Nuclear Engineering and Design 335 (2018) 339–355

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

# Inverse uncertainty quantification using the modular Bayesian approach based on Gaussian process, Part 1: Theory

Xu Wu[a,*], Tomasz Kozlowski[a], Hadi Meidani[b], Koroush Shirvan[c]

[a] Department of Nuclear, Plasma and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[b] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[c] Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

ARTICLE INFO

*Keywords:*
Inverse uncertainty quantification
Bayesian calibration
Gaussian process
Modular Bayesian
Model discrepancy

ABSTRACT

In nuclear reactor system design and safety analysis, the Best Estimate plus Uncertainty (BEPU) methodology requires that computer model output uncertainties must be quantified in order to prove that the investigated design stays within acceptance criteria. “Expert opinion” and “user self-evaluation” have been widely used to specify computer model input uncertainties in previous uncertainty, sensitivity and validation studies. Inverse Uncertainty Quantification (UQ) is the process to inversely quantify input uncertainties based on experimental data in order to more precisely quantify such ad-hoc specifications of the input uncertainty information.

In this paper, we used Bayesian analysis to establish the inverse UQ formulation, with systematic and rigorously derived metamodels constructed by Gaussian Process (GP). Due to incomplete or inaccurate underlying physics, as well as numerical approximation errors, computer models always have discrepancy/bias in representing the realities, which can cause over-fitting if neglected in the inverse UQ process. The model discrepancy term is accounted for in our formulation through the “model updating equation”. We provided a detailed introduction and comparison of the full and modular Bayesian approaches for inverse UQ, as well as pointed out their limitations when extrapolated to the validation/prediction domain. Finally, we proposed an improved modular Bayesian approach that can avoid extrapolating the model discrepancy that is learnt from the inverse UQ domain to the validation/prediction domain.

## 1. Introduction

During the last four decades, the importance of computer simulations has increased dramatically in furthering our understanding of the responses of engineered systems in real world. Large computer codes that implement complex mathematical models have been successfully applied in the design and performance assessment of real systems in many areas of scientific research. Computer modeling is especially significant to the nuclear engineering community, as physical experimentations are usually too costly or sometimes impossible.

### 1.1. Essential components of modeling and simulation

To bring up the motivation to perform inverse Uncertainty Quantification (UQ), we first briefly establish the definitions of some of the essential components that are used in the credibility evaluation of computer models. Note that these terminologies are widely used and frequently defined in many previously publications. The following definitions are based on the authors’ understanding and only used for this work.

1) *Verification*: “the process of determining that a model implementation accurately represents the developer’s conceptual description of the model and the solution to the model” (Oberkampf and Trucano, 2002, p. 215). In other words, verification aims to identify, quantify, and reduce errors during the mapping from mathematical model to a computer code.
2) *Code verification*: the process to access the reliability of the software coding, which includes two activities, *numerical algorithm verification* and *software quality engineering* (SQE) (Oberkampf and Roy, 2010). In other words, code verification deals with adequacy of the numerical algorithms and the fidelity of the computer programming to implement these algorithms.
3) *Solution verification*: also referred to as *calculation verification* (Trucano et al., 2006), or *numerical error estimation* (Oberkampf and Roy, 2010), is the process to evaluate the numerical accuracy of the solutions to a computer code. The primary difference between code and solution verification is that there is generally no known exact solution to the system of interest for the latter. Solution verification strongly depends on the quality and completeness of code verification, and both

* Corresponding author.
*E-mail address:* xuwu2@illinois.edu (X. Wu).

https://doi.org/10.1016/j.nucengdes.2018.06.004
Received 30 January 2018; Received in revised form 17 May 2018; Accepted 1 June 2018
0029-5493/ 

processes should be performed prior to validation, as defined below.

4) *Validation*: "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model" (Oberkampf and Trucano, 2002), p. 215). In other words, validation aims to determine the degree of accuracy of the considered model in representing real world phenomena. Verification and Validation together are often termed "V&V".
5) *Forward UQ*: the process of quantifying the uncertainties in Quantity-of-Interest (QoIs[1]) by propagating the uncertainties in input parameters through the computer model (Cacuci, 2003; Smith, 2014). QoIs predictions along with uncertainties are necessary for validation.
6) *Sensitivity analysis (SA)*: the study of how uncertainties in the QoIs of can be apportioned to various random input parameters (Saltelli et al., 2008). SA provides a ranking of the input parameters by their significance to QoIs.
7) *Optimization*: the process of maximizing or minimizing an objective function by systematically choosing input values from within an allowed set (Forrester and Keane, 2009; Queipo et al., 2005).
8) *Calibration*: the process of adjusting a set of input parameters implemented in the code so that the agreement of the computer code predictions with corresponding experimental data is maximized (Trucano et al., 2006).
9) *Data assimilation*: the process to incorporate observations of the actual system into the model state of a numerical model of that system (Evensen, 2009). Data assimilation can be treated as the calibration of dynamic models, which arise in many fields of geosciences such as weather forecasting.
10) *Benchmark*: "A benchmark is a choice of information that is believed to be accurate or true for use in verification, validation or calibration" (Trucano et al., 2006, p. 1333). For example, benchmarks can be measurements of QoIs from physical experiments or solutions from highly accurate numerical tests.

Fig. 1 shows the connections between some of these essential components of computer modeling. From Fig. 1 it is obvious that the forward UQ process always starts with characterization of the input uncertainties, for example, the mean values, variances, Probability Density Functions (PDFs), upper and lower limits, etc. Unfortunately, such information is not always readily available to the code users. Such condition is known as the "lack of input uncertainty information" issue. Up to now, in the uncertainty, sensitivity and validation studies of nuclear engineering, "expert opinion" or "user self-assessment" have been predominantly used (see reviews in (Wu and Kozlowski, 2017; Wu et al., 2017). Such ad-hoc specifications of input uncertainty information have been considered reasonable for a long time. However, these approaches are subjective and lack mathematical rigor, and can lead to inconsistencies.

The "lack of input uncertainty information" issue necessitates the research on inverse UQ. An early appearance of the term "inverse UQ" can be found in (Oberkampf and Trucano, 2002), in which it was also termed "backward problem". Other researchers have called it "inverse uncertainty propagation" (Unal et al., 2011). According to Oberkampf and Trucano, "The backward problem asks whether we can reduce the output uncertainty by updating the statistical model using comparisons between computations and experiments" (Oberkampf and Trucano, 2002, p. 256). In this paper, we will introduce the theory for inverse UQ under the Bayesian framework in an evolving manner, including the Bayesian formulation for inverse UQ, Gaussian Process (GP) metamodeling, full and modular Bayesian approaches, and finally an improved modular Bayesian approach.

[1] In some contexts, QoIs sometimes refer to inputs. In this work, QoIs only refers to the outputs, also called the Responses of Interest (RoIs).

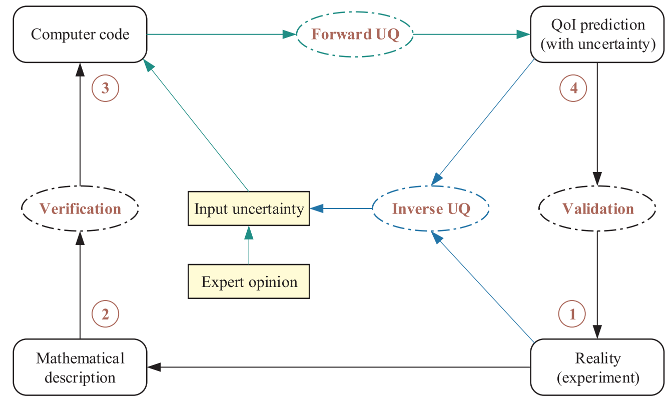


**Fig. 1.** Some essential parts of modeling and simulation (a non-exclusive list).

### 1.2. Inverse UQ vs. calibration

Inverse UQ, also referred to as *inverse problem* or *parameter estimation*, is the process to quantify the uncertainties of input parameters based on chosen experimental data. Such definition looks very similar with calibration. In this section we briefly discuss the relationship between inverse UQ and calibration.

Calibration can be classified as deterministic and statistical calibration (Campbell, 2006). *Deterministic calibration* merely determines the point estimates of best-fit input parameters such that the discrepancies between code output and experimental data can be minimized. However, *statistical calibration*, sometimes referred to as *Bayesian calibration* (Kennedy and O'Hagan, 2001); *probabilistic inversion* (Van Oijen et al., 2005) or *Calibration under Uncertainty (CUU)* (Trucano et al., 2006), produces statistical descriptions like distributions. In this sense, inverse UQ is same with Bayesian calibration and indeed they do share the same techniques. For example, both of them employ the Bayesian inference theory (Gelman et al., 2014) and explore the posterior PDF with Markov Chain Monte Carlo (MCMC) sampling (Gilks et al., 1995). They both favor surrogate models when the computational models are expensive. So what makes inverse UQ in the current study different from Bayesian calibration?

Inverse UQ only has very subtle differences with Bayesian calibration, (1): inverse UQ includes some techniques that implements the Expectation-Maximization (E-M) algorithm (Shrestha and Kozlowski, 2016) rather than sampling of the posterior PDF, even though the former is not as widely applicable as the latter; (2): they are usually performed with different motivations. Bayesian calibration aims at reducing the difference between simulation and observation, while inverse UQ emphasizes quantifying the input uncertainties. When the model outputs already agree very well with experimental data, we may conclude that no calibration is needed. However, the inverse UQ is still useful because the underlying uncertainties in model input parameters have to be quantified. Fig. 2 illustrates such a case, when the differences between simulation and measurement approximately follow Gaussian noise with a very small variance. In that case, calibration is unlikely to improve the agreement between simulation and observation. In essence, in cases where there is no need to do Bayesian calibration, inverse UQ may still be useful.

The advantage of inverse UQ (or Bayesian calibration) over deterministic calibration and "parameter tuning" is apparent: (1): firstly, information on QoIs from experiments is never sufficiently accurate to allow inference of the "true" or "exact" values of the input parameters. Instead, we can only hope to reduce our ignorance of the parameters by achieving less uncertainties in them (the so-called uncertainty reduction); (2): furthermore, it is difficult for deterministic calibration to quantify correlations between different calibration parameters. Correlations are usually calculated based on samples but deterministic calibration only produce point estimates of best-fit values; (3): thirdly,