Bayesian Methodology for Uncertainty Quantification in Complex Engineering Systems

Shankar Sankararaman and Sankaran Mahadevan

Abstract This chapter presents a Bayesian methodology for system-level uncertainty quantification and test resource allocation in complex engineering systems. The various component, subsystem, and system-level models, the corresponding parameters, and the model errors are connected efficiently using a Bayesian network. This provides a unified framework for uncertainty analysis where test data can be integrated along with computational models across the entire hierarchy of the overall engineering system. The Bayesian network is useful in two ways: (1) in a forward problem where the various sources of uncertainty are propagated through multiple levels of modeling to predict the overall uncertainty in the system-level response; and (2) in an inverse problem where the model parameters of multiple subsystems are calibrated simultaneously using test data. Test data available at multiple data are first used to infer model parameters, and then, this information is propagated through the Bayesian network to compute the overall uncertainty in the system-level prediction. Then, the Bayesian network is used for test resource allocation where an optimization-based procedure is used to identify tests that can effectively reduce the overall uncertainty in the system-level prediction. Finally, the overall Bayesian methodology for uncertainty quantification and test resource allocation is illustrated using three different numerical examples. While the first example is mathematical, the second and third examples deal with practical applications in the domain of structural mechanics.

S. Sankararaman (✉)
SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA
e-mail: shankar.sankararaman@gmail.com

S. Mahadevan
Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235, USA

S. Kadry and A. El Hami (eds.), *Numerical Methods for Reliability and Safety Assessment: Multiscale and Multiphysics Systems*, DOI 10.1007/978-3-319-07167-1_3,

1 Introduction

During the past 20 years, there has been an increasing need and desire to design and build engineering systems with increasingly complex architectures and new materials. These systems can be multi-level, multi-scale, and multi-disciplinary in nature, and may need to be decomposed into simpler components and subsystems to facilitate efficient model development, analysis, and design. The development and implementation of computational models is not only sophisticated and expensive, but also based on physics which is often not well understood. Therefore, when such models are used to design and analyze complex engineering systems, it is necessary to ensure their reliability and safety.

In order to facilitate efficient analysis and design, computational models are developed at the component-level, subsystem-level, and system-level. Each individual model may correspond to isolated features, or isolated physics, or simplified geometry of the original system. Typically, along the hierarchy of a multi-level system, the complexity of the governing physics increases, and hence, the complexity of the model increases, the cost of testing increases, and hence, the amount of available experimental data decreases. At the system-level, full-scale testing may not even be possible to predict the system performance under actual operating conditions. It is essential to quantify the overall uncertainty in the system-level prediction using the models and data available at all levels. The field of "quantification of margins and uncertainties" (QMU) has the goal of enabling this overall capability (Helton and Pilch 2011). This analysis is helpful to estimate the reliability and adjudge the safety of the overall engineering system.

An important challenge in this regard is to efficiently connect all of the models and experimental data available across the hierarchy of the entire system. This is not straightforward because there are several sources of uncertainty—physical variability, data uncertainty, and model uncertainty—at each level of the overall system. Further, the issue is complicated by the presence of individual model inputs, parameters, outputs, and model errors, all of which may be uncertain. It is important to use a computational approach that can not only facilitate integration across multiple levels but also provide a fundamental framework for the treatment of uncertainty.

This can be accomplished through the use of a Bayesian network that serves as an efficient and powerful tool to integrate multiple levels of models (including inputs, parameters, outputs, and errors of each and every model), the associated sources of uncertainty, and the experimental data at all different levels of hierarchy. The Bayesian network is based on Bayes' theorem, and can efficiently integrate all of the aforementioned information using the principles of conditional probability and total probability. It can be used for uncertainty propagation (forward problem), where the system-level prediction uncertainty is quantified by propagating all the sources of uncertainty at lower levels through the Bayesian network. The Bayesian network is also useful for model parameter calibration (inverse problem), where the data available at all levels can be simultaneously used to calibrate the underlying

parameters at different levels of models. Different types of sampling techniques can be used in conjunction with a Bayesian network; while Monte Carlo-sampling-based approaches are often used for uncertainty propagation (forward problem), Markov Chain Monte Carlo-based approaches are often used for model parameter estimation (inverse problem). Since sampling methods require several hundreds of thousands of evaluations of the system models that may be complicated to evaluate, the Bayesian approach can also include cost-efficient surrogate models like Gaussian process, as demonstrated by Kennedy and O'Hagan (2001). This chapter explains the fundamentals of Bayesian methodology and illustrates the use of Bayesian networks for uncertainty quantification in complex multi-level, multi-physics engineering systems. Methods for uncertainty quantification, model parameter estimation, and system-level uncertainty quantification are presented in detail.

Finally, it is also illustrated as to how these capabilities of Bayesian networks can be exploited to guide test resource allocation in hierarchical systems (Sankararaman et al. 2013). Test data available at multiple levels of system hierarchy are used for model parameter calibration, which in turn leads to a reduction of uncertainty in the model parameters; this reduced uncertainty is represented through the posterior distributions of the model parameters. When these posterior distributions are propagated through the Bayesian network, the uncertainty in the system-level response also decreases. Thus, testing can be used to evolve the system performance prediction and a common concern is to select that test design which leads to maximum reduction in the uncertainty (usually expressed through variance) of the system performance prediction. The tests need to be selected and designed with adequate precision (measurement error and resolution), and the simulations need to be developed with adequate resolution (model fidelity) to achieve the project requirements. This can be performed by embedding the Bayesian network within an optimization algorithm where the decision variables correspond to the test data. This formulation is very interesting because, in the past, model parameters have been calibrated with available test data; the difference now is that it is required to perform Bayesian calibration and assess the reduction in uncertainty in the system-level response even before actual testing is performed.

Two types of questions are of interest: (1) what tests to do? and (2) how many tests to do? Tests at different levels of the system hierarchy have different costs and variance reduction effects. Hence, the test selection is not trivial and it is necessary to identify an analytical procedure that helps in the optimum test resource allocation. However, current practice for this is, at best, based on simplified analysis and relevant experience, and at worst based on ad hoc rules, any of which may or may not result in truly conservative estimates of the margin and uncertainty. For multi-level systems, a rational test selection procedure should also incorporate the information from component-level and subsystem-level tests towards overall system level performance prediction. Recently, Sankararaman et al. (2013) developed an optimization-based methodology to identify the tests that will lead to maximum reduction in the system-level uncertainty, while simultaneously minimizing the cost of testing. This methodology is presented in detail, towards the end of this chapter, and illustrated using numerical examples.

2 Fundamentals of Bayesian Methodology

The Bayesian methodology is based on subjective probabilities, which are simply considered to be degrees of belief and quantify the extent to which the "statement" is supported by existing knowledge and available evidence. This may be contrasted with the frequentist approach to probability (classical approach to statistics), according to which probabilities can be assigned only in the context random physical systems and experiments. Uncertainty arising out of the model parameter estimation procedure is expressed in terms of confidence intervals, and it is not statistically meaningful to assign probability distributions to estimation parameters, since they are assumed to be "deterministic but unknown" in the frequentist methodology. This is a serious limitation, since it is not possible to propagate uncertainty after parameter estimation, which is often necessary in the case of model-based quantification of uncertainty in the system-level response. For example, if the uncertainty in the elastic modulus had been estimated using a simple axial test, this uncertainty cannot be used for quantifying the response in a plate made of the same material. Another disadvantage of this approach is that, when a quantity is not random, but unknown, then the tools of probability cannot be used to represent this type of uncertainty (epistemic). The subjective interpretation of probability, on which the Bayesian methodology relies upon, overcomes both of these limitations.

In the Bayesian approach, even deterministic quantities can be represented using probability distributions which reflect the subjective degree of the analyst's belief regarding such quantities. As a result, probability distributions can even be assigned to parameters that need to be estimated, and therefore, this interpretation facilitates uncertainty propagation after parameter estimation; this aspect of the Bayesian methodology is helpful for uncertainty integration across multiple models after inferring the underlying model parameters.

For example, consider the case where a variable is assumed to be normally distributed and it is desired to estimate the mean and the standard deviation based on available point data. If sufficient data were available, then it is possible to uniquely estimate these distribution parameters. However, in some cases, data may be sparse and therefore, it may be necessary to quantify the uncertainty in these distribution parameters. Note that this uncertainty is an example of epistemic uncertainty; the quantities may be estimated deterministically with enough data. Though these parameters are actually deterministic, the Bayesian methodology can calculate probability distributions for the distribution parameters, which can be easily used in uncertainty propagation. The fundamentals of Bayesian philosophy are well established in several textbooks (Calvetti and Somersalo 2007; Lee 2004; Leonard and Hsu 2001; Somersalo and Kaipio 2004), and the Bayesian approach is being increasingly applied to engineering problems in recent times, especially to solve statistical inverse problems. This section provides an overview of the fundamentals of the Bayesian approach, and later sections illustrate the application of Bayesian methods to uncertainty quantification in complex engineering systems.

2.1 Bayes Theorem

Though named after the eighteenth century mathematician and theologian Thomas Bayes (Bayes and Price 1763), it was the French mathematician Pierre-Simon Laplace who pioneered and popularized what is now called Bayesian probability (Stigler 1986, 2002). For a brief history of Bayesian methods, refer to Fienberg (2006). The law of conditional probability is fundamental to the development of Bayesian philosophy:

$$P(AB) = P(A|B)P(B) = P(B|A)P(A) \tag{1}$$

Consider a list of mutually exclusive and exhaustive events A_i ($i = 1$ to n) that together form the sample space. Let B denote any other event from the sample space such that $P(B) > 0$. Based on Eq. (1), it follows that:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \tag{2}$$

What does Eq. (2) mean? Suppose that the probabilities of events A_i ($i = 1$ to n) are known to be equal to $P(A_i)$ ($i = 1$ to n) before conducting any random experiments. These probabilities are referred to as prior probabilities in the Bayesian context. Suppose that a random experiment has been conducted and event B has been observed. In the light of this data, the so-called posterior probabilities $P(A_i|B)$ ($i = 1$ to n) can be calculated using Eq. (2).

The quantity $P(B|A_i)$ is the probability of observing the data conditioned on A_i. It can be argued that event B has "actually been observed," and there is no uncertainty regarding its occurrence, which renders the probability $P(B|A_i)$ meaningless. Hence, researchers "invented" new terminology in order to denote this quantity. In earlier days, this quantity was referred to as "inverse probability," and since the advent of Fisher (Aldrich 2008; Jeffreys 1998) and Edwards (1984), this terminology has become obsolete, and has been replaced by the term "likelihood." In fact, it is also common to write $P(B|A_i)$ as $L(A_i)$.

2.2 Bayesian Inference

The concept of Bayes theorem can be extended from the discrete case to the continuous case. Consider the context of statistical inference where a set of parameters $\boldsymbol{\theta}$ needs to be inferred. All the current knowledge regarding this parameter is represented in the form of a prior distribution denoted by $f'(\boldsymbol{\theta})$. The choice of the prior distribution reflects the subjective knowledge of uncertainty regarding the variable before any observation. It is assumed that the prior distribution is able to

explain the data with some degree of uncertainty; in other words, there exists a nonempty set E such that $\forall\ \theta \in E$, the prior probability density function (PDF) and likelihood values evaluated $\forall\ \theta \in E$ are both nonzero.

Measurement data (D) is collected on a quantity which depends on the parameter ($\boldsymbol{\theta}$). This information is then used to update the distribution of θ to arrive at the posterior distribution ($f''(\boldsymbol{\theta})$), as:

$$f''(\boldsymbol{\theta}) = \frac{L(\boldsymbol{\theta}) f'(\boldsymbol{\theta})}{\int L(\boldsymbol{\theta}) f'(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{3}$$

In Eq. (3), $L(\boldsymbol{\theta})$ is the likelihood function of $\boldsymbol{\theta}$ and is proportional to $P(D|\boldsymbol{\theta})$, i.e. probability of observing the data D conditioned on the parameter $\boldsymbol{\theta}$. Typically, data D is available in the form of independent, paired input–output combinations ($\boldsymbol{x}_i$ versus x_i, where i varies from 1 to n), where the input $\boldsymbol{X}$ and output Y are related to one another as:

$$y = G(\boldsymbol{x}, \boldsymbol{\theta}) \tag{4}$$

Considering an observation error ϵ that is assumed to follow a Gaussian distribution with zero mean and standard deviation σ, the likelihood can be written as:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{(2\pi)}} \exp - \left[\frac{(y_i - G(\boldsymbol{x_i}, \boldsymbol{\theta}))^2}{2\sigma^2} \right] \tag{5}$$

Note that the above equation assumes that the numerical value of σ is known. If this quantity is unknown, σ may be considered to be an argument to the likelihood function and updated along with θ. Equation (5) is substituted in Eq. (3), and the posterior distribution ($f''(\boldsymbol{\theta})$) can be computed.

Note that the denominator on the RHS of Eq. (3) is simply a normalizing constant, which ensures that $f''(\boldsymbol{\theta})$ is a valid PDF, i.e., the integral of the PDF is equal to unity. So, Eq. (3) is sometimes written as:

$$f''(\boldsymbol{\theta}) \propto L(\boldsymbol{\theta}) f'(\boldsymbol{\theta}) \tag{6}$$

The posterior in Bayesian inference is always known only up to a proportionality constant and it is necessary generate samples from this posterior for uncertainty analysis. When there is only one parameter, the proportionality constant can be calculated through one-dimensional integration. Often, multiple parameters may be present, and hence, multi-dimensional integration may not be affordable to calculate the proportionality constant. Therefore, a class of methods popularly referred to as Markov Chain Monte Carlo (MCMC) sampling is used to generate samples from the Bayesian posterior. In general, these methods can be used when it is desired to generate samples from a PDF which is known only up to a proportionality constant. The topic of MCMC will be discussed in detail later in this chapter, in Sect. 3.

2.3 Notes on the Likelihood Function

The likelihood function is defined as the probability of observing data conditioned on the parameters, i.e. $L(\boldsymbol{\theta}) = P(D|\boldsymbol{\theta})$; note that, since the *data* (D) *has actually been observed*, the terminology "probability of observing the data" is physically meaningless. Therefore, as explained earlier in Sect. 2.1, this quantity was renamed as "the likelihood." The likelihood function does not follow the laws of probability, and must not be confounded with probability distributions or distribution functions. In fact, Edwards (1984) explains that the likelihood function is meaningful only up to a proportionality constant; the relative values of the likelihood function are alone significant and the absolute values are not of interest.

The concept of likelihood is used in the context of both physical probabilities (frequentist) and subjective probabilities, especially in the context of parameter estimation. In fact, Edwards (1984) refers to the likelihood method as the third or middle way.

From a frequentist point of view (the underlying parameters are deterministic), the likelihood function can be maximized in order to obtain the maximum likelihood estimate of the parameters. According to Fisher (1912), the popular least-squares-based optimization methodology is an indirect approach to parameter estimation and one can "solve the real problem directly" by maximizing the "probability of observing the given data" conditioned on the parameter θ (Aldrich 1997; Fisher 1912). Further, it is also possible to construct likelihood-based confidence intervals for the inferred parameters (Pawitan 2001).

On the other hand, the likelihood function can also be interpreted using subjective probabilities. Singpurwalla (2006, 2007) explains that the likelihood function can be viewed as a collection of "weights" or "masses" and therefore is meaningful only up to a proportionality constant (Edwards 1984). In other words, if $L(\theta^{(1)}) = 10$, and $L(\theta^{(2)}) = 100$, then it is 10 ten times more likely for $\theta^{(2)}$ than $\theta^{(1)}$ to correspond to the observed data. The entire likelihood function can be used in Bayesian inference, as in Eq. (3), in order to obtain the entire PDF of the parameters.

3 MCMC Sampling

The class of MCMC methods can be used to generate samples from an arbitrary probability distribution, especially when the CDF is not invertible or when the PDF is known only up to a proportionality constant. In Sect. 2.2, it was explained that the latter is the case in Bayesian inference, where the objective is to compute the posterior distribution. Therefore, MCMC sampling can be used to draw samples from the posterior distribution, and these samples can be used in conjunction with kernel density estimation (Rosenblatt 1956) procedure to construct the posterior distribution.

There are several algorithms which belong to the class of MCMC sampling methods. Two such algorithms, the Metropolis algorithm (Metropolis et al. 1953) and the slice sampling (Neal 2003) algorithm are discussed below.

3.1 The Metropolis Algorithm

Assume that a function that is proportional to the PDF is readily available, as $f(x)$; this means that $f(x)$ is not a valid PDF because $\int f(x)dx \neq 1$. For the purpose of illustration, consider the one-dimensional case, i.e. $x \in \mathbb{R}$. The following steps constitute the algorithm in order to generate samples from the underling PDF. Note that the function $f(x)$ is always evaluated at two points and the ratio is only considered; the effect of the unknown proportionality constant is therefore nullified.

1. Set $i = 0$ and select a starting value x_0 such that $f(x_0) \neq 0$.
2. Initialize the list of samples $X = \{x_0\}$.
3. Repeat the following steps; each repetition yields a sample from the underlying PDF.
   (a) Select a prospective candidate from the proposal density $q(x^*|x_i)$. The probability of accepting this sample is equal to $\frac{f(x^*)}{f(x_i)}$.
   (b) Calculate acceptance ratio $\alpha = \min\left(1, \frac{f(x^*)}{f(x_i)}\right)$.
   (c) Select a random number u, uniformly distributed on [0, 1].
   (d) If $u < \alpha$, then set $x_{i+1} = x^*$, otherwise set $x_{i+1} = x_i$.
   (e) Augment the list of samples in X by x_{i+1}.
   (f) Increment i, i.e. $i = i + 1$.
4. After the Markov chain converges, the samples in X can be used to construct the PDF of X using kernel density estimation.

The common practice is to generate a few hundreds of thousands of samples and discard the first few thousand samples to ensure the convergence of the Markov Chain.

The Metropolis algorithm (Metropolis et al. 1953) assumes that the proposal density is symmetric, i.e. $q(x^*|x_i) = q(x_i|x^*)$. A generalization of this algorithm assumes asymmetric proposal density functions $q_1(x^*|x_i)$ and $q_2(x_i|x^*)$; this algorithm is referred to as Metropolis–Hastings algorithm (Hastings 1970). The only difference is that the probability of accepting the prospective candidate is calculated as $\frac{f(x^*)q_2(x_i|x^*)}{f(x_i)q_1(x^*|x_i)}$.

3.2 Slice Sampling

Consider the same function $f(x)$, i.e. the PDF of X, known up to a proportionality constant. The steps of the slice sampling algorithm are as follows:

1. Set $i = 0$ and select a starting value x_0 such that $f(x_0) \neq 0$.
2. Draw a random number y from the uniform distribution $[0, f(x)]$.
3. Consider the set $f^{-1}[y, \infty)$; note that this set may not be convex, especially when the target distribution is multi-modal. Select a sample which is uniformly distributed on this set. Assign $i = i + 1$, and call this sample x_i.
4. Repeat Steps 1–3 to generate multiple samples of X and construct the PDF of X using kernel density estimation.

In contrast with the previously discussed Metropolis algorithm, the slice sampling algorithm is not a acceptance–rejection algorithm.

3.3 MCMC Sampling: Summary

In addition to the above algorithms, other MCMC sampling algorithms such as Gibbs sampling (Geman and Geman 1984), multiple-try Metropolis (Liu et al. 2000), and Metropolis-within-Gibbs (Roberts and Rosenthal 2006) are also discussed in the literature. One critical disadvantage of MCMC sampling approaches is that they may require several hundreds of thousands of samples, and in turn, several hundreds of thousands of evaluations of G in Eq. (4), which may be computationally prohibitive. Therefore, it is common in engineering to replace G (which may be a complicated physics-based model) with an inexpensive surrogate, such as the Gaussian process surrogate model.

4 Gaussian Process Surrogate Modeling

The use of sampling techniques involves repeated evaluations of mathematical models, which may be computationally intensive. One approach to overcome this computational difficulty is to make use of surrogate models to replace the original physics-based model. A few evaluations of the original model are used to train this inexpensive, efficient surrogate model. Different types of surrogate modeling techniques such as polynomial response surface (Rajashekhar and Ellingwood 1993), polynomial chaos expansion (Ghanem and Spanos 1990), support vector regression (Boser et al. 1992), relevance vector regression (Tipping 2001), and Gaussian process interpolation (Bichon et al. 2008; Rasmussen 2004; Santner et al. 2003) have been investigated in the literature.

The Gaussian process interpolation is a powerful technique based on spatial statistics and is increasingly being used to build surrogates to expensive computer simulations for the purposes of optimization and uncertainty quantification (Bichon et al. 2008; Rasmussen 2004; Santner et al. 2003). The GP model is preferred in this research because (1) it is not constrained by functional forms; (2) it is capable of representing highly nonlinear relationships in multiple dimensions; and (3) can estimate the prediction uncertainty which depends on the number and location of training data points.

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 vs. 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 n input combinations simultaneously. Then, the joint density of the output values y_P can be calculated as:

$$p(y_P|x_P, x_T, y_T; \Theta) \sim N(m, S) \tag{7}$$

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:

$$\begin{aligned} m &= K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} y_T \\ S &= K_{PP} - K_{PT}(K_{TT} + \sigma_n^2 I)^{-1} K_{TP} \end{aligned} \tag{8}$$

In Eq. (8), 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 composed of squared exponential terms, where each element of the matrix is computed as:

$$K_{ij} = K(x_i, x_j; \Theta) = -\frac{\theta}{2}\left[\sum_{q=1}^{d} \frac{(x_{i,q} - x_{j,q})^2}{l_q}\right] \tag{9}$$

Note that all of the above computations require the estimate of the hyperparameters Θ; the multiplicative term (θ), the length scale in all dimensions (l_q, $q = 1$ to d), and the noise standard deviation (σ_n) constitute these hyperparameters ($\Theta = \{\theta, l_1, l_2 \ \ldots \ l_d, \sigma_n\}$). As stated earlier, these hyperparameters are estimated based on the training data by maximizing the following log-likelihood function:

$$\log p\left(y_T|x_T;\Theta\right) = -\frac{y_T^T}{2}\left(K_{TT}+\sigma_n^2 I\right)^{-1} y_T - \frac{1}{2}\log\left|\left(K_{TT}+\sigma_n^2 I\right)\right| + \frac{d}{2}\log(2\pi) \tag{10}$$

Once the hyperparameters are estimated, then the Gaussian process model can be used for predictions using Eq. (8). Note that the "hyperparameters" of the Gaussian process are different from the "parameters" of a generic parametric model (e.g., linear regression model). This is because, in a generic parametric model, it is possible to make predictions using only the parameters. On the contrary, in the case of the Gaussian process model, all the training points and the hyperparameters are both necessary to make predictions, even though the hyperparameters may have estimated previously. For details of this method, refer to Bichon et al. (2008); Chiles and Delfiner (1999); Cressie (1991); McFarland (2008); Rasmussen (1996, 2004); Santner et al. (2003), and Wackernagel (2003).

An important issue in the construction of the Gaussian process model is the selection of training points. In general, the training points may arise out of field experiments or may be generated using a computer code. Model parameter estimation considers the latter case and hence, there is no noise in the data, thereby eliminating σ_n from the above equations. Adaptive techniques can be used to select training points for the GP model, in order to construct the response surface to a desired level of accuracy or precision. Since the GP model is capable of estimating the variance in model output, a variance minimization algorithm proposed by McFarland (2008) identifies the next training point at the input variable value which corresponds to the largest variance. This selection algorithm is repeated and training points are adaptively identified until the estimated variance is below a desired threshold. Alternatively, another training point selection algorithm has been developed by Hombal and Mahadevan (2011), where the focus is to select successive training points so that the bias error in the surrogate model is minimized.

Once the training points are selected and the surrogate model is constructed, it can be used for several uncertainty quantification activities such as uncertainty propagation [through Monte Carlo simulation (MCS)], inverse analysis and parameter estimation (through MCMC simulation), and sensitivity analysis. It must be noted that the replacement of a complex computer simulation with an inexpensive surrogate leads to approximations; therefore, it is important to include the effect of this approximation in the procedure for overall uncertainty quantification (Sankararaman 2012).

5 Bayesian Networks

The previous sections of this chapter discussed certain fundamental concepts of the Bayesian methodology, in general. The Bayesian inference approach for model parameter estimation was presented, and the use of MCMC sampling and the importance of using Gaussian process surrogate models were explained. Most of this discussion dealt with single-level models that may represent a component

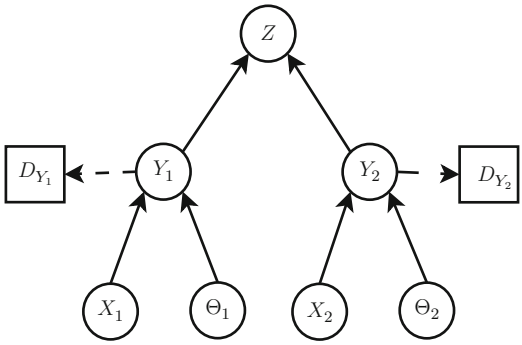


Fig. 1 Bayesian network illustration

or a subsystem. Since complex engineering systems consist of many components and subsystems, it is important to integrate information across all components and subsystems, in order to compute the overall uncertainty in the system-level prediction. This goal can be accomplished through the use of a Bayesian network.

A Bayesian network (Heckerman 2008; Jensen 1996) is an acyclic, graphical representation of conditional probability relationships between uncertain quantities. Each uncertain quantity is represented as a node and successive links are connected to each other using unidirectional arrows that express dependence in terms of conditional probabilities. Disconnected nodes imply independence between the corresponding random variables. Figure 1 shows a conceptual Bayesian network that aids in uncertainty quantification across multiple levels of models and observed data. Circles correspond to uncertain variables and squares represent observed data. A solid line arrow represents a conditional probability link, and a dashed line arrow represents the link of a variable to its observed data if available.

In Fig. 1, a system level output Z is a function of two subsystem level quantities Y_1 and Y_2; in turn, Y_1 is a function of subsystem-level input X_1 and model parameter Θ_1, and similarly, Y_2 is a function of subsystem-level input X_2 and model parameter Θ_2. For example, in a beam deflection study, the applied force is an input, the elastic modulus is a model parameter, while the deflection is measured and a model is built to predict the same. Experimental data D_{Y_1} and D_{Y_2} are available for comparison with the respective model predictions Y_1 and Y_2.

5.1 Uncertainty Propagation: Forward Problem

In the forward problem, the probability distributions of the inputs (X_1 and X_2) and model parameters (Θ_1 and Θ_2) are known or assumed, and these distributions are used to calculate the PDF of Y_1 and Y_2, which in turn are used to calculate the PDF of the system-level output Z, as:

$$f_Z(z) = \int f_Z(z|y_1, y_2) f_{Y_1}(y_1) f_{Y_2}(y_2) f_{Y_1}(y_1) f_{Y_2}(y_2) dy_1 dy_2$$
$$f_{Y_1}(y_1) = \int f_{Y_1}(y_1|x_1, \theta_1) f_{X_1}(x_1) f_{\Theta_1}(\theta_1) dx_1 d\theta_1$$
$$f_{Y_2}(y_2) = \int f_{Y_2}(y_2|x_2, \theta_2) f_{X_2}(x_2) f_{\Theta_2}(\theta_2) dx_2 d\theta_2 \quad (11)$$

Note that uppercase letters are used to denote random variables and the corresponding lowercase letters are used to denote realizations of those random variables. Equation (11) can be solved using methods of uncertainty propagation such as MCS, first-order reliability method (FORM), and second-order reliability method (SORM) (Haldar and Mahadevan 2000).

5.2 *Inference: Inverse Problem*

In the inverse problem, the probability densities of the model parameters (Θ_1 and Θ_2 in Fig. 1) can be updated based on the observed data (D_{Y_1} and D_{Y_2}) using Bayes' theorem as:

$$f_{\Theta_1,\Theta_2}(\theta_1, \theta_2|D_{Y_1}, D_{Y_2}) = CL(\theta_1, \theta_2) f_{\Theta_1}(\theta_1) f_{\Theta_2}(\theta_2) \quad (12)$$

In Eq. (12), the prior distributions of the model parameters Θ_1 and Θ_2 are given by $f_{\Theta_1}(\theta_1)$ and $f_{\Theta_2}(\theta_2)$, respectively. The choice of the prior distribution reflects the subjective knowledge of uncertainty regarding the variable before any testing. It is assumed that the prior distribution is able to explain the data with some degree of uncertainty; in other words, there exists a nonempty set E such that $\forall \{\Theta_1, \Theta_2\} \in E$, the prior PDF ($f_{\Theta_1}(\theta_1) f_{\Theta_2}(\theta_2)$) and likelihood ($L(\theta_1, \theta_2)$) values evaluated at $\{\Theta_1, \Theta_2\}$ are both non-zero.

The joint posterior density of the parameters is given by $f_{\Theta_1,\Theta_2}(\theta_1, \theta_2|D_{Y_1}, D_{Y_2})$. The likelihood function $L(\theta_1, \theta_2)$ is calculated as the probability of observing the given data (D_{Y_1}, D_{Y_2}), conditioned on the parameters being updated, i.e. Θ_1 and Θ_2. The likelihood function accounts for the uncertainty in the inputs X_1 and X_2. For details of the likelihood function, refer to Edwards (1984), Pawitan (2001), and Sankararaman (2012). As explained earlier in Sect. 3, Eq. (12) can be evaluated by generating samples of model parameters (Θ_1 and Θ_2) through MCMC sampling.

6 Test Resource Allocation

Typically, a multi-level, multi-physics system has several parameters that influence the overall system-level output, and the uncertainty in these parameters can be updated by tests at multiple levels of the system and multiple types of physics

coupling. When the posterior distributions of the parameters are propagated through the system model to calculate the overall system-level output, the posterior variance of the overall system-level prediction can be computed. With more acquisition of data, a decreasing trend can be observed in the variance of the system-level output.

Two types of questions need to be answered: (1) What type of tests to do (which component, isolated physics, etc.)? and (2) How many repetitions of each type? Each type of test has a different testing cost and an associated reduction in the variance of system-level prediction. Further, the same type of test may need to be repeated on nominally identical specimens of the same component or subsystem. Such repetition is performed in order to account for the effect of natural variability across nominally identical specimens; while each repetition may have the same monetary cost, the associated reduction in the variance of system-level prediction may be different.

The test conducted on one subsystem is assumed to be statistically independent of another test on another subsystem; in other words, one type of test is independent of any other type. Further, for a given type of test, the repetitions across multiple replicas are also considered to be independent. It is assumed that a model is available to predict the quantity being measured in each type of test; the model may have several outputs but only that output which is measured is of concern. The overall objective is to identify how many tests of each type must be performed so as to achieve the required reduction in the variance of the system-level output. If there are several system-level outputs, either an aggregate measure or the most critical output can be considered. However, multi-objective optimization formulations to simultaneously reduce the variance of more than one system-level output have not yet been addressed in the literature.

6.1 Sensitivity Analysis

The method of sensitivity analysis has been used to quantify the sensitivity of model output to parameters. While derivative-based methods only compute local sensitivities, the method of global sensitivity analysis (Saltelli et al. 2008) to apportion the variance in the system-level output to the various sources of uncertainty, and thereby guide in the reduction of system-level prediction uncertainty.

The first step of the resource allocation methodology is to use sensitivity analysis and identify those parameters that have a significant influence on the variance of the overall system-level prediction. Once the "important" parameters are identified, only those tests that aid in reducing the uncertainty in these important parameters can be performed. For example, consider a system-level output that is very sensitive to the uncertainty in the parameters of sub-system-I but not sensitive to the parameters of sub-system-II, then it is logical to perform more sub-system-I tests than sub-system-II tests. Note that this procedure for test identification is only a preliminary approach. This approach can answer the question—"which tests to do?" In order to answer the question, "how many tests to do?", it is necessary to quantify

the decrease in variance that may be caused due to a particular test. The effect of a particular test on variance reduction can be quantified by using Bayesian updating. Therefore, the resource allocation methodology first uses sensitivity analysis for selection of calibration parameters and then uses Bayesian updating to quantify the effect of a test on the variance of system-level prediction.

6.2 Optimization Formulation

In order to solve the resource allocation problem and identify the number of tests to be performed for each type, the optimization problem can be formulated in two ways, as explained below.

In the first formulation shown in Eq. (13), the goal is to minimize the variance of the system-level output subject to satisfying a budget constraint.

$$\begin{aligned} &\underset{N_{\text{test}}}{\text{Minimize}}\ E(\text{Var}(R)) \\ &\text{s.t.}\ \sum_{i=1}^{q}(C_i N_i) \leq \text{Total Budget} \\ &N_{\text{test}} = [N_1, N_2 \ldots N_q] \end{aligned} \tag{13}$$

In Eq. (13), q refers to the number of different types of possible tests. The cost of the ith ($i = 1 \text{to} q$) type of test is equal to C_i, and N_i (decision variable) denotes the number of repetitions of the ith type of test. Let D_i denote all the data collected through the ith type of test. Let N_{test} denote the vector of all N_i's and let $\boldsymbol{D}$ denote the entire set of data collected from all q types of tests.

Alternatively, the resource allocation problem can be formulated by minimizing the cost required to decrease the variance of the system-level output below a threshold level, as:

$$\begin{aligned} &\underset{N_{\text{test}}}{\text{Minimize}} \sum_{i=1}^{q}(C_i N_i) \\ &\text{s.t.}\ E(\text{Var}(R)) \leq \text{Threshold Variance} \\ &N_{\text{test}} = [N_1, N_2 \ldots N_q] \end{aligned} \tag{14}$$

Sankararaman et al. (2013) pursued the first formulation (Eq. (13)) because the threshold level for the variance is assumed to be unknown. Using $\boldsymbol{D}$, the model parameters are calibrated and the system-level response ($R(\boldsymbol{D})$) is computed. The optimization in Eq. (13) calculates the optimal values of N_i, given the cost values C_i, such that the expected value of variance of the system-level prediction ($E(\text{Var}(R))$) is minimized, while the budget constraint is satisfied.

This optimization formulation uses $E(\text{Var}(R))$ as the objective function because R is a function of $\boldsymbol{D}$, which is not available before testing. Hence, random realizations of the test data set ($\boldsymbol{D}$) are generated; each random realization is used to compute $\text{Var}(R|\boldsymbol{D})$, and the expectation over such random realizations is calculated to be the objective function, as:

$$E(\text{Var}(R)) = \int \text{Var}(R|\boldsymbol{D}) f(\boldsymbol{D}) d\boldsymbol{D} \tag{15}$$

where $f(\boldsymbol{D})$ is the density considered for the test data. Assuming that one type of test is performed independent of the another (i.e., a subsystem-level test is independent of a material-level test), Eq. (15) can be written as:

$$E(\text{Var}(R)) = \int \text{Var}(R|D_1, D_2 \ldots D_q) f(D_1) f(D_2) \ldots f(D_q) dD_1 dD_2 \ldots dD_q \tag{16}$$

where $f(D_i)$ is the density considered for the data obtained through the ith test. Before any testing is done, all prior knowledge regarding the model parameters and the mathematical models constitute the only information available for the calculation of $f(D_i)$. Therefore, $f(D_i)$ is calculated as:

$$f(D_i) = \int f(y_i|\boldsymbol{\theta}_i) f'(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i \tag{17}$$

where y_i represents the output of the mathematical model corresponding to the ith type of test, $\boldsymbol{\theta}_i$ represents the underlying parameters, and $f'(\boldsymbol{\theta}_i)$ represents the prior knowledge regarding those parameters. Note that Eq. (17) is simply an uncertainty propagation problem, where the other sources of uncertainty (such as physical variability in inputs, solution approximation errors, data uncertainty) can also be included in the computation of $f(y_i|\boldsymbol{\theta}_i)$.

Equations (15)–(17) are implemented using a numerical algorithm, where a finite number of realizations of $\boldsymbol{D}$ are generated and $E(\text{Var}(R))$ is computed over these realizations. Then, $E(\text{Var}(R))$ can be minimized using the optimization in Eq. (13), and the ideal combination of tests can be identified.

Note that an inequality constraint (for the budget), and not an equality constraint, is considered in Eq. (13). This means that the optimal solution which minimizes $E(\text{Var}(R))$ need not necessarily exhaust the budget. Consider the simple case where there are two possible test types ($C_1 = 2$ and $C_2 = 3$), and the budget is equal to six cost units. There are two test combinations which exhaust the budget: (1) [$N_1 = 3$, $N_2 = 0$], and (2) [$N_1 = 0$, $N_2 = 2$]. Suppose that these two combinations lead to a value of $E(\text{Var}(R))$ which is *greater* than that achieved through the test combination [$N_1 = 1$, $N_2 = 1$]. Then, obviously the combination [$N_1 = 1$, $N_2 = 1$] must be selected because it achieves the goal of reducing $E(\text{Var}(R))$ even though it may not exhaust the budget.

6.3 Solution of the Optimization Problem

Equation (13) is a complicated integer optimization problem, where Bayesian updating and forward propagation need to be repeated for each random realization of the test data in order to evaluate the objective function, thus increasing the computational cost several fold. In spite of the use of Gaussian process surrogate models to replace the expensive system model, high computing power is still needed to solve the optimization problem.

Integer optimization is sometimes solved using an approximation method, where the integer constraint is first relaxed, and the integers nearest to the resulting optimal solution are used in further solution of the original (un-relaxed) problem. Unfortunately, this approach is not applicable to the solution of Eq. (13), since the objective function (system-level prediction variance) is defined and computed only for integer-valued decision variables (number of tests). It is meaningless to have a non-integer number of tests.

A multi-step procedure for solving the optimization problem was proposed by Sankararaman et al. (2013). Within each step, the global optimal solution is computed using an exhaustive search process, whereas across steps, a greedy algorithm is pursued. The step size is chosen in cost units, and additional steps are added until the budget constraint is satisfied.

Let the size of the first step be equal to ϕ^1 cost units; the globally optimal testing combination for this cost ($= \phi^1$) is denoted by N_{test}^1, and is calculated using exhaustive search, as:

$$\begin{aligned} &\underset{N_{\text{test}}^1}{\text{Minimize}}\ E(\text{Var}(R)) \\ &\text{s.t. } \textstyle\sum_{i=1}^{q}(C_i N_i^1) \le \phi^1 \\ &N_{\text{test}}^1 = [N_1^1, N_2^1 \ldots N_q^1] \end{aligned} \tag{18}$$

The optimization procedure in the second stage is dependent on the optimal solution from the first stage, i.e. N_{test}^1. In general, the optimization for the jth stage, given the solution in the previous stage (i.e., N_{test}^{j-1}), is performed for cost = ϕ^j. Note that $\sum_j \phi^j$ = Total budget. The jth optimization is formulated as:

$$\begin{aligned} &\underset{N_{\text{test}}^{j,\text{new}}}{\text{Minimize}}\ E(\text{Var}(R)) \\ &\text{s.t. } \textstyle\sum(C_i N_i^{j,\text{new}}) \le \phi^j\,(i = 1\text{to}q) \\ &N_{\text{test}}^{j} = N_{\text{test}}^{j-1} + N_{\text{test}}^{j,\text{new}} \\ &N_{\text{test}}^{j,\text{new}} = [N_1^{j,\text{new}}, N_2^{j,\text{new}} \ldots N_q^{j,\text{new}}] \end{aligned} \tag{19}$$

As seen in Eq. (19), the decision variables for the jth stage are $N_{\text{test}}^{j,\text{new}}$, i.e. those tests which need to be performed in the jth stage; therefore the total number of

tests is equal to the sum of $N_{\text{test}}^{j,\text{new}}$ and N_{test}^{j-1}, i.e. the optimal number of tests in the previous stage. The same procedure is repeated until no additional test can be performed with the budget constraint satisfied.

The selection of step size for a given budget is an important issue. The true global optimal solution can be calculated by considering *one step* whose size is equal to the entire budget. However, due to the large number of possible testing combinations, this approach may be computationally infeasible. In a practical problem, several steps are considered, and the step sizes must be chosen judiciously based on (1) the costs of each type of test; (2) time required for each Bayesian update; (3) number of random realizations of data needed to compute $E(\text{Var}(R))$; and (4) the test combinations that are suitable for the chosen step size; a very small step size may not even include an expensive type of test.

6.4 Summary of the Optimization Methodology

The various steps of the optimization-based methodology for test resource allocation are summarized below:

1. **Construction of the Bayesian network:** The first step is to identify the various component-level, subsystem-level, and system-level models. Each model has an output quantity and correspondingly, a test can be performed to measure this quantity. All the models are connected through the Bayesian network, and the data available across the nodes is also indicated. The model errors, if available, can also be included in the Bayesian network. Though solution approximation errors can be calculated prior to testing and included in the Bayes network, model form error cannot be calculated before testing. It must be noted that the Bayesian network, due to its acyclic nature, cannot account for feedback coupling between models. When the system-level response is a coupled physics-based solution, the overall coupled solution is directly included in the Bayesian network instead of considering the individual physics separately.
2. **Sensitivity analysis:** The next step is to perform global sensitivity analysis and identify the "important" parameters that significantly contribute to the uncertainty in the system-level response. Then, those tests which can aid in the reduction of uncertainty in these "important" parameters are selected for consideration in the optimization for test resource allocation.
3. **Bayesian updating:** The next step is to perform Bayesian updating and calibrate parameters for a particular realization of measurement data. Then, this needs to be repeated by generating multiple realizations of measurement data in order to compute the expected value of variance, as in Eq. (15). Due to the required computational expense, the original physics models can be replaced with Gaussian process surrogates. Though this does not lead to analytical calculation of the posterior, it increases the computational efficiency several fold.

4. **Resource allocation optimization:** The final step is to perform the resource allocation optimization using the multi-step procedure developed in Sects. 6.2 and 6.3. It may be useful to verify that the resultant solution is actually optimal by computing $E(\text{Var}(R))$ for few other N_{test} values.

The rest of this chapter illustrates the optimization-based test resource allocation methodology using three different numerical examples. While Sect. 7 deals with a simple mathematical example, Sects. 8 and 9 consider multi-physics and multi-level engineering systems, respectively.

7 Illustration Using Mathematical Example

This subsection presents a simple illustrative example to illustrate the optimization-based methodology for test resource allocation. In order to focus on this objective, simple mathematical relationships are chosen (even the system-level response has no coupling), and measurement errors are assumed to be negligible. Other features such as coupled system response, measurements errors, and solution approximation errors (while replacing the underlying physics-based model with a Gaussian process approximation) are considered later in Sects. 9 and 8.

The Bayesian network for this problem is exactly the same as that in Fig. 1. There are four independent quantities and three dependent quantities; the numerical details of this problem are specified in Table 1. The notation $N(\mu, \sigma)$ is used to represent a normally distributed quantity with mean μ and standard deviation σ. Two types of tests (on two different lower levels) can be done and this information is used to update the uncertainty in the system-level response based on the tests.

Probability distributions are assumed to be available for the inputs X_1 and X_2; if this information was not available, and only sparse and/or interval data was available for the inputs, then the likelihood-based method developed in Sankararaman and Mahadevan (2011) can be used to construct a probability distributions for them.

Table 1 Numerical details

Quantity	Type	Description
X_1 (input)	Independent	N(100,5)
Θ_1 (parameter)	Independent	N(50, 10)
X_2 (input)	Independent	N(10,1)
Θ_2 (parameter)	Independent	N(15, 4)
Y_1	Dependent	Model : $y_1 = x_1 + x_2$
Y_2	Dependent	Model : $y_2 = x_3 + x_4$
Z	System-level response	Model : $z = y_1 - y_2$
Quantity to measure	Cost	No. of tests
Y_1	10	N_1
Y_2	5	N_2

Table 2 Resource allocation: results

Cumulative cost	N_1	N_2	$E(\text{Var}(z))$
\$10	**1**	**0**	**62.0**
	0	2	127.0
\$20	2	0	53.0
	1	**2**	**46.6**
\$30	**2**	**2**	**37.6**
	1	4	46.1
\$40	**3**	**2**	**34.0**
	2	4	37.6
\$50	**4**	**2**	**32.5**
	3	4	33.8

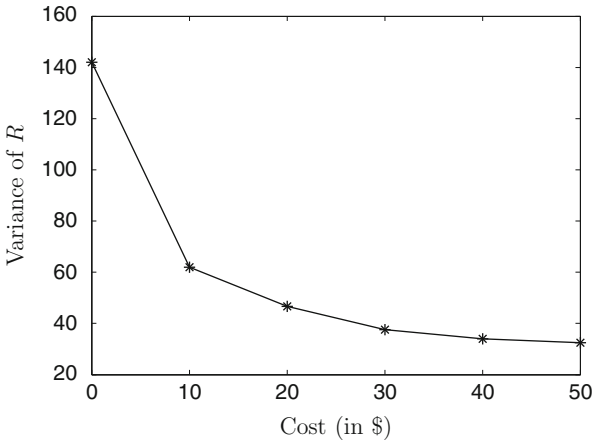


Fig. 2 Variance vs. cost

The variance of the system-level response quantity Z before conducting any test (i.e., by propagating the above distributions of X_1, X_2, Θ_1, and Θ_2 through the models) is 142 units. The objective is calculate the number of tests on Y_1 and Y_2 (N_1 and N_2), that will lead to a minimum variance in Z, subject to a total budget of \$50 cost units. Since there are only two parameters, global sensitivity analysis is not necessary, and hence, both Θ_1 and Θ_2 are chosen for calibration. The optimization-based methodology discussed in Sect. 6 is used for this purpose; five different stages are considered and the available budget in each stage is considered to be \$10. The results of test prioritization are given in Table 2 (the optimal value in each stage is indicated in bold) and Fig. 2.

At the end of the optimization procedure, the optimal combination is found to be four tests on Y_1 and 2 tests on Y_2. Further, this solution was verified by considering all other combinations (exhaustive search) of N_1 and N_2 and computing the corresponding $E(\text{Var}(R))$; for this illustrative example, this verification is numerically affordable. However, for practical examples, a few random values of $N_{\text{test}} = [N_1, N_2]$ (if not all) can be considered and it can be verified if the estimated solution is really optimal.

8 Numerical Example: Multi-Physics System

8.1 Description of the Problem

This coupled-physics thermal vibration example illustrates a laboratory experiment which can be used to study and simulate the behavior in solar arrays of telescopes and spacecraft booms (Thornton 1996). In this experiment, a thin walled circular tube is rigidly attached at its top and supports a concentrated mass at its bottom. The tube and the mass are initially at rest and a constant heat flux is applied on one side along the length of the tube. The application of the heat flux causes an increase in the temperature on the incident surface while the unheated side remains at the initial temperature. The temperature gradient causes the beam to bend away from the lamp, due to the thermal moment. The displacement of the beam, in turn, changes the distribution of temperature along the length of the beam, leading to a change in the temperature gradient and the thermal moment, which in turn affects the flexural behavior. Thus the combination of heat transfer and flexural mechanics leads to oscillations of the beam. The setup of this experiment is shown in Fig. 3.

The temperature at the tip mass (T_m) is given by the following differential equation:

$$\frac{\partial T_m}{\partial t} + \frac{T_m}{\tau} = \frac{T^*}{\tau}\left(1 - \frac{v(x,t)}{\beta^*}\right) \tag{20}$$

In Eq. (20), $v(x,t)$ represents the displacement of the beam as a function of length and time. Thornton (1996) explains how to calculate the parameters T^*, τ, β^* as a function of the incident solar flux (S).

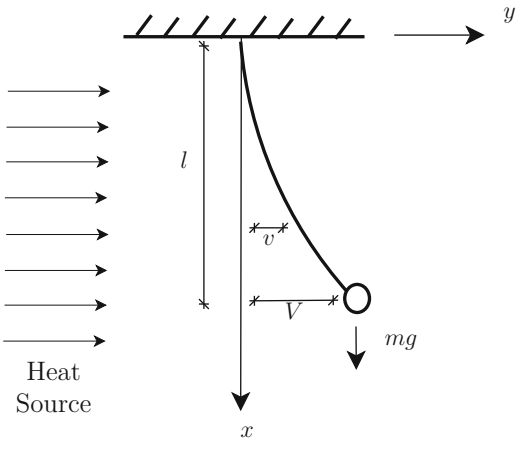


Fig. 3 Thermally induced vibration

The displacement $v(x,t)$ can be related to the displacement of the tip mass $V(t)$ as:

$$v(x,t) = \left(\frac{3x^2}{2l^2} - \frac{x^3}{2l^3}\right)V(t) \tag{21}$$

The tip mass displacement $V(t)$, in turn, depends on the forcing function as follows:

$$\ddot{V} + 2\xi\omega_0\dot{V} + \left(\omega_0^2 + \frac{6g}{5l}\right)V = \frac{F(t)}{m} \tag{22}$$

In Eq. (5), ξ is the damping ratio, and ω_0 is the angular frequency. The forcing function $F(t)$ depends on the thermal moment which in turn depends on the temperature, thereby casing coupling between the thermal equation and the structural equation. These relations are shown in the following equations:

$$F(t) = -\frac{3}{l^3}\int_0^l \int_0^x M(u,t)dudx \tag{23}$$

$$M(x,t) = \int E\alpha T_m(x,t)cos(\Phi)ydA \tag{24}$$

In Eq. (24), E is the elastic modulus, α is the coefficient of thermal expansion, Φ is the angle of incident flux on the cross section, y is the distance from the center of the cross section and the integral is over the area of the cross section A. Refer Thornton (1996) for a detailed description of this problem.

The overall objective of test resource allocation is to minimize the variance of the system-level output (R), which is defined to be the ratio of displacement amplitudes at two different time instants for the coupled system when the incident solar flux (S) is 2,000 W/m^2. If $R < 1$, the system is stable with oscillations diminishing as a function of time. If $R > 1$, the system is unstable, commonly referred to as flutter, an undesirable scenario. While a Gaussian process model is constructed to calculate the multi-physics response R, individual physics predictions are performed using the above described physics-based models.

There are several parameters (both thermal and structural) in the above equations that can be calibrated using test data. The method of sensitivity analysis is used to identify five parameters, which significantly contribute to the uncertainty in the system-level prediction. The prior means are based on Thornton (1996), and the assumed coefficients of variation (CoV) are tabulated in Table 3; note that the radius being a geometric property has a lower CoV. The calibrated parameters are then used to quantify the uncertainty in R.

The calibration parameters need to be estimated during test data; four different types of tests are considered, as shown in Table 4. The total budget available for testing is assumed to be $2,000. It is assumed that the entire multi-disciplinary system cannot be tested.

Table 3 Calibration quantities: thermal vibration problem

Symbol	Quantity	Property	Prior CoV
E	Elastic modulus	Structural	0.1
c	Independent	Thermal	0.1
ξ	Independent	Structural	0.1
r	Independent	Geometric	0.03
e	Dependent	Thermal	0.1

Table 4 Types of tests: thermal vibration problem

Test type	Physics	Calibrate	Input–Output	Cost	No. of tests
Material-level	Thermal	c	Heat-Temperature rise	\$100	N_{m_1}
Material-level	Structural	ξ	Amplitude decay	\$100	N_{m_2}
Subsystem-level	Thermal	c, e, r	Heat-Temperature rise	\$500	N_T
Subsystem-level	Structural	ξ, E	Acceleration	\$500	N_F

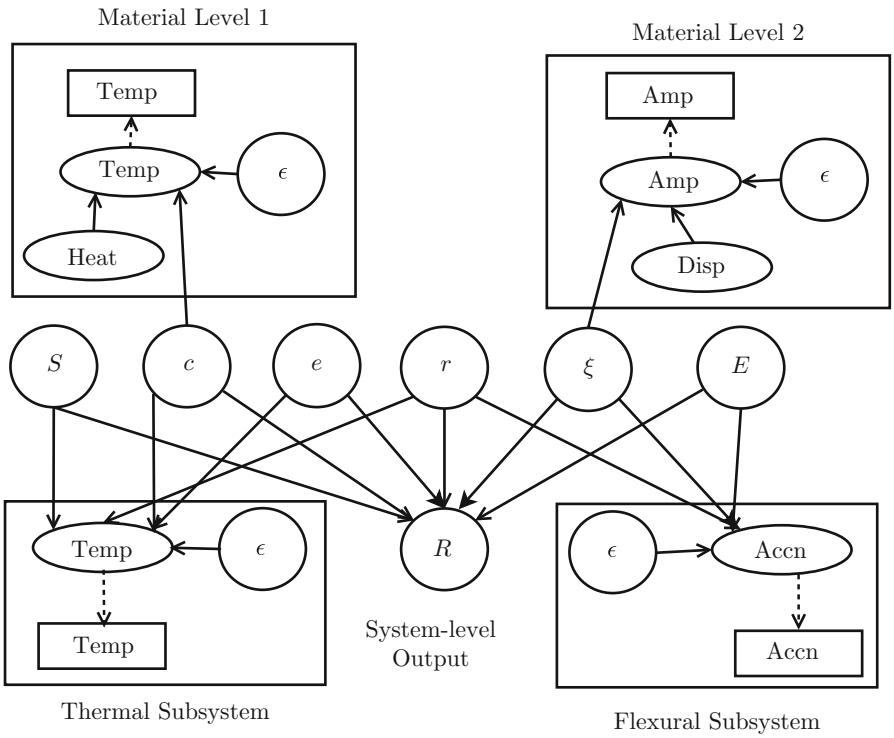


Fig. 4 Thermal vibration: Bayesian network

The calibration quantities, the model predictions, and the test data are connected through a Bayesian network, as shown in Fig. 4.

In the Bayesian network in Fig. 4, "Temp" refers to temperature, "Accn" refers to the acceleration, "Disp" refers to the displacement, and "Amp" refers to the

Table 5 Resource allocation results: thermal vibration problem

Stage no.	N_{m_1}	N_{m_2}	N_F	N_T	$E(\text{Var}(R))$ (in %)
No tests	0	0	0	0	100.0
Stage 1 : \$500	1	4	0	0	74.6
Stage 2 : \$1,000	1	4	1	0	51.4
Stage 3 : \$1,500	1	4	1	1	44.8
Stage 4 : \$2,000	1	9	1	1	44.2

amplitude of vibration. Measurement errors (ϵ) are assumed to have a standard deviation that is equal to 10 % of the model prediction. This Bayesian network is used for uncertainty quantification, Bayesian updating and resource allocation.

8.2 Resource Allocation

The objective is to calculate the number of tests that lead to maximum reduction in variance in R. Let N_{test} denote the number of tests, where $N_{\text{test}} = [N_{m_1}, N_{m_2}, N_F, N_T]$; where N_{m_1} is the number of material level temperature tests, N_{m_2} is the number of material level pluck tests, N_F is the number of flexural subsystem tests, and N_T is the number of thermal subsystem tests. Let $D = [D_{m_1}, D_{m_2}, D_F, D_T]$ denote the test measurements. The optimization problem for resource allocation can be formulated as shown in Eq. (25)

$$\begin{aligned} &\underset{N_{\text{test}}}{\text{Minimize}}\ E(\text{Var}(R)) \\ &\text{s.t. } 100(N_{m_1} + N_{m_2}) + 500(N_F + N_T) \leq 2000 \\ &N_{\text{test}} = [N_{m_1}, N_{m_2}, N_F, N_T] \end{aligned} \tag{25}$$

The above optimization is solved using the multi-stage optimization procedure discussed in Sects. 6.2 and 6.3. Four stages and a budget of \$500 for each stage are considered, thereby accounting for the total budget of \$2,000. Each stage has eight options (as against two in the mathematical example in Sect. 7); only the optimal solution in each stage is shown.

Note that Table 5 expresses the expectation of variance of R in terms of percentage of the variance before any testing; this variance is equal to 5.69×10^{-7}; since R is a ratio, this variance is dimensionless.

For a \$2,000 budget, it is seen that one temperature test, nine pluck tests, one thermal subsystem test and one flexural subsystem test are required to achieve the maximum reduction in the variance of R. The results show that while it is useful to do all the tests, repeating the pluck test which calibrates structural damping, is not only cheap but also leads to effective decrease in the variance of R. The decrease of variance with cost is shown in Fig. 5.

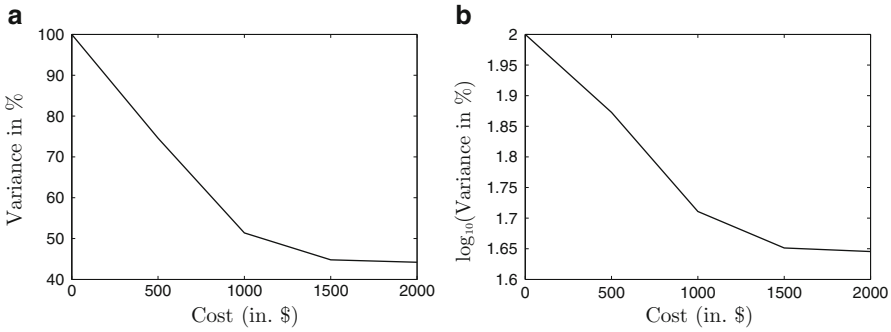


Fig. 5 Decrease of variance with cost. (**a**) Variance with cost; (**b**) log(Variance) with cost

It is seen that the reduction in variance using the last $1,000 (i.e., from $1,000 to $2,000) was much smaller when compared to the reduction in variance using the initial $1,000. Such information is very useful for budgeting purposes, since all the above computation (and practical resource allocation) is done before any test is actually conducted.

9 Numerical Example: Multi-Level System

9.1 Description of the Problem

A three-level structural dynamics developed at Sandia National Laboratories (Red-Horse and Paez 2008) is considered as shown in Fig. 6.

The first level (component) consists of a single spring-mass-damper. Three such spring-mass dampers are integrated to form a spring-mass-damper subsystem in the second-level. In the third level, the integrated spring-mass-damper subsystem is mounted on a beam to form the overall system.

The models to represent the first two levels are straightforward (Chopra 1995). Red-Horse and Paez (2008) describe in detail the modeling and simulation of the overall system (third-level). The overall objective is to test resource allocation to minimize the variance of the system level output (R) which is defined to be the maximum acceleration of mass m_3, when a random force is applied as specified in Red-Horse and Paez (2008). The first-level and second-level responses are computed using physics-based models while the third-level and system-level responses are computed by constructing two Gaussian process surrogate models.

In this numerical example, the stiffness values of the three masses, i.e. k_1, k_2, and k_3 are all the parameters that need to be calibrated with test data; since all parameters are calibrated, sensitivity analysis is not used in this example. The numerical values (in SI units) of three calibration parameters are summarized in Table 6.

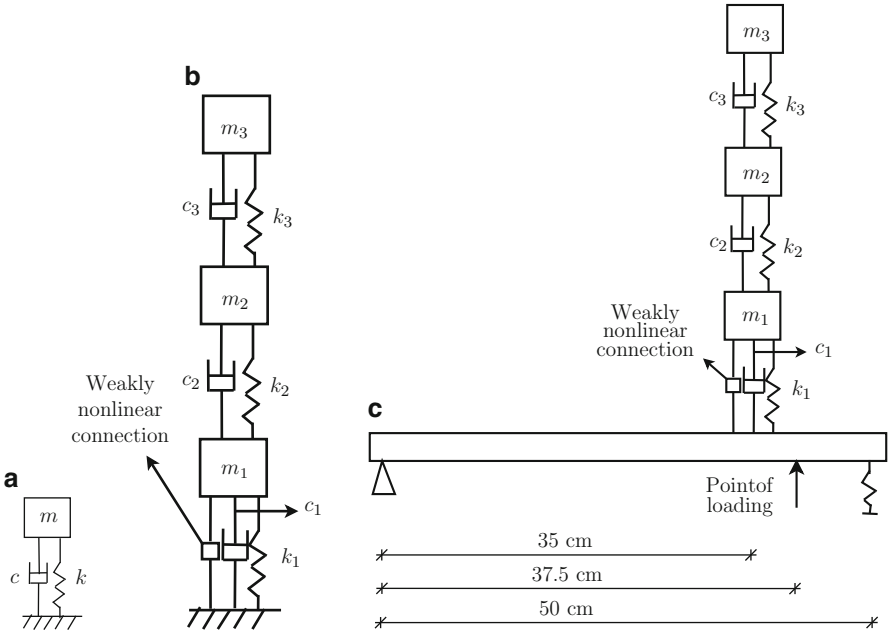


Fig. 6 Multi-level structural dynamics problem. (**a**) Level 1; (**b**) level 2; (**c**) level 3

Table 6 Model parameters: structural dynamics problem

Number	Mass (m) (in kg)	Damping (c) (in Ns/m)	Prior mean of stiffness (μ_k) (in N/m)	Prior std. dev. of mean (σ_k) (in N/m)
1	0.012529	0.023466	5,600	560
2	0.019304	0.021204	11,000	1,100
3	0.035176	0.031216	93,000	9,300

The mass of the beam is taken to be 0.1295. Further numerical details of the beam are given in Red-Horse and Paez (2008).

Data for calibration is assumed to be available through five different types of tests. The details of these different types of tests are provided in Table 7. For each test, a sinusoidal load (amplitude = 10,000 and angular velocity = 10 rad s^{-1}) is used. For the first and second level tests, the sinusoidal load is applied at the base; for the third level test, the sinusoidal load is applied as specified in Red-Horse and Paez (2008).

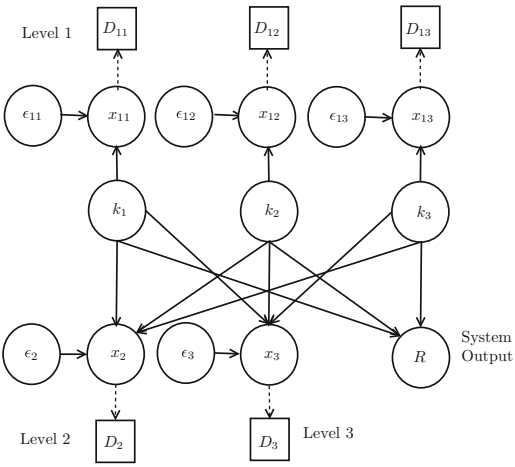
The model predictions, experimental data, and the calibration quantities are connected using the Bayesian network, shown in Fig. 7. The corresponding experimental errors are denoted by ϵ_{11}, ϵ_{12}, ϵ_{13}, ϵ_2, and ϵ_3, respectively, and assumed to be equal to 10 % of the prediction.

This Bayesian network can be used for uncertainty quantification, Bayesian updating, and resource allocation, as explained below.

Table 7 Model parameters: structural dynamics problem

Test type	Description	Model prediction	Data	Cost	No. tests
Level-1	Only mass m_1	Acceleration (x_{11})	D_{11}	\$100	N_{m_1}
Level-1	Only mass m_1	Acceleration (x_{12})	D_{12}	\$100	N_{m_2}
Level-1	Only mass m_1	Acceleration (x_{13})	D_{13}	\$100	N_{m_3}
Level-2	3-mass assembly	Acceleration of m_3 (x_2)	D_2	\$500	N_2
Level-3	3-mass assembly on beam	Acceleration of m_3 (x_3)	D_3	\$1,000	N_3

Fig. 7 Bayesian network



9.2 Resource Allocation

In the resource allocation problem, testing is yet to be done and hence realizations of future experimental data are generated randomly. Then, $E(\text{Var}(R))$ is computed so as to identify which set of tests will lead to the maximum reduction in variance. Let $N_{\text{test}} = [N_{m_1}, N_{m_2}, N_{m_3}, N_2, N_3]$. The optimization problem for resource allocation can be formulated as shown in Eq. (26).

$$\begin{aligned} &\underset{N_{\text{test}}}{\text{Minimize}}\ E(\text{Var}(R)) \\ &\text{s.t. } 100(N_{m_1} + N_{m_2} + N_{m_3}) + 500N_2 + 1000N_3 \leq 1000 \\ &N_{\text{test}} = [N_{m_1}, N_{m_2}, N_{m_3}, N_2, N_3] \end{aligned} \tag{26}$$

First, the resource allocation is solved for a budget of \$1000. There are 54 possible testing combinations and out of these 54, ten testing combinations lead to the same minimum variance of system-level output R, approximately 0.8 % of the variance before testing. These combinations are given in Table 8. The value of $E(\text{Var}(R))$ for these ten cases are close enough that it is not possible to determine whether the difference is due to reality or due to sampling/numerical errors.

Table 8 Resource allocation results: structural dynamics problem

N_{m_1}	N_{m_2}	N_{m_3}	N_2	N_3
0	1	4	1	0
0	4	1	1	0
0	3	2	1	0
0	2	3	1	0
1	1	3	1	0
3	1	1	1	0
1	3	1	1	0
1	2	2	1	0
2	1	2	1	0
2	2	1	1	0

It is a subjective decision as to which one of these ten test combinations is selected. However, all ten combinations unanimously suggest that no tests are needed for the overall system and one test is needed for the second level three spring-mass-damper subsystem. The first four rows in Table 8 suggest that testing is not needed for the first spring-mass-damper. However, it may be desirable to have at least one test for each component, and hence one amongst the latter six options may be preferred.

It was also found that an extra budget of $1,000 caused no further reduction in the variance of R. If the available budget is $2,000, a subjective decision may be made to conduct the full system test (which costs $1,000) in order to further improve the confidence in uncertainty quantification.

10 Conclusion

This chapter presented a Bayesian methodology for uncertainty quantification in complex engineering systems consisting of multiple physics behavior and multiple levels of integration. The various component, subsystem, and system models, and their inputs, parameters, and outputs, and experimental data were efficiently connected through a Bayesian network. Further, the various sources of uncertainty—physical variability, data uncertainty, and model uncertainty were also included in the Bayesian network. The Bayesian network was used for three different tasks: (1) calibrate the parameters of models at multiple levels using all available test data from multiple levels; (2) propagate the various sources of uncertainty (including the previously estimated model parameters) through the Bayes network to predict the overall uncertainty in the system-level response; and (3) aid in resource allocation for test selection, in order to identify the most effective tests to reduce the overall uncertainty in the system-level prediction. The procedure for test resource allocation required Bayesian calibration and assessment of system-level prediction uncertainty even before actual testing was performed. This was achieved by generating multiple samples of test data and estimating the expected reduction in variance of the system-level prediction.

The algorithm for test resource allocation leads to several insights. A lower level test can easily isolate individual components and hence, the model parameters can be effectively updated, leading to a significant reduction in the variance of the system-level prediction. However, such a test would not account for the interaction between the higher level models and the corresponding parameters. In contrast, a higher level test would readily include the effects of interaction between multiple subsystem-level and component-level models. However, the calibration of parameters across multiple models may be difficult and may not lead to a significant reduction in the variance of the system-level prediction. The optimization-based test resource allocation procedure trades off between lower level tests and higher level tests by accounting not only for the resultant reduction in variance of the system-level prediction but also for the costs involved in testing.

References

Aldrich J (1997) R.A. Fisher and the making of maximum likelihood 1912–1922. Stat Sci 12(3):162–176

Aldrich J (2008) R.A. Fisher on Bayes and Bayes' theorem. Bayesian Anal 3(1):161–170

Bayes T, Price M (1763) An essay towards solving a problem in the doctrine of chances, by the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS. Philos Trans (1683–1775) 53:370–418

Bichon BJ, Eldred MS, Swile LP, Mahadevan S, McFarland JM (2008) Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J 46(10):2459–2468

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory. ACM, New York, pp 144–152

Calvetti D, Somersalo E (2007) Introduction to Bayesian scientific computing: ten lectures on subjective computing, vol 2. Springer, New York

Chiles JP, Delfiner P (1999) Geostatistics: modeling spatial uncertainty, vol 344. Wiley-Interscience, New York

Chopra AK (1995) Dynamics of structures. Prentice Hall, Englewood

Cressie N (1991) Spatial statistics. Wiley, New York

Edwards AWF (1984) Likelihood. Cambridge University Press, New York

Fienberg SE (2006) When did Bayesian inference become Bayesian? Bayesian Anal 1(1):1–40

Fisher RA (1912) On an absolute criterion for fitting frequency curves. Messenger Math 41(1):155–160

Geman S, Geman D (1984) Stochastic relaxation, gibbs distributions and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6:721–741

Ghanem R, Spanos PD (1990) Polynomial chaos in stochastic finite elements. J Appl Mech 57(1):197–202

Haldar A, Mahadevan S (2000) Probability, reliability, and statistical methods in engineering design. Wiley, New York

Hastings WK (1970) Monte carlo sampling methods using markov chains and their applications. Biometrika 57(1):97

Heckerman D (2008) A tutorial on learning with Bayesian networks. In: Innovations in Bayesian networks. Springer, Berlin, pp 33–82

Helton JC, Pilch M (2011) Quantification of margins and uncertainties. Reliab Eng Syst Saf 96(9):959–964

Hombal V, Mahadevan S (2011) Bias minimization in Gaussian process surrogate modeling for uncertainty quantification. Int J Uncertain Quantif 1(4):321–349
Jeffreys H (1998) Theory of probability. Oxford University Press, Oxford
Jensen FV (1996) An introduction to Bayesian networks, vol 210. UCL Press, London
Kennedy MC, O'Hagan A (2011) Bayesian calibration of computer models. J R Stat Soc Ser B Stat Methodol 63(3):425–464
Lee PM (2004) Bayesian statistics. Arnold, London
Leonard T, Hsu JSJ (2001) Bayesian methods. Cambridge Books, New York
Liu JS, Liang F, Wong WH (2000) The multiple-try method and local optimization in metropolis sampling. J Am Stat Assoc 95:121–134
McFarland JM (2008) Uncertainty analysis for computer simulations through validation and calibration. Ph.D. thesis, Vanderbilt University
Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21(6):1087
Neal RM (2003) Slice sampling. Ann Stat 31(3):705–741
Pawitan Y (2001) In all likelihood: statistical modelling and inference using likelihood. Oxford University Press, New York
Rajashekhar MR, Ellingwood BR (1993) A new look at the response surface approach for reliability analysis. Struct Saf 12(3):205–220
Rasmussen CE (1996) Evaluation of Gaussian processes and other methods for non-linear regression. Ph.D. thesis, University of Toronto
Rasmussen CE (2004) Gaussian processes in machine learning. In: Advanced lectures on machine learning. Springer, Berlin, pp 63–71
Red-Horse JR, Paez TL (2008) Sandia national laboratories validation workshop: structural dynamics application. Comput Methods Appl Mech Eng 197(29–32):2578–2584
Roberts GO, Rosenthal JS (2006) Harris recurrence of metropolis-within-gibbs and trans-dimensional markov chains. Ann Appl Probab 16(4):2123–2139
Rosenblatt M (1956) Remarks on some nonparametric estimates of a density function. Ann Math Stat 27:832–837
Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S (2008) Global sensitivity analysis: the primer. Wiley, Hoboken
Sankararaman S (2012) Uncertainty quantification and integration in engineering systems. Ph.D. thesis, Vanderbilt University
Sankararaman S, Mahadevan S (2011) Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data. Reliab Eng Syst Saf 96(7):814–824
Sankararaman S, McLemore K, Mahadevan S, Bradford SC, Peterson LD (2013) Test resource allocation in hierarchical systems using bayesian networks. AIAA J 51(3):537–550
Santner TJ, Williams BJ, Notz W (2003) The design and analysis of computer experiments. Springer, New York
Singpurwalla ND (2006) Reliability and risk: a Bayesian perspective, vol 637. Wiley, Hoboken
Singpurwalla ND (2007) Betting on residual life: the caveats of conditioning. Stat Probab Lett 77(12):1354–1361
Somersalo E, Kaipio J (2004) Statistical and computational inverse problems. Springer, New York
Stigler SM (1986) Laplace's 1774 memoir on inverse probability. Stat Sci 1(3):359–363
Stigler SM (2002) Statistics on the table: the history of statistical concepts and methods. Harvard University Press, Cambridge
Thornton EA (1996) Thermal structures for aerospace applications. AIAA J, Reston, VA
Tipping ME (2001) Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244
Wackernagel H (2003) Multivariate geostatistics: an introduction with applications. Springer, New York