Submitted to *Physica D*, September, 1998

A framework for assessing uncertainties in simulation predictions

Kenneth M. Hanson [1]

Los Alamos National Laboratory, MS P940, Los Alamos, New Mexico 87545, USA

Abstract

A probabilistic framework is presented for assessing the uncertainties in simulation predictions that arise from model parameters derived from uncertain measurements. A probabilistic network facilitates both conceptualizing and computationally implementing an analysis of a large number of experiments in terms of many intrinsic models in a logically consistent manner. This approach permits one to improve one's knowledge about the underlying models at every level of the hierarchy of validation experiments.

Keywords: uncertainty analysis, error analysis, simulation code, validation, probabilistic network, model checking, adjoint differentiation

1 Introduction

Simulation codes are everywhere. They are used to design things like automobiles, airplanes, bridges, chemical plants, and computers. Simulation codes are the basis for predicting the evolution of large-scale natural phenomena such as the weather, ocean currents, and climate. With the increasing reliance on simulation codes, it is becoming critically important to determine how well they predict actual physical phenomena. Uncertainty in simulation code predictions has many sources, including the lack of knowledge of the underlying physics models, the variability of the initial geometry and materials, and the degree of variability in the physical phenomenon itself. In this paper I will concentrate on the first issue by presenting a framework within which to handle uncertainties in the parameters associated with the physics models on which

[1] Work supported by U. S. Dept. of Energy under contract W-7405-ENG-36.
Telephone: 505-667-1402, Fax: 505-665-3359, E-mail: kmh@lanl.gov

a simulation code is based. The goal is to determine how these uncertainties affect our ability to predict the behavior of a physical system. This framework may accommodate the other degradations to predictability and may provide a worthwhile foundation for the *verification and validation* (V&V) of simulation codes [1,2]. Validation is the process of determining how well the physics equations describe physical reality and verification is the process of determining whether the simulation code accurately evaluates those equations.

I will focus on complex simulation codes that rely on a number of basic physics models, each of which specifies a specific physical behavior. For example, these physics models may model the equation of state, a stress-strain relationship, or strength characteristics of each material included in the physical system being simulated. A physics model may provide a set of dynamical equations that approximate more fundamental relationships. The focus here is on the uncertainties in these physics models and how they propagate into uncertainties in the predictions made by the simulation code that uses them. I will ignore calculational errors arising from coding mistakes or inadequacies of the resolution of the calculation. These fall under the category of code verification, which must go hand in hand with the overall validation effort [1,2].

The process of validation of a simulation code for an intended purpose should be based on a hierarchy of experiments. There should be experiments that are designed to characterize the basic physics models. Such experiments, which optimally require only one physics model to analyze, can be called *basic experiments*. Often, these do not even require the full simulation code for their analysis. Whereas basic experiments involve only a single physics model, *integrated experiments* require two or more physics models. Ideally the hierarchy of experiments should include some with various degrees of integration, up to fully integrated experiments that involve all relevant physics models. Clearly the experiments should span the range of physical conditions relevant to the target application.

This paper starts with a description of a general approach to analyzing individual experiments with an emphasis on uncertainty analysis. A Monte Carlo method is presented for propagating uncertainties in underlying physics models into uncertainties in simulation predictions. A probabilistic network is proposed for conceptualizing the process of analyzing a large number of experiments. It also provides the basis for a logically consistent, complete, and comprehensive implementation, as explained below.

While several ways to implement this approach are described, the best choice for the uncertainty assessment of any specific simulation code will depend on many factors, including the computational complexity of the simulation, the number of experiments, the number of parameters, and the complexity of the experimental analysis.

2 Analysis of individual experiments

Figure 1 schematically outlines the conceptual steps taken to analyze any experiment. The basic idea is that one tries to account for the measurements, represented by the vector $\mathbf{Y}$, obtained in an experiment by analyzing them in terms of a model of the experimental situation. The preparation of the physical system is shown as a time-dependent state $\Psi(t)$. This representation is perhaps more general than required for some experiments, but may be needed to analyze integrated experiments. The physical system believed to be observed is transformed into a set of predicted measurements $\mathbf{Y}^*$ by a model of the measurement system. This model should include all known effects in measurement process, for example, the effects of finite time or spatial resolution, sensitivity curve for the sensors used, etc. The predicted measurements $\mathbf{Y}^*$ are a function of the parameters imbedded in the physics model $\boldsymbol{\alpha}$, which are to be determined from the analysis. Figure 1 is meant to imply a point evaluation of the quantities shown, that is, the state of the system and the predicted measurements are evaluated for each parameter vector $\boldsymbol{\alpha}$ specified.

The typical approach taken to determine the model parameters $\boldsymbol{\alpha}$ is to find the parameters that minimize the mean square difference between $\mathbf{Y}^*$ and $\mathbf{Y}$, which, when normalized to the variance in the measurement uncertainty σ^2, is what physicists call *chi-squared*: $\chi^2 = \sum_i (Y_i - Y_i^*)^2/\sigma_i^2$. This general approach is referred to by physicists as fitting a model to data, and by statisticians as regression. The rationale for this approach and the accompanying uncertainty analysis is outlined in the next section.

In integrated experiments, where more than one physics model is required to simulate the experiment, the data-flow diagram in Fig. 1 still applies, except that the model box feeding the simulation is replaced by several model boxes.

The quality of experimental measurements is crucially important since they form the basis for inference about the physics models. All the usual care must be exercised in conducting these experiments. The experimentalist and analyst must pay particular attention to correlations between uncertainties. Correlations often exist between the uncertainties associated with different measurements. Furthermore, the process of analyzing a data set in terms of several parameters almost always results in correlations between the uncertainties in those parameters. It is crucially important to the final inference process that correlations be understood and included in each step of the analysis. A related issue is that of *systematic uncertainties*, which are uncertainties in an experiment that affect many (or all) measurements [3]. All means should be taken to reduce systematic uncertainties as much as possible and then to include them in the uncertainty analysis.

An underlying issue in solving the minimization problem implied by Fig. 1 is how to efficiently find the minimum, particularly when there are many parameters to be found and the simulation calculation takes a long time. Under the right circumstances, when there are many parameters to optimize, it is very beneficial to use gradient-based optimization methods. The gradient of the objective function with respect to the parameters of interest may be calculated in a time comparable to the forward simulation time using the technique of *adjoint differentiation* [4]. In principle, adjoint differentiation can be implemented for any computational code for which the outputs are differentiable with respect to the variables in the problem. Giering [5] has developed a useful compiler for forward codes written in FORTRAN to automatically create the code necessary to calculate the desired gradients [6].

Experience with several kinds of simulation codes [7–11] reinforces my conviction that adjoint differentiation is an enabling technique for problems of this sort. In this paper I will mention the areas where it might be of benefit. However, the overall framework presented here does not rely on using adjoint differentiation.

The data-flow diagram shown in Fig. 1 may be employed not only as a description of the analysis, but also as a basis for implementing the calculation. In the well-known image-processing application Khoros [12], the course of the calculation may be laid out using a graphically programmed data-flow diagram. In the Bayes Inference Engine (BIE) [7], which we have developed at Los Alamos for radiographic modeling, the forward modeling process is graphically specified by an analyst in terms of a diagram very similar to that shown in Fig. 1 and the BIE solves the inverse problem, i.e., finds the parameters that minimize χ^2, for example. This close connection between a graphical representation of a conceptual approach and the actual implementation of the solution will be suggested as a viable means to assess the uncertainties in simulation codes.

2.1 Uncertainty analysis

In the past the traditional process of analyzing the measurements from physics experiments and estimating the uncertainties in the derived model parameters has been called *error analysis* [13,14,3]. In common usage, the term 'error' often connotes an identifiable mistake, which is potentially correctable. It seems that this process might be better called *uncertainty analysis*. Furthermore, this phrase implies what we really intend, a statement of what we know (or do not know) about the parameters that we have tried to measure.

In a probabilistic approach to uncertainty analysis, uncertainties are expressed in terms of a probability density function (PDF) defined on the parameters [15–17]. As in the previous section, let's assume that an experiment is performed and the measurements, represented by the vector $\mathbf{Y}$, are obtained. A model $\mathcal{M}$ with associated parameters $\boldsymbol{\theta}$, is used to analyze the experimental results. (I use $\boldsymbol{\theta}$ here to represent generic parameters.) The PDF describing the uncertainty the parameters is $p(\boldsymbol{\theta}|\mathbf{Y},\mathcal{M})$. This so-called posterior is given by the fundamental rule of probability, Bayes law, as

$$p(\boldsymbol{\theta}|\mathbf{Y},\mathcal{M}) \propto p(\mathbf{Y}|\boldsymbol{\theta},\mathcal{M})\, p(\boldsymbol{\theta}|\mathcal{M}) \ , \tag{1}$$

where $p(\mathbf{Y}|\boldsymbol{\theta},\mathcal{M})$, is the likelihood of the measurements and $p(\boldsymbol{\theta}|\mathcal{M})$ is the prior on the parameters.

The *likelihood* comes from comparing the actual measurements to the measurements predicted on the basis of the model of the physical system. The predicted measurements are generated using a model for how the measurements are related to the physical system, which we call the measurement system model. Under the assumption that the measurements are stochastically degraded by uncorrelated and additive Gaussian noise, the likelihood is proportional to the exponential of $-\frac{1}{2}\chi^2$. The *prior* in (1) can supply knowledge coming from previous measurements, specific information regarding the object itself, or simply general knowledge about the parameters, e.g., that they are nonnegative.

In terms of this nomenclature, we see that the approach described in the previous section amounts to finding the parameters that maximize the posterior under the assumption of a Gaussian distribution for the measurement uncertainties and in the absence of prior information, i.e., a flat prior. The approach of estimating the parameters by maximizing the posterior is referred to as *maximum a posteriori* (MAP) estimation. Although not important for the present discussion, it should be noted that other types of estimators can be argued to be more appropriate in some circumstances.

It is often more convenient to work with logarithms of probabilities rather than with the probabilities themselves. Then Bayes law (1) becomes additive instead of multiplicative:

$$\varphi = -\log[\,p(\boldsymbol{\theta}|\mathbf{Y},\mathcal{M})] = -\log[\,p(\mathbf{Y}|\boldsymbol{\theta},\mathcal{M})] - \log[\,p(\boldsymbol{\theta}|\mathcal{M})] + C(\mathcal{M}) \ . \tag{2}$$

The condition for minimizing φ, to obtain the MAP parameter estimate, is that its gradient with respect to the parameters is zero: $\nabla_{\boldsymbol{\theta}}\, \varphi = 0$.

As the posterior describes our state of knowledge about the parameters, parameter uncertainties are estimated by characterizing the width or spread of the posterior. The most often used means of summarizing uncertainties is in terms of root mean-square deviation. Another often relevant characterization is in terms of confidence interval, e.g., stating the probability that the true value of a parameter is within a specified interval. The point is that the posterior can be used to determine any characterization of the uncertainties that one wishes to use.

One of the potential difficulties in probabilistic analysis is handling *nuisance parameters*, that is, parameters that are necessary to describe the model of the physical situation, but are ultimately of no interest. For example, if the initial state of the simulation of an experiment is uncertain, it would be important to include that uncertainty in the analysis of the data because it will degrade the uncertainty in the desired model parameters. However, the estimated initial state is immaterial, as far as model inference is concerned. The posterior will typically be a joint probability distribution in all the parameters, those of interest, as well as the nuisance parameters. The appropriate way to handle nuisance parameters is to integrate the joint distribution over the useless parameters, a process called *marginalization*. If there are a lot of nuisance parameters, this marginalization can be difficult, even computationally intractable.

A subtle, but real, source of uncertainty in simulation predictions is the uncertainty associated with the form of the incorporated physics models. Often the form used to represent a physical relationship is heuristically chosen. Even if the form is chosen on the basis of physical reasoning, it may still be incorrect. Uncertainties in the form of the models, also called *structural uncertainties* [18,19], obviously must be taken into account in assessing simulation uncertainties. The general approach to estimating structural uncertainties is to consider alternative suitable models and determine how well they are rejected by experimental data.

There is a close connection between these structural uncertainties and the issue of the operating range of the underlying independent physical variables that have been measured by the experiments. When it is necessary to extrapolate beyond range probed by the available experiments, structural uncertainties will lead to rapid growth in the uncertainties in the model behavior.

2.2 Gaussian approximation

It is often the case, or at least often assumed, that uncertainties follow a Gaussian distribution, which means that the minus-log-posterior φ is quadratic:

$$\varphi = \varphi_0 + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^{\mathbf{T}}\, \mathbf{K}\, (\boldsymbol{\theta} - \boldsymbol{\theta}_0) \quad , \tag{3}$$

where $\boldsymbol{\theta}_0$ is the parameter vector at the minimum in φ and $\mathbf{K}$ is the curvature matrix of φ,

$$\mathbf{K} = \nabla^2_{\boldsymbol{\theta}} \varphi \quad . \tag{4}$$

This matrix is also called the *Hessian*. It is fundamentally important in uncertainty analysis because of the the well-know relationship that the covariance matrix of a multi-dimensional Gaussian is given by $\mathbf{C} = \mathbf{K}^{-1}$.

The *covariance matrix* is the second moment of the posterior about its mean: $[C]_{ij} =< (\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}_i)(\boldsymbol{\theta}_j - \bar{\boldsymbol{\theta}}_j) >$, where the angle brackets indicate an expectation over the posterior distribution. It completely summarizes the second-order statistics of a Gaussian posterior, including its width and correlations between the uncertainties in different parameters through its off-diagonal elements. Under the assumption of Gaussian distributions and using $\varphi = \frac{1}{2}\chi^2$, the calculation of the covariance matrix described above is equivalent to standard uncertainty analysis [13]. Marginalization is taken into account in the covariance matrix $\mathbf{C}$ in the sense that one can simply ignore the elements of $\mathbf{C}$ corresponding to the nuisance parameters.

For Gaussian prior and likelihood distributions, characterized by their Hessians $\mathbf{K}_{in}$ and $\mathbf{K}_{exp}$, and by their mean positions, $\boldsymbol{\theta}_{in}$ and $\boldsymbol{\theta}_{exp}$, respectively, Bayes law, in its logarithmic form, yields the simple result for the posterior

$$\mathbf{K}_{out} = \mathbf{K}_{exp} + \mathbf{K}_{in} \tag{5}$$
$$\boldsymbol{\theta}_{out} = \mathbf{K}_{out}^{-1} \left[\mathbf{K}_{exp}\boldsymbol{\theta}_{exp} + \mathbf{K}_{in}\boldsymbol{\theta}_{in}\right] \quad , \tag{6}$$

where $\mathbf{K}_{out}$ is the Hessian and $\boldsymbol{\theta}_{out}$ the mean position of the posterior. One obtains the usual rule for combining the covariance and the mean for the product of two Gaussian distributions [14] by replacing the Hessians in (5) and (6) with the inverse of the corresponding covariance matrices.

A novel method of estimating uncertainties for Gaussian distributions is to probe the *model stiffness* [20]. This method amounts to minimizing φ in the presence of a force applied to the model. The displacement of the parameters away from their MAP solution is given by $\Delta\boldsymbol{\theta} = \mathbf{C}\mathbf{f}$, where $\mathbf{f}$ is the applied force, specified in the space of the parameters. This method provides correlations between every parameter and the probing force vector.

2.3 Markov chain Monte Carlo

The *Markov chain Monte Carlo* (MCMC)technique provides a way to generate a sequence of random parameter vectors drawn from an arbitrary target PDF. The usefulness of MCMC in probabilistic analysis is well established [21–24].

The simplest approach is to use the Metropolis algorithm [25], in which one tries to move from the current position in parameter space by randomly selecting a trial displacement from a symmetric probability distribution. Each trial step is either accepted or rejected on the basis of the probability at the new position relative to that at the previous one. This algorithm is widely employed because of its simplicity. In the data-flow diagram shown in Fig. 1, MCMC would take the place of the optimizer, allowing it to change the parameters and observe the corresponding minus-log-posterior. The covariance matrix $\mathbf{C}$ may be obtained using MCMC by directly computing the second moments of the MCMC sequence.

One of the advantages of MCMC is that it can handle most arbitrary distributions, alleviating the need to approximate the posterior by a Gaussian distribution. As such, MCMC provides a generally usable means to determine the extent of a posterior and thus to assess uncertainties. Another advantage of MCMC is that it automatically accomplishes marginalization over nuisance parameters. By sampling the full joint probability distribution, as MCMC does, when one determines the distribution of a subset of parameters, the remaining parameters are automatically integrated out.

The inefficiency of MCMC is a potential problem. The steps for the Metropolis algorithm are often chosen independently for each parameter. However, most posterior distributions possess some degree of correlation between the parameters. By not taking these correlations into account, the independent step distribution can lead to substantial calculational inefficiency. This inefficiency results in correlation between successive samples taken from the MCMC sequence, which means that a reduced number of samples can adequately represent the full sequence. A number of schemes for improving the efficiency of MCMC exist [26–30], most of which are adaptive, and many of which require the gradient of φ with respect to the parameters. Therefore, adjoint differentiation can be very helpful in making MCMC more efficient.

3 Uncertainty in simulation-code predictions

The goal of the present work is to determine the uncertainties in simulating a new situation, which are produced by the uncertainties in the physics models. A conceptually simple Monte Carlo approach to obtaining these uncertainties is shown in Fig. 2. The process amounts to drawing random parameter vectors from their posterior distribution, which can be thought of as comprising a plausible set of parameter vectors. Running the simulation code separately for each random parameter vector results in a set of corresponding plausible simulation predictions, which can be used to characterize the uncertainty in the prediction in whatever way appropriate. This process produces the *posterior* *predictive distribution* [22], as it is known in statistics.

The same technique can be employed if one is working directly with the Hessian or covariance matrices to describe the posterior. Then matrix techniques can be used to directly generate a sequence of random draws from the posterior, approximated by a Gaussian. One needs to calculate the square root of the covariance matrix $\mathbf{C}^{\frac{1}{2}}$ by singular-value decomposition [31]. Random parameter vectors with the correct covariance are then obtained by multiplying random vectors drawn from an independent, univariate Gaussian distribution by $\mathbf{C}^{\frac{1}{2}}$.

A virtue of most Monte Carlo calculations is that new sources of stochastic behavior are easily incorporated. For example, uncertainties in the initial state of a physical system may be included in the above process by randomly drawing initial states from the probability distribution associated with those uncertainties each time the simulation is run with a new random parameter vector.

Other approaches to estimating the uncertainty in a simulation prediction may potentially be useful, including extensions to a posteriori error analysis [32], which can effectively propagate the error through the simulation. However, because this technique requires an auxiliary forward code that solves another set of differential equations, it may not be easy to implement.

4 Analysis of many experiments

Consider the sequential analysis of two experiments, as depicted by the *bubble diagram* shown in Fig. 3. Experiment 1 is a basic experiment since it only involves a single model, associated with $\boldsymbol{\alpha}$. Using the type of analysis described in Sect. 2.1, starting with a flat prior $p(\boldsymbol{\alpha})$, the analysis of this experiment results in improved knowledge about the parameter vector $\boldsymbol{\alpha}$, as summarized by the posterior $p(\boldsymbol{\alpha}|\mathbf{Y}_1)$. The second experiment requires the model associated with $\boldsymbol{\alpha}$, as well as another model associated with $\boldsymbol{\beta}$, to analyze. It is an integrated experiment. One approach might be to consider $\boldsymbol{\alpha}$ to be completely determined by Experiment 1 and keep it fixed in the second analysis. However, it seems desirable to use Experiment 2 to learn something further about $\boldsymbol{\alpha}$. The implication is that both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ should be varied in analyzing the second experiment.

4.1 Probabilistic network

A full probabilistic analysis of the two experiments yields the posterior in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, given the experimental measurements from both experiments. Using the rules of probability:

$$\begin{aligned} p(\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{Y}_1,\mathbf{Y}_2) &\propto p(\mathbf{Y}_1,\mathbf{Y}_2|\boldsymbol{\alpha},\boldsymbol{\beta})\, p(\boldsymbol{\alpha},\boldsymbol{\beta}) &(7)\\ &= p(\mathbf{Y}_1|\boldsymbol{\alpha})\, p(\mathbf{Y}_2|\boldsymbol{\alpha},\boldsymbol{\beta})\, p(\boldsymbol{\alpha})\, p(\boldsymbol{\beta}) \ \ . &(8)\end{aligned}$$

In going from (7) to (8), it is necessary to invoke statistical independence of the priors on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and of the measurements from the two experiments. Taking the minus logarithm of Eq. (8), we obtain

$$\begin{aligned} -\log[\,p(\boldsymbol{\alpha},\boldsymbol{\beta}|\mathbf{Y}_1,\mathbf{Y}_2)] = &-\log[\,p(\mathbf{Y}_1|\boldsymbol{\alpha})] - \log[\,p(\boldsymbol{\alpha})] \\ &-\log[\,p(\mathbf{Y}_2|\boldsymbol{\alpha},\boldsymbol{\beta})] - \log[\,p(\boldsymbol{\beta})] \ \ . \qquad (9)\end{aligned}$$

Comparison of Eq. (9) with Fig. 3 shows that the effective action of each bubble is to add its inputs, consisting of minus-log-priors, to the minus-log-likelihood of its experimental measurements to get the minus-log-posterior. Each bubble implements Bayes law in its logarithmic form (2). Equation (9) illustrates the cumulative property of Bayes law; by using the posterior of one analysis as the prior of the next, the result is the same as if both likelihoods were simultaneously used in a single grand analysis. We see that Fig. 3 actually represents a *probabilistic network*.

Note that the output of each bubble is a PDF over the parameters shown next to the output. A consequence of this approach is that the joint PDF involves increasingly more parameters as one moves up the hierarchy from basic experiments up to fully integrated ones.

The crucial requirement for the simplicity of the additive rule is that the experimental measurements be independent. If they are not independent, that can be taken into account in the algebra associated with the bubbles.

A more elaborate probabilistic network involving five experiments is shown in Fig. 4. It is clear from the diagram that Experiments 1, 3, and 4 are basic experiments. Experiment 2 involves a low level of integration since it provides information about two models. Experiment 5 is fully integrated since it incorporates all the models. The output of the last bubble is a joint PDF in all the model parameters.

This kind of bubble diagram has clear benefits for describing a sequence of analyses. I will explain in the next section how such a probabilistic network forms a useful basis for a computational approach to keeping track of a complicated series of analyses. The bubble diagrams shown in Figs. 3 and 4 are similar to other forms of probabilistic networks [33] or Bayesian networks [34].

4.2 Implementation considerations

Validation of complex simulation codes that employ many physics models will typically require numerous experiments, performed over a wide range of levels of integration. The task of comprehending which experiments have been conducted and how their results influence the final configuration of the simulation code is potentially overwhelming. It is natural to consider using the bubble diagram described in the preceding section as a means for graphically depicting the flow of the analysis. The corresponding probabilistic network provides the means to quantitatively summarize our knowledge about the models.

The validation process would greatly benefit by implementation of the probabilistic network using an *object-oriented (OO) design*. The graphical display of the full analysis sequence would help visualize the logic of the analysis. One would have the ability to check intermediate results in the analysis sequence, which may be important when it comes to model checking (see next section). Since each bubble represents an experiment and its related data analysis, the information associated the probabilistic network really forms a database for the validation process. The logic of the graph may be checked automatically, e.g., by checking dependencies of final results on each experiment to make sure a single experiment is not counted more than once.

Operationally, the output of each bubble represents a PDF defined over a certain set of parameters. In a Gaussian approximation, this joint PDF is defined in terms of a mean vector and covariance matrix. Because of the simplicity of using minus-log-probabilities already mentioned, it is proposed to use the curvature matrix (or Hessian), instead of the covariance matrix, to represent the width of the Gaussian. Of course, as one moves through the sequence of analyses, the number of parameters steadily increases. This varying data structure is easy to handle in an OO design. Because of the additive nature of the logarithmic form of Bayes law (9), updates of the parameter mean and the Hessian affect only the parameter components over which they are defined. When new parameters are included, they do not mix with the old parameters except through the likelihood of a new experiment. Whenever the covariance describing the uncertainties in the parameters is desired, it is necessary to invert the Hessian.

Critical to the proper functioning of the probabilistic network is the quality of the information employed in each bubble. A bubble must supply a probabilistic description of the likelihood in terms of the model parameters used to analyze the experimental measurements. Thus, a full uncertainty analysis must be carried out, taking into account the details described above, e.g., marginalization over nuisance parameters, such as uncertainties in the experimental initial conditions. It is perhaps simplest to think of that process in terms of using MCMC to sample the likelihood, computing the mean vector and covariance matrix of the relevant parameters, and taking the inverse of the covariance matrix to obtain the Hessian.

As mentioned above, one must be careful to not count the contribution from any given likelihood twice. Therefore, the output of each bubble is not really just the sum of its input minus-log-priors and internal minus-log-likelihood. There is a need to check dependencies at each point where the accumulated posterior is desired. One way to handle this is to have each accumulating bubble broadcast backwards through the graph a request for contributions. As each bubble connected to it through the backward path responds, duplicated contributions can be flagged and avoided in the summation process.

Finally, it is possible to translate an instantiated probabilistic network into a computer script, which effectively summarizes the network. This script can be used to archive the analysis or to translate it into text for a written summary.

4.3 Model checking

Model checking is an essential aspect of any effort to build and understand models [22]. *Model checking* amounts to looking for inconsistencies between experimental measurements and the models. This process ideally takes place at every level of analysis, from analyzing separate data runs in a single experiment, to the analysis of the experiment in terms of a model, all the way up to the comparison between several experiments. Since in the scheme suggested here, all parameters are subject to change as new experiments are added to the comprehensive analysis, one must be able to revisit previous analyses and resolve discrepancies. All disagreements should be diagnosed to determine the source of problem, namely, are the experimental results suspect, or plainly wrong, or does the physics model need to be altered.

It may be helpful to try to identify model shortcomings using sensitivity analysis. One of the most potent uses of adjoint differentiation, when viewed in terms of the elemental representation of a model, is to recognize how the structure of that model needs to be changed to accommodate the data [9].

Model-based approaches to dealing with discrepant data, or *outliers*, will be useful in model checking. One approach is to assign to each experimental result a probability that it is acceptable, or similarly, a multiplicative deweighting factor [35]. In combining several experiments, the weight of each one is effected by how well it agrees with the others. Another approach is to treat the likelihood function of each experiment, not as a Gaussian, but as a long-tailed distribution, such as the Cauchy distribution [36], which effectively invokes a principle of majority rule when several results are combined.

5 Discussion

I have presented a framework for tackling one of the basic tasks needed for validating simulation codes, that of assessing the uncertainties in predictions that arise from uncertainties in the underlying models. Many details regarding implementation need to be resolved. How these are best handled will depend on the specifics of simulation code and the kind of measurements available. The suggested implementation is consistent with the notion that the validation should be viewed as an ongoing process [2]. Its dynamic and interactive nature permits one to link experimental results to a progressively changing simulation code, determine weak spots in the models, and design and conduct new experiments for reducing the uncertainty in predictions.

Acknowledgements

Many have helped me understand various aspects of simulation codes and uncertainty estimation in a Bayesian context including Julian Besag, Anges de Crécy, Greg Cunningham, Eric Ferm, Wally Gilks, James Gubernatis, Rudy Henninger, Malvin Kalos, Bill Oberkampf, Richard Silver, John Skilling, and Tim Trucano.

References

[1] W. L. Oberkampf and F. B. Blottner, Amer. Inst. Aeron. and Astron. J. 36 (1998) 687

[2] *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, Tech. Rep. G-077-98 (Amer. Inst. Aeronautics and Astronautics, 1998).

[3] S. G. Rabinovich, *Measurements Errors: Theory and Practice* (Amer. Inst. Phys., New York, 1995).

[4] W. C. Thacker, Automatic differentiation from an oceanographer's perspective, in *Automatic Differentiation of Algorithms: Theory,*

Implementation, and Application, edited by A. Griewank and G. F. Corliss (SIAM, Philadelphia, 1991) 191.

[5] R. Giering, Tangent linear and Adjoint Model Compiler, Technical Report TAMC 4.7 (Max-Planck-Institut für Meteorologie, 1997) (e-mail: giering@dkrz.de).

[6] G. Burgers, R. Giering, and M. Fischer, Ann. Geophysicae. C14 (1996) 390.

[7] K. M. Hanson and G. S. Cunningham, The Bayes inference engine, in *Maximum Entropy and Bayesian Methods*, edited by K. M. Hanson and R. N. Silver (Kluwer Academic, Dordrecht, 1996) 125.

[8] S. S. Saquib, K. M. Hanson, and G. S. Cunningham, Proc. SPIE 3034 (1997) 369.

[9] K. M. Hanson, G. S. Cunningham, and S. S. Saquib, Inversion based on computational simulations, to be published in *Maximum Entropy and Bayesian Methods*, edited by G. Erickson (Kluwer Academic, Dordrecht, 1998).

[10] R. J. Henninger, P. J. Maudlin, and M. L. Rightley, Accuracy of differential sensitivity for one-dimensional shock problems, to be published in *Shock Compression of Condensed Matter*, edited by S. C. Schmidt, et al. (Amer. Inst. Physics, Woodbury, NY, 1998).

[11] M. L. J. Rightley, R. J. Henninger, and K. M. Hanson, Adjoint differentiation of hydrodynamic codes, in *CNLS Research Highlights* (Center for Nonlinear Studies, Los Alamos National Laboratory, April, 1998); WWW: http://cnls.lanl.gov/Publications/highlights.html.

[12] Khoral Research, Inc., 6200 Uptown Blvd. NE, Albuquerque, NM 87110-4142; URL: www.khoral.com.

[13] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill, New York, 1992).

[14] W. T. Eadie, D. Dryard, F. E. James, R. Roos, and B. Sadoulet, *Statistical Methods in Experimental Physics* (North-Holland, Amsterdam, 1971).

[15] J. J. K. Ó Ruanaidh and W. J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing* (Springer, New York, 1996).

[16] D. S. Sivia, *Data Analysis: A Bayesian Tutorial* (Clarendon, Oxford Univ. Press, New York, 1996).

[17] D. J. C. MacKay, Neural Computation 4 (1992) 415.

[18] M. D. McKay and J. D. Morrison, Structural model uncertainty in stochastic simulations, in *Computing Science and Statistics 28* (Interface Foundation, Fairfax Station, VA 22039-7460, 1997).

[19] D. Draper, J. R. Statist. Soc. B 57 (1995) 49.

[20] K. M. Hanson and G. S. Cunningham, Proc. SPIE 2434 (1995) 416.

[21] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice* (Chapman and Hall, London, 1996).

[22] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis* (Chapman & Hall, London, 1995).

[23] J. Besag, P. Green, D. Higdon, and K. Mengersen, Stat. Sci. 10 (1995) 3.

[24] K. M. Hanson, G. S. Cunningham, and R. J. McKee, Int. J. Imaging Systems and Technology 8 (1997) 506.

[25] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. 21 (1953) 1087.

[26] W. R. Gilks and G. O. Roberts, Strategies for improving MCMC, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Chapman and Hall, London, 1996) 89.

[27] A. E. Raftery and S. M. Lewis, Implementing MCMC, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Chapman and Hall, London, 1996) 115.

[28] K. M. Hanson and G. S. Cunningham, Proc. SPIE 3338 (1998) 371.

[29] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer, New York, 1996).

[30] J. Skilling, D. R. T. Robinson, and S. F. Gull, Probabilistic displays, in *Maximum Entropy and Bayesian Methods*, edited by W. T. Grandy, Jr. and L. H. Shick (Kluwer Academic, Dordrecht, 1991) 365.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations – Third Edition* (Johns Hopkins University, Baltimore, 1996).

[32] D. Estep, SIAM J. Numer. Anal. 32 (1995) 1.

[33] G. Shafer, *Probabilistic Expert Systems* (SIAM, Philadelphia, 1996).

[34] F. V. Jensen, *An Introduction to Bayesian Networks* (Springer, New York, 1997).

[35] V. Dose and W. von der Linden, Outlier tolerant parameter estimation, to be published in *Maximum Entropy and Bayesian Methods*, edited by V. Dose et al. (Kluwer Academic, Dordrecht, 1998).

[36] K. M. Hanson and D. R. Wolf, Estimators for the Cauchy distribution, in *Maximum Entropy and Bayesian Methods*, edited by G. Heidbreder (Kluwer Academic, Dordrecht, 1996) 255.

Figure Captions

Fig. 1. Data-flow diagram showing the general scheme for analyzing the measurements $\mathbf{Y}$ from a single experiment. The simulation code uses the parameters α to predict the time-dependent behavior of a physical system. A measurement system model describes the connection between the physical system and the experimental measurements. The parameters are estimated by minimizing the difference between the measurements predicted by the simulation code $\mathbf{Y}^*$ and the actual $\mathbf{Y}$, as quantified by the minus-log-likelihood.

Fig. 2. Data-flow diagram showing how the uncertainties in a simulation prediction arising from parameter uncertainties can be estimated by running the simulation code for a set of plausible parameters $\{\boldsymbol{\alpha}\}$. The resulting set of plausible predictions represents the uncertainty distribution of the prediction.

Fig. 3. A sequential analysis involving two experiments. Interpreted as a probabilistic network, each bubble represents the use of Bayes law by combining the likelihood PDF from the measurements with the PDF inputs from the left for the prior information to obtain the output PDF on the right, the posterior.

Fig. 4. A probabilistic network for analyzing five experiments results in the joint posterior in the parameters for all models involved, i.e., α, β, γ, and δ.