

# Quantifying the Uncertainty of Computational Predictions

Robert G. Easterling

Sandia National Laboratories<sup>1</sup>  
 P.O. Box 5800  
 Albuquerque, NM 87185-0417

**Abstract**

Confidence in a computational prediction is enhanced if its potential ‘error’ (the difference between the prediction and nature’s outcome for the event being modeled) can be credibly bounded. To develop such bounds I first develop a conceptual framework for designing and conducting a suite of physical (model-validation) experiments and calculations, then analyzing the results to characterize prediction uncertainty under the experimental conditions and to provide a basis for inferring the uncertainty of a computational prediction in a system application environment or configuration that cannot or will not be tested. Attendant issues and potential solutions are discussed, then illustrated via a shock physics example.

**Nomenclature**

- $M(x;\phi)$  computational model
- $x$  vector of model input variables
- $\phi$  vector of model parameters
- $y^*(x)$  computational prediction at  $x$
- $y(x)$  experimental outcome at  $x$
- $e_x$  prediction error at  $x$
- $E(.)$  expectation
- $\delta_x$  expected value of  $e_x$
- $var(.)$  variance
- $\sigma_x^2$  variance of  $e_x$  at  $x$
- df degrees of freedom
- $t_{.025}(f)$  the .025 quantile on the t-distribution with  $f$  df

**Introduction**

Users of computational predictions, from designers to decision-makers, need to be provided with information on how accurate the prediction is and on what basis. E.g., the goal is a statement like “Based on our understanding of the underlying physics, our ability to translate that understanding to a computational code, and our analysis of an extensive suite of experiments and corresponding computations, we are confident that actual system response will differ from the computational prediction by no more than 30%.” Such prediction uncertainty limits define predictive capability and provide the necessary yardstick against which a computational prediction can be compared to a requirement. Obtaining credible, defensible limits-of-error for computational predictions of complex phenomena, however, is an extremely challenging problem

**Framework**

Confidence in computational predictions comes (in large part) from comparisons with data. The term model-validation<sup>[1, 2]</sup> is conventionally used for this comparison and experimental programs are conducted for this purpose. Model-validation experiments can range from single-phenomenon tests, through a range of combined phenomena tests, to system-level multi-phenomena tests. Test units can range from simple geometric shapes of single materials to complex assemblies. At each level, comparisons of computational predictions to experimental results provide information on predictive capability. My view is that the full purpose of model-validation experiments should be the measurement of predictive capability and uncertainty.

Figure 1 is my view of this total process, set in the context of comparing a computational prediction to a system requirement.

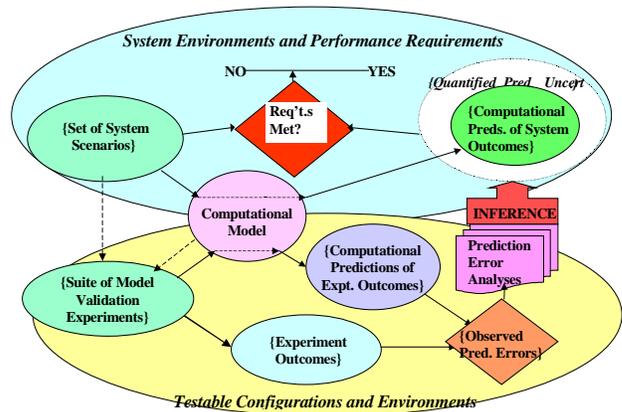


Figure 1. Schematic for Characterization of Prediction Uncertainty

The top ellipse in Fig. 1 depicts the intended use of the computational tool: system requirements specify various performance goals and the computational model will be used to predict system performance in scenarios that embody these requirements. Comparing the prediction to the requirement requires an uncertainty yardstick, or frame of reference, depicted by the uncertainty ‘cloud’ surrounding the prediction. To develop such a yardstick, experiments and computations must be conducted – depicted by the bottom ellipse. The design of these

experiments is driven by the system scenarios and the structure of the computational model. These experiments and computations provide first for an evaluation of prediction capability in the situations tested. Next, and most importantly, the ensemble of observed differences are potentially the basis of an inference about prediction uncertainty in the system applications of interest -- the connection to the upper ellipse.

To frame this paper's discussion I mathematically represent a prediction generated by a computational simulation as:

$$y^*(x) = M(x;\phi), \quad (1)$$

where  $M(x;\phi)$  represents the computational model of the phenomenon of interest;  $x$  is model input variables,  $\phi$  is model parameters; and  $y^*$  is the model output or *prediction*. All the terms in expression (1),  $x$ ,  $\phi$ , and  $y^*$ , could be vectors or fields.

In general, the model's input vector  $x$  describes a physical entity and the environment to which it is subjected. This vector will include physical dimensions, materials, environmental variables, and initial and boundary conditions. The numerical model parameter vector  $\phi$  includes parameters that are needed to specify material response in the model. Think generally of  $\phi$  as constants such as transfer coefficients in the set of equations on which  $M$  is based. I further assume that all numerical aspects of  $M(x;\phi)$ , such as grid size, time steps, and convergence criteria, are fully specified in  $M$ . The computer model,  $M(x;\phi)$ , is thus an operator that transforms input  $x$  into the predicted result  $y^*$ . This transformation is assumed to be deterministic in this paper in the sense that for a given specification of  $x$  and  $\phi$  the code always gives the same  $y^*$ . Repeated runs of a deterministic code, as in a Monte Carlo analysis, however, will be considered.

As an example of this mathematical representation, in a Sandia case study pertaining to the vaporization of foam in a thermal environment,  $x$  included the Temperature-time environment to which a foam specimen is subjected as well as variables that define the foam composition and geometry. The parameter vector  $\phi$  included the 'activation energies' associated with various chemical structures in the foam. The response  $y^*$  was the remaining solid fraction of the foam specimen as a function of time.

Now, corresponding to the prediction,  $y^*(x)$ , consider an experiment conducted at the specified  $x$  and represent its outcome by  $y(x)$ . I define the *prediction error* of the model at  $x$  as

$$e_x = y(x) - y^*(x) \quad (2)$$

Evaluating model predictive capability means characterizing  $e_x$  at selected  $x$ -points or over selected  $x$ -regions. This evaluation requires selecting a set of  $x$ -points, then obtaining computational predictions and experimental results for each.

### Statistical Model

All computational models, no matter how extensively the underlying processes are modeled, are approximations to nature. Things happen in nature that are not captured by a computational model. Consequently, a variety of random and systematic effects can contribute to prediction error. For these reasons, my approach to model-validation experimental design, data analysis, and the subsequent quantification of prediction uncertainty will be via the statistical model:

$$y(x) = y^*(x) + e_x, \quad (3)$$

where  $e_x$  is a random variable with an unknown probability distribution that possibly depends on  $x$ .

George Box<sup>[3]</sup> has stated, "All models are wrong, but some are useful." The nature of  $e_x$  in (3) will determine the usefulness of a model. Understanding the nature of  $e_x$  should be the purpose of model-validation experimentation.

Equation (3) is a statistical model of the relationship between the 'true' experiment outcome  $y$  and model prediction  $y^*$ . What we observe in practice are measurements of nature, so observed  $e_x$  contains measurement error. Methods for adjusting for this source of variation in the observed prediction errors will be discussed below. For the moment, I treat measured  $y$  as nature's  $y$ .

Viewing the differences between experiment and model as statistical has engineering precedent. For example, in bridge design, civil engineers use a mathematical model for "scour" – the erosion of soil around a bridge's foundation due to river flooding<sup>[4]</sup>. This model is a function of soil type, flood magnitude, river velocity and other pertinent variables. For predictions civil engineers incorporate an additional "modeling factor" to represent the deviation of actual scour depths from the theoretical model. This modeling factor corresponds to  $e_x$  in (3).

A key aspect of the statistical model (3) is that the probability distribution of prediction error is unknown. Thus, measuring prediction capability is fundamentally approximate and nondeterministic, that is, fundamentally statistical. Any measure of predictive capability and the confidence derived thereby will be derived from the experimental data and corresponding computational predictions and hence will be an "estimate" in the statistical sense. In terms of (3), the problem of assessing model prediction capability is first to estimate the probability distribution of  $e_x$  at the  $x$ -points at which computations and experiments are conducted. This must then be followed by estimation of the distribution of  $e_x$  at  $x$ -points pertaining to physical entities and environments that have not, can not, or will not be tested. From this estimated distribution one can statistically bound  $e_x$  and hence statistically bound the difference between computational prediction and nature. A further objective is to characterize the reliability of the estimate. Statistical confidence and tolerance limits<sup>[5]</sup> provide such characterizations.

Implementing the process represented by Fig. 1 and the analysis based on eq. (3) leads to a variety of issues. The next two sections discuss some of the problems that are likely to be encountered and indicates directions to take.

### Experimental Design

In broad terms, model-validation experimental design means selecting a set of  $x$ -points (that define test hardware and environments) at which to do experiments and computational predictions. In detail, this also means determining experimental plans that specify the test hardware, methods, conditions, instrumentation, data collection, and post-processing techniques used to obtain information required for subsequent data analyses. All of these elements have different nuances for experiments that are designed for model validation studies as opposed to phenomena discovery or exploration. It is critical to emphasize this point. It is also important to recognize at the outset that measuring predictive capability has profound implications for the experimental sciences, not just the analytic.

The role of experimental design in the inference problem is illustrated in Fig. 2 in which the space of validation experiments and system applications is defined by two meta-variables, configuration and environment. Because of economic and other reasons it may not be possible to test actual systems in their required environments. (For this reason, Fig. 2 depicts an extrapolation situation; intuitively, interpolation should be easier.) Thus, we have to extend what we can learn about predictive capability (represented by the prediction errors,  $\{y(x)-y^*(x)\}$ , in Fig. 2) at the selected  $x$ -points where we can evaluate it to an inference about predictive capability where we cannot. This inference requires an extension of the model itself plus an extension of what we know about unmodeled phenomena, as represented by the observed prediction errors. Making this extension successfully and credibly requires subject-matter knowledge about the axes along which we can make such extensions and it requires a suite of experiments suitably located in the configuration-environment space to provide the data necessary to make such extensions. The design of this set of experiments thus has to be driven by the ultimate applications for which computational predictions and a model's predictive capability are required, as was discussed above.

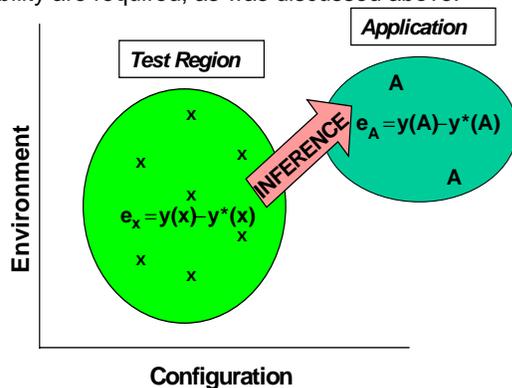


Figure 2. Inferring predictive capability

### a. Experimental Objectives.

Meaningful validation experiments are designed to meet one or more explicit objectives. In general, the experiments conducted (1) should provide a sufficient test of predictive capability for the selected experimental situations and (2) the collective set of experiments and associated computational predictions should provide a basis for making the desired inference of predictive confidence.

There are various ways to translate the first objective into a basis for experimental design. For example, one measure of predictive capability at  $x$  is the standard deviation of prediction error,  $e_x$ , at that point. One could define the objective to estimate this standard deviation within P% and then derive the number of experiments required to achieve that precision. These experiments could either be  $n$  replications at the selected  $x$ -point or  $n$  total experiments at different  $x$ -points within a region within which it is reasonable to expect a constant standard deviation.

The conduct of a validation experiment also influences how well predictive capability can be measured. As mentioned above, a variety of random and systematic factors can differentiate computational prediction and nature. Validation experiments need to be conducted in ways that allow these factors to be manifested as they would in an application of interest. For example, predictive capability measured in a tightly-controlled, pristine lab environment may not be appropriate for inferring predictive capability for predictions for a much less controlled, noisier application environment. The objective of assessing predictive capability in a specific application influences experimental design in terms of both what is controlled and what is not controlled in the experiments.

Time, resources, and experimental capability constrain validation experimental design and conduct. Such constraints must be balanced against the experimental objectives in arriving at a plan for model validation experimentation. A difficult decision will have to be made as to whether a meaningful evaluation of predictive capability is possible under existing constraints in any given situation.

### b. Experiment-Model Compatibility.

The computational and experimental elements of the model validation process cannot be executed in isolation. The vector  $x$  needs to be meaningful to both the experimentalist and modeler in order to align experiment and model so that both computational predictions and experiments at selected  $x$ -points can be run and compared. Further, this alignment needs to be meaningful in terms of the system scenarios for which computational predictions are required.

The discussion so far has assumed that the full  $x$ -vector could be controlled or measured in an experiment. If the modeler's  $x$ -vector contains variables that have no experimental meaning, this is not the case and it may not be possible to make meaningful comparisons. If the modeler's  $x$ -vector requires measurements that cannot be made, the result will be increased prediction uncertainty.

To avoid this misalignment, there may be a need to develop new experimental and instrumentation capabilities. The definition of the variables in the  $x$ -vector is not just a modeling issue. The experimenter, the requirements-setter, and the decision-maker have to be able to operate and communicate in terms of this  $x$ -space.

### c. *Simplification.*

The objective of characterizing predictive capability over some high-dimensional  $x$ -space can quickly require an experimental design that exceeds available or reasonable resources. One way to avoid this problem is to vary only a subset of the variables in  $x$  while holding the others fixed at nominal or bounding, values. Statistical experimental design methods<sup>[6]</sup> should be used to efficiently and adequately explore the specified  $x$ -space.

Model simplification is another route to reduce the cost of predictive capability measurement. For example, suppose a model contains high-order effects or phenomena that cannot be controlled or measured in an experiment. It may be more appropriate to make computational predictions without those effects in the model and then capture those effects experimentally through the observed prediction errors. Where computational resources are constraining factors, model simplification increases in importance and attractiveness, but may also increase the complexity of inferring confidence from the validation process.

### **Analysis.**

After conducting a suite of experiments and computational predictions the next task is to analyze the resulting data,  $\{x_i, y(x_i), y^*(x_i) : i = 1, 2, \dots, n\}$ . It is important to note that the subscript  $i$  refers to distinct experiments. Both  $y$  and  $x$ , though, may be fields or vectors containing thousands of measurements or calculations. It is decidedly not the case, however, that thousands of measurements, e.g., of temperature over a fine grid of space and time, for one experiment is equivalent to thousands of separate experiments. The number and nature of the experiments conducted will determine the precision with which predictive capability is measured, not the number of measurements per experiment.

Given the computational and experimental outcomes from the suite of experiments, the objective of the analysis of these results is to measure and/or estimate predictive capability. The following subsections address issues that arise in this analysis.

### a. *Metrics.*

Predictive capability at an  $x$ -point can be characterized by a variety of "parameters" (in the statistical sense of being a characteristic of a probability distribution) of the probability distribution of  $e_x$ . The expected value and the standard deviation of  $e_x$  are two important possibilities. Others might be the square root of the expected squared error, the 99<sup>th</sup> percentile of the distribution of absolute error; the lower and upper 95<sup>th</sup> percentiles on the distribution of  $e_x$ ; and others.

If the computational model was designed to be conservative on the high-side (i.e.,  $e_x$  is intended to be negative), the metric of interest might be  $\text{Prob}(e_x < 0)$ . When  $e_x$  has a normal distribution all of these distributional characteristics are functions of the two parameters that characterize a normal distribution, the mean and the standard deviation.

Any of these measures of predictive capability must be estimated from the experimental and computational results. With limited data, estimation uncertainty will be appreciable. Statistical methods account for estimation uncertainty by methods such as confidence and tolerance limits<sup>[5]</sup>. For example, a conclusion might be stated as: with 90% confidence the upper 95<sup>th</sup> percentile of the distribution of  $e_x$  is no more than  $UL_{90/95}$ . The essential point is that any "metric" of predictive capability derived from the model validation process will be a statistical estimate and the reliability of that estimate must also be evaluated and communicated.

### b. *Choice of Analysis Variables.*

In both experiments and computations there are a large number of variables that can be observed and compared. Making the analysis manageable and the results meaningful and communicable requires a careful selection of variables for which to evaluate predictive capability.

The selection of variables should first be driven by system requirements. If the requirement is that peak strain at a given location should not exceed some value, for example, then the model validation objective is to measure the predictive capability pertaining to calculated peak strain at that location. While it would add confidence in the computational model to know that the complete strain vs. time history at various sites in the test device can be reasonably well predicted, it is really not appropriate to devote a lot of analysis to measuring predictive capability over an extensive time and space grid. This requirements focus is also a way to greatly reduce the dimensionality of the data, which in general may be time-histories of responses such as acceleration, strain, or temperature in time and space, to a small number of 'integral' variables such as peak acceleration or the time to reach critical temperatures at selected points in a system or component.

### c. *Inference.*

The inference bridge in Fig. 2 can be constructed if the underlying scientific relationships are known to extend over a region containing both the  $x_A$  and the  $x_i$ , and if there is a credible basis for similarly extending the error distribution which, after all, reflects factors in nature not captured by the scientific model. When there is a mathematical connection, statistical methods can account for and reflect the 'distance' between these points. The greater the distance, the greater the prediction uncertainty. In passing, I note that the spatial representation of the experimental design (in  $x$ -space) and inference problems suggests that spatial statistical methods<sup>[7]</sup>, such as kriging, can be used to model a metric, such as the estimated standard deviation at  $x$ , as a function of  $x$ , then estimate the value of

that metric at  $x_A$  and estimate the uncertainty of that estimate.

There may be many situations in which a mathematical/statistical inference bridge cannot be built. In such cases, the linkage may have to be achieved through expert judgment. E.g., "In our suite of model-validation experiments, we never observed a prediction error greater than 20%. In light of the additional complexity of the system application, we believe the prediction error could be no worse than 40%."

If no credible inference is possible, one may have to re-examine everything from requirements to system design to test program. More system-like testing may be required to reduce the inferential gap. A system may have to be redesigned so as not to be vulnerable to an environment whose effect cannot be well-predicted computationally. The sort of framework proposed here provides a vehicle for addressing such fundamental issues.

#### d. Model Tuning.

When the analysis of prediction error data shows evidence of a bias, one can potentially either incorporate that bias into subsequent prediction error limits, in essence calibrating out the model's bias, or one can modify the model in an attempt to remove the bias. One mode of modification is to adjust the  $\phi$  parameters, which may often be uncertain estimates of, e.g., materials properties. Such 'tuning' can be suspect, but there are legitimate analyses that compensate for parameter estimation in characterizing the uncertainty of subsequent predictions.

Consider the case of a simple linear model,  $y^* = \alpha + \beta x$ . If an experiment is done at  $x_1$ , yielding  $y_1$ , then there are infinite ways to adjust  $\alpha$  and  $\beta$  to achieve perfect agreement between  $y^*$  and  $y_1$ . No rational statement could be made, however, about predictive capability for the adjusted model. If a second experiment is done at  $x_2$ , then a unique  $\alpha$  and  $\beta$  can be found to achieve perfect agreement at both points, but no statement about subsequent predictive capability can be made (obviously, a claim of perfect predictions is bogus). For three or more experiments, however, we can use standard statistical methods to estimate  $\alpha$  and  $\beta$  and characterize the prediction error for subsequent predictions based on these estimates. The example in the next section demonstrates this analysis. This sort of prediction-error analysis that accounts for tuning needs to be extended to the situation of more complex, higher-dimensional models.

For complex codes and corresponding experiments, one computation and one experiment can each yield thousands of data-values – traces of multiple response variables over time and space. There may be many parameters in  $\phi$  that could be adjusted to improve the agreement between computation and data. Even when there is a scientific basis for selecting the parameters on which to tune the computation, the residual errors over time and space after tuning to one experimental outcome do not contain any information about predictive capability. One could only

infer at best that: If another similar experiment were run and tuned, the resulting residual errors should look like the post-tuning errors obtained in the first experiment. One could not infer: If we used the tuned model to make a prediction in a similar experiment, the error of that prediction should be in line with the post-tuning errors we obtained in the initial experiment.

#### e. Distributional Predictions

A deterministic code calculates a prediction for a single, completely specified situation. Predictions of interest, though, are often 'statistical,' or distributional predictions, not single point predictions, as considered up to this point. For example, in a weapon systems context, systems are not identical and delivery and target conditions, such as impact angle, impact velocity, and target hardness, vary from mission to mission. In such situations the objective may be to predict the resulting probability distribution of some characteristic of weapon-performance, such as maximum shock on a key component, over some probability distribution of system variables and environmental conditions, and then to predict characteristics of this distribution. These characteristics could be the distribution's mean, its upper two-sigma point, or the probability of exceeding a failure threshold.

Suppose that  $x_r$ , a subset of the variables in  $x$ , is to be treated as random to obtain a distributional prediction. Suppose further, as a starting point, that the probability distribution of  $x_r$  is a given. Our objective is to estimate the resulting distribution of  $y$  and parameters associated with it. The statistical model specified above in (3) provides the means to do this, given appropriate experiments and data.

The law of total variance<sup>[8]</sup> says that

$$\text{var}(y) = \text{var}_x[E(y|x)] + E_x[\text{var}(y|x)], \quad (4)$$

where  $\text{var}(\cdot)$  denotes variance,  $E(\cdot)$  denotes expectation, and  $|$  denotes conditioning. The subscript indicates the random variable over which these moments are calculated. In words, (4) says that the unconditional variance of  $y$  is the sum of the variance of the conditional expectation of  $y$ , given  $x$ , and the expected value of the conditional variance of  $y$ , given  $x$ . Applying this relationship to the problem at hand leads to:

$$\text{var}_r(y) = \text{var}_r[y^*(x) + \delta_x] + E_r[\text{var}(e_x)], \quad (5)$$

where the subscript  $r$  denotes that the indicated variance or expectation is with respect to the distribution of  $x_r$ .

Suppose, to simplify things for this discussion, that  $\delta_x = 0$ , for all  $x$  in the  $x$ -region of interest. Then (5) becomes

$$\text{var}_r(y) = \text{var}_r(y^*) + E_r[\sigma_x^2]. \quad (6)$$

Propagation of the assumed distribution of  $x_r$  through  $M(x;\phi)$ , by methods such as Monte Carlo, provides an estimate of the first right-hand term in (6). Model-validation experiments and data analysis, if successful, provide an estimate of  $\sigma_x^2$ , as a function of  $x$ . The expectation of this

function with respect to the distribution of  $x_r$  could then be calculated or approximated to estimate the second right-hand term in (6). In the ideal situation in which  $\sigma_x$  is independent of  $x$  in the region of interest, the second right-hand term is simply  $\sigma^2$ , the variance of the difference between nature and computation. In either case I call  $\sigma_x^2$  the 'extra-model' variability. Similarly to (6), other functionals of the distribution of  $y$ , such as an exceedance probability, would have to be estimated by folding in the extra-model variability represented by the distribution of  $e_x$ .

Equation (6) shows that the role of the extra-model variability is not to provide bounds on the computational prediction, as was the case for point predictions. Rather, it is to add an additional variance component to the analysis; the effect of this addition is to inflate the variance one would get from propagation through the code. By itself, the code propagation variance, the first right-hand term in (6), underestimates the variance of nature's  $y$ , the left-hand term. If the code propagation variance,  $var_r(y^*)$ , was used as an estimate of nature's variation, then, e.g., failure probabilities would tend to be underestimated, sometimes drastically, as will be shown below, even if the model has been deemed valid via a hypothesis test. To obtain valid distributional predictions it is necessary to combine the estimated 'extra-model variability' with the estimated model-propagated variability.

Traditional code uncertainty-propagation analyses work the first right-hand term in (6), in various manifestations. Much research has been and continues to be conducted trying to wring out one more significant digit in approximations to this first term, all the while ignoring the second term (sometimes of necessity in situations in which meaningful model-validation experiments cannot be run). The only way to know whether the second term is ignorable is to run the model-validation experiments and perform the analyses to evaluate it. Estimating the second term and the bias function,  $\delta_x$ , should be the objective of model-validation programs. This is a much harder problem to work. It requires designing and running experiments, not just conducting computer exercises. It requires test facilities. It requires collaboration with experimentalists. It is messy. But it is necessary if credible measures of predictive capability are to be obtained. See Aeschliman and Oberkamp<sup>[9]</sup> for discussions and illustrations on this point in the context of fluid dynamics.

**Example.**

To illustrate the concepts and methods of prediction uncertainty quantification as applied to a linear model situation, I consider the model and a portion of the data considered by Hills and Trucano<sup>[10]</sup>. The situation of interest is the impact of a small aluminum pellet on an aluminum plate. Hills and Trucano use the CTH shock physics code<sup>[11]</sup> to predict shock wave velocity in the aluminum plate as a function of particle velocity, which is one-half the pellet's impact velocity.

To a good approximation, for the material and geometry considered and over the range of particle velocities of

interest, the CTH predictions are well-fitted by the linear model:

$$Us^* = 5263 + 1.368Up, \tag{7}$$

for  $Up$  between roughly 300 and 4000 m/s, where  $Us$  is shock wave velocity and  $Up$  is particle velocity, both in m/s. Suppose for the sake of illustration that we are interested in predictions in the neighborhood of  $Up = 3500$  m/s. At  $Up = 3500$  the model prediction is  $Us^* = 10,051$  m/s. What can we say about prediction-uncertainty bounds associated with this prediction?

For this illustration, I will use six of the 232 tests reported by Hills and Trucano as my illustrative model-validation experiments – three experiments near  $Up = 2000$ , three near  $Up = 3000$ . Limiting the amount of data is representative of the situation in which the cost of testing constrains the amount of testing that can be done. The experimental limits on  $Up$ , relative to the  $Up$  of interest, are representative of the situation in which available test facilities cannot achieve the application environment; thus extrapolation, as in Fig. 2, is required. The experimental results, the corresponding computational predictions, and the observed prediction errors are given in Table 1.

Table 1.  
Model-Validation Experiment Results, Predictions, and Prediction Errors  
(All values are in units of m/s)

Up	Us	Us*	Us – Us*
1957	8054	7940	114
1959	8015	7943	72
2095	8114	8129	-15
2987	9401	9349	52
3030	9177	9408	-231
3031	9180	9409	-229
		ave. =	-40
		RMS =	146

There is some evidence in Table 1 that the prediction errors are more negative and more variable in the neighborhood of  $Up = 3000$  (the last three data rows in Table 1) than they are near  $Up = 2000$  (the first three rows). However, with such limited data, these sorts of patterns are not too unlikely just by chance, so I will here illustrate the analysis of these data based on the assumption that prediction errors at specified values of  $Up$  are normally distributed with a mean of zero and a constant standard deviation  $\sigma$ , for  $Up$  ranging from 2000 – 3000 m/s. With this set-up,  $\sigma$  is estimated by the root mean square (RMS = square root of average squared-error) of the six observed errors, namely  $s = 146$  m/s. Other analyses of these data are provided by Easterling.<sup>[12]</sup>

Next I suppose that (experts say that) both the physics model and the statistical model extrapolate on  $Up$  from 3000 to 3500 m/s.

Before proceeding with the prediction error analysis, it is pertinent to discuss the possible sources of the observed prediction errors. One possible contributor is modeling

error that introduces bias in the CTH predictions and consequently to the linear approximation to these predictions. Though the limited data don't definitively indicate bias, there is an indication of bias in the pattern of errors. The full set of data, as shown in Hills and Trucano, confirms this bias. A second possibility is measurement error. The measured velocities, which are all we have to analyze, no doubt differ from the actual velocities. If we know or have a good estimate of the variance of measurement error, we can subtract it out of the observed prediction-error variance. A third possible contributor to prediction error is the effects of variables not included in the model. Not all aluminum is identical; the pellets and plates will vary dimensionally and compositionally; impact angles may vary from test to test and differ randomly from what is assumed in the calculations; surfaces are not perfectly smooth, etc. These sources of variability are not in the above simple linear model. In general, further analyses are required to quantify and eliminate sources of variability that are not pertinent to predictions of interest. On the other hand, we do not want to eliminate or underestimate sources of variability that would be present in an application for which predictions are desired.

#### a. Point Prediction

Under the assumptions that the linear model (7) predicts  $U_s$  without bias and that prediction errors at  $U_p$ 's in the 2000 – 3500 m/s range are normally distributed, with mean zero and a constant standard deviation,  $\sigma$ , which is estimated by  $s = 146$  m/s, on six df (degrees of freedom), various (statistical) inferences can be derived. For example, 95% prediction limits for a single future outcome are given by  $\pm t_{.025}(6) * s = \pm 2.447 * 146 = \pm 356$  m/s. Thus, at  $U_p = 3500$  m/s, the predicted shock wave velocity for a single future test, at the 95% confidence level, is  $U_s^* = 10,051 \pm 356$  m/s = (9695, 10,407) m/s.

Probably of more interest than a bound on single outcomes is a bound on the distribution of errors in future predictions. For example, an upper 95% confidence limit on the upper 99<sup>th</sup> percentile of the distribution of prediction errors, a limit that is termed an upper 95/99 statistical tolerance limit, is given by  $4.45 * s = 650$  m/s. Thus, at  $U_p = 3500$ , the inference is that with 95% confidence, 99% of the distribution of  $U_s$  would fall below 10,701 m/s. If, e.g., failure was defined as  $U_s > 11,000$  m/s, these results would tell us that there is good reason to conclude that the failure probability at  $U_p = 3500$  is less than .01, given the assumptions on which this inference is based. If the failure threshold was 10,500 m/s, the data do not support that strong of a conclusion, so further testing, or perhaps a redesign, might be required to achieve a .99 reliability with adequate confidence.

Other analyses by Easterling<sup>[12]</sup> consider a  $U_p$ -dependent error variance and a "tuned" model.

#### b. Probability Prediction.

Now, suppose for the sake of illustration that  $U_p$  in a pellet/plate impact scenario of interest is assumed to be random, with a Normal distribution with mean  $\mu_p = 3500$

m/s, and standard deviation  $\sigma_p = 100$  m/s. Suppose further that the failure threshold is  $U_s = 10,500$  m/s.

Propagating the assumed distribution of  $U_p$  through the approximate CTH model,  $U_s^* = 5263 + 1.368U_p$  (7), leads to the result that  $U_s^*$  is normally distributed with a mean of 10,051 m/s and a standard deviation of 136.8 m/s. Thus the failure limit of 10,500 m/s is 3.28 sigmas above the mean  $U_s$ , in which case, the predicted failure probability is .0005. This is just the sort of prediction that is developed from conventional code uncertainty analyses. For the case at hand, this prediction turns out to be quite optimistic; because of the substantial extra-model observed in the validation experiments.

When the variance of  $U_p$  is estimated as in (6) by adding the extra-model estimated error variance of  $\sigma_e^2 = 146^2$  m/s to the model-based variance, the total variance of  $U_p$  is estimated to 200<sup>2</sup>. Thus, the failure threshold is 2.24 standard deviations above the mean which corresponds to a predicted failure probability is about .013, nearly a factor of 30 times the model-based prediction of .0005. Accounting for the substantial uncertainty of the estimated  $\sigma_e$  yields an upper 95% confidence limit on the failure probability of .075, more than two orders of magnitude greater than the model-based estimated probability!

#### c. Measurement-Error Adjustment.

The preceding analyses are based on the conservative assumption that all of the observed extra-model variability was due to unmodeled variables and effects in the experiments. In fact, measurement variability is a component of the observed prediction errors and we would like not to include it in our prediction uncertainty quantification. Our interest is predicting actual  $y$ , not measured  $y$ .

If the measurement error variance is well-estimated, it can be subtracted from the total variance estimates to provide a more appropriate, less conservative, estimate of extra-model variability. For example, if the measurement error standard deviation is assumed to be 90 m/s, which is about 1% of the measured  $U_s$  values in Table 1, then, for the case in which the original model is assumed to be adequate, with a residual standard deviation estimate of 146 m/s, the adjusted estimate of the extra-model standard deviation (assuming measurement error is independent of  $U_p$ ) would be  $\sqrt{146^2 - 90^2} = 115$  m/s. With some modification, the preceding analyses could be carried out with this adjusted value and the results would be less conservative. If a 25 m/s standard deviation (again about 1% of the measured values) was assumed for measured  $U_p$  (the x-variable in the model), then this would translate into error in  $U_s^*$  with a standard deviation of  $1.368 * 25 = 34$  m/s, leading to a further adjustment to a prediction error standard deviation of  $\sqrt{115^2 - 34^2} = 110$  m/s. If there was reason to claim that total measurement error had a standard deviation of around 150 m/s, then measurement error accounts for all the observed prediction-error variability, so we could conclude that extra-model variability was negligible. We would be in the happy situation in which the model can be used as a surrogate for nature.

## Path Forward.

Implementing the general approach presented in this paper will be difficult. First, defining, then achieving an adequate and efficient set of experiments and computations for characterizing prediction error in the testable x-region will be difficult for complex phenomena and high-dimensional x. Next, extending what we learn about prediction error in testable situations to a quantification of prediction uncertainty in nontestable applications may be difficult or impossible in many applications. Solutions and work-arounds will have to be application-specific, but the general direction must be toward simplification – reduced dimensionality, reasonable approximations. Where solution is not presently possible, we will at least have a clear understanding of what the barrier is.

Implementing the proposed approach has substantial implications for both experimentalists and modelers. Both experimental facilities and computational models may have to be modified so that they are not only compatible, but synergistic. Again, solutions will have to be application-specific. Collaboration among experimentalists, modelers, and analysts is essential.

The path forward is to ‘just do it.’ General guidelines can be provided, but progress will come through implementation. By testing proposed methods on increasingly difficult problems, we will develop an understanding of these methods’ strengths and weaknesses.

## Conclusions

The primary points reached in this paper are:

1. The only way to measure ‘computational-prediction uncertainty’ is via suites of experiments and corresponding computations in testable environments and configurations.
2. A critical, subject matter-based inferential link is required to connect observed prediction errors in experimental contexts to bounds on prediction errors in untestable applications.
3. Model validation tests should be designed and conducted in ways that permit a realistic estimate of extra-model variability (prediction errors) in application environments.
4. Extra-model variability, which is estimated via model vs. nature experiments, should be included in probabilistic predictions. To omit this variability on the assumption of model-validity can lead to serious prediction error, particularly for reliability predictions.
5. There are trade-offs between model complexity and fidelity vs. model prediction-uncertainty “quantifiability” that need to be addressed in any particular application.
6. Adequate quantification of prediction errors, even in greatly simplified situations, can require a substantial number of experiments.

## Acknowledgment

This paper benefited greatly from extensive discussions with and careful review by Bill Oberkampf, Tim Trucano, Marty Pilch, Kevin Dowding, John Garcia, and Tom Paez (all Sandia) and Rich Hills (New Mexico State University).

## References

- [1] American Institute of Aeronautics and Astronautics. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, AIAA-G-077-1998.
- [2] Pilch, M., Trucano, T., Moya, J. L., Froehlich, G. Hodges, A., and Peercy, D. *Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format: Version 2.0*. Sandia National Laboratories Report SAND2000-3101, January, 2001.
- [3] Box, G. E. P. Robustness in the Strategy of Scientific Model Building, *Robustness in Statistics*, ed. by Launer, R. L., and Wilkinson, G. N., Academic Press, New York, 201-236 (1979).
- [4] Johnson, P.A. Comparison of Pier Scour Equations Using Field Data, *ASCE Journal of Hydraulic Engineering*, v. 121, 626-629 (1995).
- [5] Hahn, G. J., and Meeker, W. Q. *Statistical Intervals*, John Wiley & Sons, Inc., New York (1991).
- [6] Box, G. E. P., Hunter, W.G., and Hunter, J. S. *Statistics for Experimenters*, John Wiley and Sons, Inc., New York (1978).
- [7] Chiles, J. P., and Delfiner, P. *Geostatistics, Modeling Spatial Uncertainty*, John Wiley and Sons, Inc., (1999).
- [8] Parzen, E. *Stochastic Processes*, Holden-Day, San Francisco (1962).
- [9] Aeschliman, D. P., and Oberkampf, W. L. *Experimental Methodology for Computational Fluid Dynamics Code Validation*, Sandia National Laboratories report SAND95-1189, September 1997.
- [10] Hills, R. G., and Trucano, T. G. *Statistical Validation of Engineering and Scientific Models with Application to CTH*. Draft Sandia National Laboratories Report, September, 2000.
- [11] McGlaun, J. M., Thompson, S. L., and Elrick, M. G. CTH: A Three-Dimensional Shock Wave Physics Code, *Int. J. Impact Engng.* v. 10, 350-360. (1990).
- [12] Easterling, R. G. *Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations*, Sandia National Laboratories Report SAND2001-0243, February, 2001.

---

<sup>1</sup> Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL8500.