 13. Multi-level structural dynamics problem: (a) level 1 and (b) level 2.

[61]. This is the overall system-level model (denoted by Z); no test data are available at this level. The uncertainty in R needs to be computed based on information from lower-level data. Two types of tests can be performed, at each of the levels:

1. *Level* 1: The three mass assembly in Fig. 13(a) is tested under sinusoidal loading (amplitude = 10,000 and angular velocity = 10 rad s^{-1}), and the acceleration of the top mass m_3 is measured. A model (denoted by G) is built to predict this response x_1. The construction of this model is straightforward and can be found in several textbooks [80]. Let D_1 denote test data; similar to the previous sections, two sets of test data are available: D_1^C for calibration, and D_1^V for validation.
2. *Level* 2: The beam with the 3-mass subsystem in Fig. 13(b) is tested under sinusoidal loading (amplitude = 10,000 and angular velocity = 10 rad s^{-1}), and the acceleration of the top mass m_3 is measured. A model (denoted by H) is provided in [61] to predict this response x_2 (level-2 acceleration response of mass m_3).

Similar to the previous sections, two sets of test data are available: D_1^C and D_2^C for calibration, D_1^V and D_2^V for validation, corresponding to the aforementioned two levels.

In this numerical example, for the sake of illustration, the stiffness values k_1, k_2, and k_3 are identified as the parameters to be calibrated using available test data. An additional set of data is used to validate the lower-level models and all of this information is used to predict the system-level response R, defined earlier.

Prior distributions are assumed for k_1, k_2, and k_3 and later updated with test data to calculate posterior distributions. The system-level output R, in turn, is calculated by propagating the posterior distributions through the model Z. The numerical values (in SI units) of all the parameters are summarized in Table 1.

The mass of the beam is taken to be 0.1295 kg. Further numerical details of the beam are given in [61]. The calibration parameters (k_1, k_3, k_3), level-1 model prediction (x_1), level-2 model prediction (x_2), experimental data (D_1^C, D_1^V, D_2^C, D_2^V), experimental errors (ϵ_1, ϵ_2), and the solution approximation errors (ϵ_{soln} in each of the three levels) are connected using the Bayesian network, shown in Fig. 14. In order to construct this network, it is necessary to understand and represent the mathematical relationships between different variables using models. All the three output quantities depend on the model parameters (k_1, k_3, k_3) and the solution approximation errors (ϵ_{soln} in each of the three levels) that arise due to the use of the Gaussian process surrogate model. Further, note that the solution approximation errors themselves are functions of the model parameters (k_1, k_3, k_3). The experimental errors are associated with the available measurement data.

7.2. Surrogate modeling

Two Gaussian process surrogate models are constructed to reduce the computational effort; the first is to replace the model H while the second is to compute the response quantity R for the system of interest. These surrogate models are constructed based on the computer codes provided in [61]. Though these two functions are not as complicated as finite element analysis, it is necessary to perform several hundreds of thousands of evaluations of these functions during calibration, validation, and uncertainty quantification. That is why they need to be replaced with surrogate models. A design of experiments (based on 5000 Latin hypercube samples in three dimensions) was conducted in the parameter space (containing k_1, k_2, and k_3), and the resultant values (k_1, k_2, k_3 versus x_2, and k_1, k_2, k_3 versus R) are used as training input–output points to these two surrogate models. Gaussian process surrogate models were chosen against simpler regression models, mainly because no functional form needs to be assumed. The GP surrogates were further validated using a new set of data (100 Latin hypercube samples in three dimensions), and the maximum prediction error (by comparing the mean of the prediction against the output-training value) was observed to be less than 1%. Further, the maximum variance of the Gaussian was also less than 1% of the mean, thereby indicating good precision and accuracy of the Gaussian process surrogate model. This surrogate model is then used for prediction, and the results of

Table 1
Model parameters: structural dynamics problem.

Number	Mass (m) (in kg)	Damping (c) (in N s/m)	Prior mean of stiffness (μ_k) (in N/m)	Prior Std. Dev. of Mean (σ_k) (in N/m)
1	0.0125	0.023	5600	560
2	0.0193	0.021	11,000	1100
3	0.0351	0.031	93,000	9300

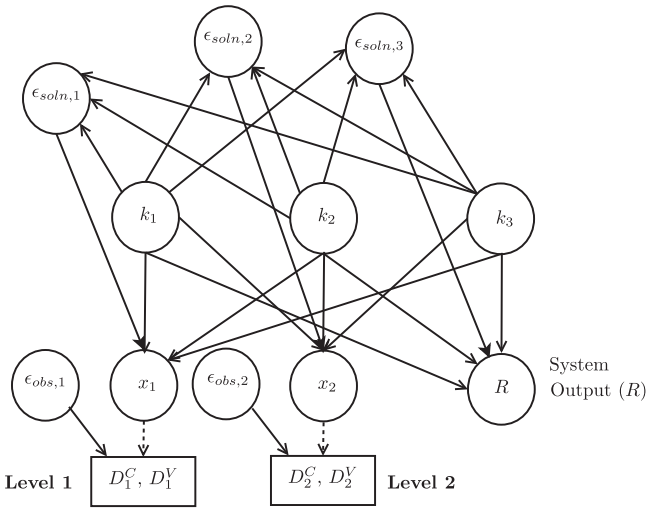

Fig. 14. Bayesian network: structural dynamics problem.

verification, validation, and calibration need to be integrated into such prediction.

7.3. Verification, calibration, and validation

Once the surrogate models are developed, they can be used for prediction, and the associated surrogate model error, which is stochastic as explained earlier in Section 2.1, is also included in the Bayesian network in Fig. 14, through the nodes ϵ_{soln}. This error must be explicitly included in both calibration and validation, in order to integrate the result of verification into the overall uncertainty quantification procedure.

The model parameters k_1, k_2 and k_3 are estimated using the calibration data, and shown in Figs. 15, 16 and 17 respectively. All the four PDFs ($f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G \cap H)$, $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G \cap H')$, $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G' \cap H)$, and $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}|G' \cap H')$ needed for the evaluation of Eq. (22) are also shown.

The next step is to validate the calibrated models. The models G and H are validated using two test measurements each, using the Bayesian hypothesis testing approach. The probabilities that the two models are correct are given by $P(G) = 0.25$ and $P(H) = 0.6$. It is assumed that the events that the models G and H are correct are independent; hence, $P(G \cap H) = 0.15$, $P(G \cap H') = 0.1$, $P(G' \cap H) = 0.45$, and $P(G' \cap H') = 0.3$. If the conditional probability that $P(G$ is correct $| H$ is correct) is available and not equal to $P(G$ is correct), then this information can be included to calculate $P(G \cap H)$. The fact that the models G and H are not fully accurate needs to be accounted for, since the calibration procedure was based on these models, and hence assumed the validity of these models. Therefore, it is necessary to compute the unconditional probability distribution of model parameters k_1, k_2, and k_3 by accounting for the probabilities that the models G and H are correct, as explained in the next subsection.

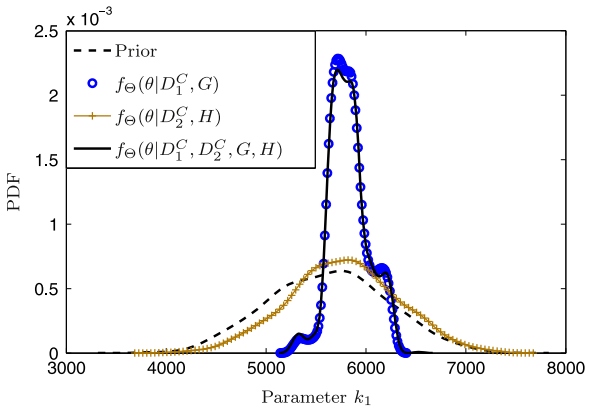

Fig. 15. PDF of parameter k_1.

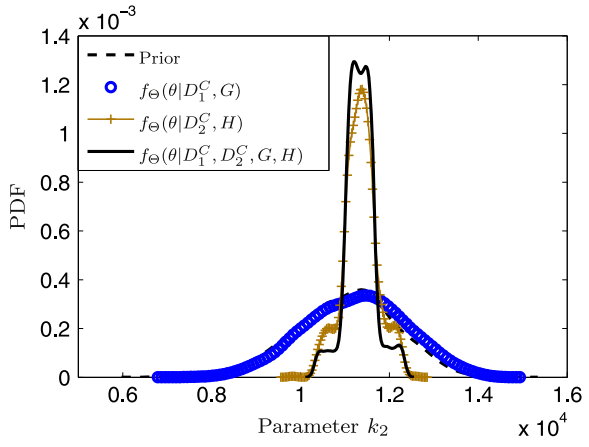

Fig. 16. PDF of parameter k_2.

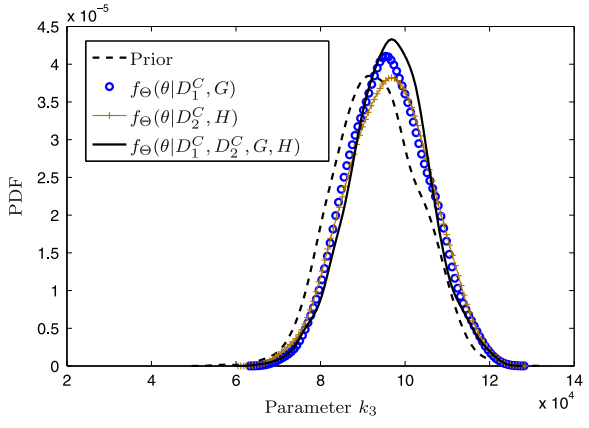

Fig. 17. PDF of parameter k_3.

7.4. Integration for overall uncertainty quantification

The unconditional PDFs of the calibration parameters k_1, k_2, and k_3 systematically integrate the results of validation and calibration. Note that the results of verification were accounted for during calibration and validation. Therefore, these unconditional PDFs represent the results of overall integration of verification, validation, and calibration. These unconditional PDFs are computed as per Eq. (22), and then, they are used to compute the system-level response R by propagating the uncertainty through the model Z; the resultant PDF is shown in Fig. 18.

Fig. 18 shows three PDFs; the first PDF ($f_R(r|G, H)$) is obtained by simply propagating the prior PDFs of the stiffnesses through

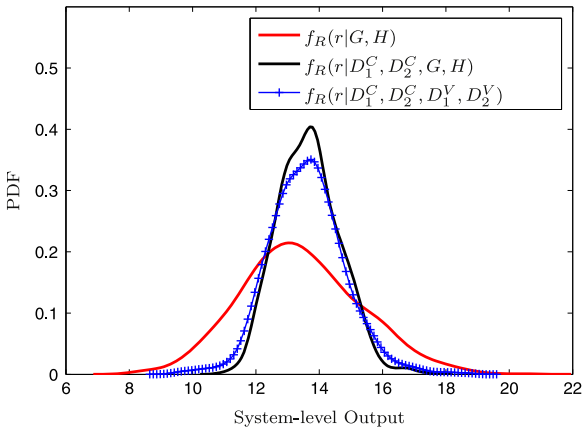

Fig. 18. PDF of system output R.

models and hence is representative of all knowledge before test data collection. The second PDF ($f_R(r|D_1^C,D_2^C,G,H)$) includes the effect of verification (by considering surrogate model uncertainty) and calibration (by updating parameters using calibration data) but does not include the effect of validation (i.e., assumes the correctness of the lower-level models). The third PDF ($f_R(r|D_1^C,D_2^C,D_1^V,D_2^V)$) is the unconditional PDF and accounts for the results of verification, validation, and calibration activities in the lower-level models. Similar to the previous section, the differences between the PDFs can be quantified. Differences in the tail region of the distribution of R have a larger impact on reliability calculations. For example, if failure is defined by the event $R>16$, then the raw model prediction (i.e., simply using the calibrated model for prediction, without accounting for the result of model validation) leads to a failure probability of 0.014, whereas the proposed method (that integrates verification, validation, and calibration) leads to a failure probability of 0.025.

8. Conclusion

Verification, validation, and calibration are significant activities in the process of model development. While methods for individual activities have been addressed in the past, the quantification of the combined effect of these activities on the overall system-level prediction uncertainty is addressed in this paper.

This topic is of specific importance in systems which are studied using multiple models where data may be available only at lower-levels and it may be desired to quantify the uncertainty in the system-level prediction using lower-level data. This paper proposed a Bayesian network-based methodology to integrate the various uncertainty quantification activities, including verification, validation, and calibration, performed at lower-levels, and rigorously account for their effects on the system-level prediction uncertainty. The Bayesian network is first used to connect the multiple models, the corresponding inputs, parameters, outputs, error estimates, and all available data. During the verification procedure, the solution approximation errors are quantified and accounted for. Both deterministic and stochastic errors are properly included, and the model is corrected before calibration and validation. Two independent sets of test data are considered: the first set is used to calibrate the model parameters and the second set is used to validate the calibrated model. The principles of conditional probability and total probability are then used to integrate the results of calibration and validation in order to compute the overall uncertainty in the model-based prediction.

The integration methodology is first developed for single-level models and then extended to multi-level systems that consist of interacting models. Two types of interactions are discussed in detail: (1) Type-I interaction, where the output of a lower-level model becomes an input to the higher-level model; and (2) Type-II interaction, where models and experiments at various levels of reduced complexity are used to infer system model parameters. While verification is performed before calibration and validation in both the cases, in order to account for the results of verification during calibration and validation, the procedure for the integration of calibration and validation results at lower-levels is different for Type-I and Type-II interactions; in the former case, the key idea is to compute the unconditional PDF of the *output* of the lower-level model, whereas in the latter case, the key idea is to compute the unconditional PDF of the system *model parameters*. If a system-level prediction is based on models with both types of interactions (both Type-I and Type-II), then the unconditional PDFs of the intermediate output and the parameters can both be used to compute the uncertainty in the overall system-level prediction uncertainty.

The proposed methodology offers considerable promise towards the quantification of margins and uncertainties in multi-level system prediction. While calibration and validation have previously been performed independently at individual levels, this methodology systematically integrates all such activities in order to compute the system-level prediction uncertainty, thereby aiding in risk-informed decision making with all available information. Only two types of interactions between multiple models were considered in this paper. Further research is necessary to extend the proposed integration methodology to other possible configurations of multiple models encountered in practical applications, including feedback coupling between some of the models.

Acknowledgments

The study in this paper was supported by funds from Sandia National Laboratories through Contract no. BG-7732 (Technical Monitor: Dr. Angel Urbina). The support is gratefully acknowledged.

References

[1] Alvin, KF, Oberkampf WL, Diegert KV, Rutherford BM. Uncertainty quantification in computational structural dynamics: a new paradigm for model validation. In: The 16th international modal analysis conference. Society for Experimental Mechanics, Inc., Santa Barbara, CA, vol. 2; 1998. p. 1191–98.
[2] Babuska I, Oden JT. Verification and validation in computational engineering and science: basic concepts. Comput Methods Appl Mech Eng 2004;193(36):4057–66.
[3] Roy CJ. Review of code and solution verification procedures for computational simulation. J Comput Phys 2005;205(1):131–56.
[4] AIAA. Guide for the verification and validation of computational fluid dynamics simulations. Technical report. Reston, VA: AIAA; 1998.
[5] Defense Modeling and Simulation Office. Verification, validation, and accreditation (VV & A) recommended practices guide. Technical report. Alexandria, VA: Office of the Director of Defense Research and Engineering; 1998.
[6] Oberkampf William L, Blottner Frederick G. Issues in computational fluid dynamics code verification and validation. AIAA J 1998;36(5):687–95.
[7] Oberkampf WL, Trucano TG. Verification and validation in computational fluid dynamics. Prog Aerosp Sci 2002;38(3):209–72.
[8] Benay R, Chanetz B, Délery J. Code verification/validation with respect to experimental data banks. Aerosp Sci Technol 2003;7(4):239–62.
[9] Roy Christopher J, Oberkampf William L. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Comput Methods Appl Mech Eng 2011;200(25–28):2131–44.
[10] Roache PJ. Verification of codes and calculations. AIAA J 1998;36(5):696–702.
[11] Roache PJ. Verification and validation in computational science and engineering. Albuquerque, NM: Hermosa Publishers; 1998.
[12] Oberkampf WL, Trucano TG, Hirsch C. Verification, validation, and predictive capability in computational engineering and physics. Appl Mech Rev 2004;57(5):345–84.
[13] Roy CJ, McWherter-Payne MA, Oberkampf WL. Verification and validation for laminar hypersonic flowfields, Part 1: verification. AIAA J 2003;41(10):1934–43.
[14] Rebba R, Mahadevan S, Huang S. Validation and error estimation of computational models. Reliab Eng Syst Saf 2006;91(10):1390–7.

[15] Liang B, Mahadevan S. Error and uncertainty quantification and sensitivity analysis in mechanics computational models. Int J Uncertain Quantif 2011;1(2):147–61.
[16] Rangavajhala S, Sura V, Hombal V, Mahadevan S. Discretization error estimation in multidisciplinary simulations. AIAA J 2011;49(12):2673–712.
[17] Ferziger JH, Peric M. Computational methods for fluid dynamics. New York: Springer-Verlag; 1996.
[18] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. Comput Methods Appl Mech Eng 1997;142(1–2):1–88.
[19] Oberkampf WL, DeLand SM, Rutherford BM, Diegert KV, Alvin KF. Error and uncertainty in modeling and simulation. Reliab Eng Syst Saf 2002;75(3):333–57.
[20] Richards SA. Completed Richardson extrapolation in space and time. Commun Numer Methods Eng 1997;13(7):573–82.
[21] Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. New York, NY: John Wiley & Sons, Inc.; 2000.
[22] Ghanem R, Spanos PD. Polynomial chaos in stochastic finite elements. J Appl Mech 1990;57(1):197–202.
[23] Buhmann MD. Radial basis functions: theory and implementations. Cambridge, UK: Cambridge University Press; 2003.
[24] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, MA, USA: The MIT Press; 2006.
[25] Seber GAF, Wild CJ. Nonlinear regression. New York: John Wiley; 1989.
[26] Edwards AWF. Likelihood. Cambridge, UK: Cambridge University Press; 1984.
[27] Pawitan Y. In all likelihood: statistical modelling and inference using likelihood. USA: Oxford University Press; 2001.
[28] Leonard T, Hsu JSJ. Bayesian methods. Cambridge, UK: Cambridge University Press; 2001.
[29] Lee PM. Bayesian statistics. London, UK: Arnold; 2004.
[30] Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J R Stat Soc: Ser B (Stat Methodol) 2001;63(3):425–64.
[31] Malinverno A, Briggs VA. Expanded uncertainty quantification in inverse problems: hierarchical Bayes and empirical Bayes. Geophysics 2004;69(4):1005–16.
[32] Park Inseok, Amarchinta Hemanth K, Grandhi Ramana V. A bayesian approach for quantification of model uncertainty. Reliab Eng Syst Saf 2010;95(7):777–85.
[33] Oliver TA, Moser RD. Accounting for uncertainty in the analysis of overlap layer mean velocity models. Phys Fluids 2012;24:075108.
[34] Sankararaman S, McLemore K, Mahadevan S, Bradford SC, Peterson LD. Test resource allocation in hierarchical systems using bayesian networks. AIAA J 2013;51(3):537–50.
[35] ASME. V&V 10-2006: guide for verification and validation in computational solid mechanics. New York: American Society of Mechanical Engineers (ASME); 2006.
[36] Coleman HW, Stern F. Uncertainties and CFD code validation. J Fluids Eng 1997;119(4):795–803.
[37] Oberkampf WL, Barone MF. Measures of agreement between computation and experiment: validation metrics. J Comput Phys 2006;217(1):5–36.
[38] Ferson S, Oberkampf WL, Ginzburg L. Model validation and predictive capability for the thermal challenge problem. Comput Methods Appl Mech Eng 2008;197(29):2408–30.
[39] Hills RG, Leslie IH. Statistical validation of engineering and scientific models: validation experiments to application. Technical report. Albuquerque, NM, USA; Livermore, CA, USA: Sandia National Labs; 2003.
[40] Urbina A, Paez TL, Hasselman T, Wathugala W, Yap K. Assessment of model accuracy relative to stochastic system behavior. In: The 44th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference; 2003.
[41] Gelfand AE, Dey DK. Bayesian model choice: asymptotics and exact calculations. J R Stat Soc Ser B (Methodol) 1994:501–14.
[42] Geweke John. Bayesian model comparison and validation. Am Econ Rev 2007;97(2):60–4.
[43] Zhang R, Mahadevan S. Bayesian methodology for reliability model acceptance. Reliab Eng Syst Saf 2003;80(1):95–103.
[44] Mahadevan S, Rebba R. Validation of reliability computational models using Bayes networks. Reliab Eng Syst Saf 2005;87(2):223–32.
[45] Sankararaman S, Mahadevan S. Model validation under epistemic uncertainty. Reliab Eng Syst Saf 2011;96(9):1232–41.
[46] Rebba R, Mahadevan S. Computational methods for model reliability assessment. Reliab Eng Syst Saf 2008;93(8):1197–207.
[47] Sankararaman S, Mahadevan S. Assessing the reliability of computational models under uncertainty. In: The 54th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference; 2013.
[48] Thacker BH, Paez TL. A simple probabilistic validation metric for the comparison of uncertain model and test results. In: The 16th AIAA non-deterministic approaches conference; 2014.
[49] Liu Y, Chen W, Arendt P, Huang HZ. Toward a better understanding of model validation metrics. Trans ASME J Mech Des 2011;133(7):071005.
[50] Ling You, Mahadevan Sankaran. Quantitative model validation techniques: New insights. Reliabil Eng Syst Saf 2013;111:217–31.
[51] Urbina Angel, Mahadevan Sankaran, Paez Thomas L. Quantification of margins and uncertainties of complex systems in the presence of aleatoric and epistemic uncertainty. Reliabil Eng Syst Saf 2011;96(9):1114–25.
[52] Sankararaman S, Ling Y, Mahadevan S. Uncertainty quantification and model validation of fatigue crack growth prediction. Eng Fract Mech 2011;78(7):1487–504.
[53] Sankararaman S, Mahadevan S. Model parameter estimation with imprecise and unpaired data. Inverse Probl Sci Eng 2012;20(7):1017–41.
[54] Sankararaman S, Mahadevan S. Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. Reliab Eng Syst Saf 2011;96(7):814–24.
[55] Jeffreys H. Theory of probability. USA: Oxford University Press; 1998.
[56] Jiang X, Mahadevan S. Bayesian risk-based decision method for model validation under uncertainty. Reliab Eng Syst Saf 2007;92(6):707–18.
[57] Sankararaman S, Ling Y, Shantz C, Mahadevan S. Uncertainty quantification in fatigue crack growth prognosis. Int J Progn Health Monit 2011;2(1):15.
[58] Sankararaman S, Ling Y, Shantz C, Mahadevan S. Inference of equivalent initial flaw size under multiple sources of uncertainty. Int J Fatigue 2011;33(2):75–89.
[59] Sankararaman S, Ling Y, Mahadevan S. Statistical inference of equivalent initial flaw size with complicated structural geometry and multi-axial variable amplitude loading. Int J Fatigue 2010;32(10):1689–700.
[60] Sankararaman S, Mahadevan S. Likelihood-based approach to multidisciplinary analysis under uncertainty. J Mech Des 2012;134:031008.
[61] Red-Horse JR, Paez TL. Sandia national laboratories validation workshop: structural dynamics application. Comput Methods Appl Mech Eng 2008;197(29–32):2578–84.
[62] Babuška I, Rheinboldt WC. A-posteriori error estimates for the finite element method. Int J Numer Methods Eng 1978;12(10):1597–615.
[63] Demkowicz L, Oden JT, Strouboulis T. Adaptive finite elements for flow problems with moving boundaries. Part I: variational principles and a posteriori estimates. Comput Methods Appl Mech Eng 1984;46(2):217–51.
[64] Rasmussen CE. Evaluation of Gaussian processes and other methods for non-linear regression [Ph.D. thesis]. University of Toronto; 1996.
[65] Rasmussen CE. The infinite Gaussian mixture model. Adv Neural Inf Process Syst 2000;12:554–60.
[66] Rasmussen CE. Gaussian processes in machine learning. In: Advanced lectures on machine learning; 2004. p. 63–71
[67] Santner TJ, Williams BJ, Notz W. The design and analysis of computer experiments. New York: Springer-Verlag; 2003.
[68] Bichon BJ, Eldred MS, Swile LP, Mahadevan S, McFarland JM. Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J 2008;46(10):2459–68.
[69] McFarland JM. Uncertainty analysis for computer simulations through validation and calibration [Ph.D. thesis]. Vanderbilt University; 2008.
[70] Cressie N. Spatial statistics. New York, NY: John Wiley and Sons; 1991.
[71] Chiles JP, Delfiner P. Geostatistics: modeling spatial uncertainty. New York, NY: Wiley-Interscience; 1999.
[72] Wackernagel H. Multivariate geostatistics: an introduction with applications. New York: Springer-Verlag; 2003.
[73] Arendt PD, Apley DW, Chen W. Quantification of model uncertainty: calibration, model discrepancy, and identifiability. J Mech Des 2012;134:100908.
[74] Ling Y, Mullins J, Mahadevan S. Options for the inclusion of model discrepancy in Bayesian calibration. In: The 16th AIAA non-deterministic approaches conference; 2014.
[75] Sankararaman S. Uncertainty quantification and integration in engineering systems [PhD thesis]. Vanderbilt University; 2012.
[76] Gilks WR, Richardson S, Spiegelhalter DJ. Markov chain Monte Carlo in practice. London, UK: Chapman & Hall; 1996.
[77] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953;21(6):1087.
[78] Gilks WR, Wild P. Adaptive rejection sampling for Gibbs sampling. J R Stat Soc Ser C (Appl Stat) 1992;41(2):337–48.
[79] Neal RM. Slice sampling. Ann Stat 2003:705–41.
[80] Chopra AK. Dynamics of structures. Upper Saddle River, New Jersey: Prentice Hall; 1995.