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/

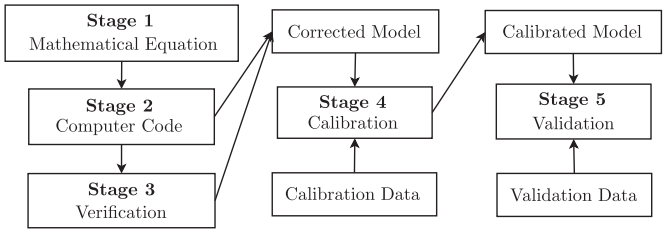

Fig. 1. Stages in model development.

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