SANDIA REPORT
SAND2001-0243
Unlimited Release
Printed February 2001

Measuring the Predictive Capability of Computational Methods: Principles and Methods, Issues and Illustrations

Robert G. Easterling

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2001 – 0243
Unlimited Release
Printed February 2001

Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations

Robert G. Easterling
Nuclear Weapons Program Integration and Studies Center
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0417

Abstract

It is critically important, for the sake of credible computational predictions, that model-validation experiments be designed, conducted, and analyzed in ways that provide for measuring predictive capability. I first develop a conceptual framework for designing and conducting a suite of physical experiments and calculations (ranging from phenomenological to integral levels), then analyzing the results first to (statistically) measure predictive capability in the experimental situations then to provide a basis for inferring the uncertainty of a computational-model prediction of system or component performance in an application environment or configuration that cannot or will not be tested. Several attendant issues are discussed in general, then illustrated via a simple linear model and a shock physics example. The primary messages I wish to convey are:

1. The only way to measure predictive capability is via suites of experiments and corresponding computations in testable environments and configurations.
2. Any measurement of predictive capability is a function of experimental data and hence is statistical in nature.
3. A critical inferential link is required to connect observed prediction errors in experimental contexts to bounds on prediction errors in untested applications. Such a connection may require extrapolating both the computational model and the observed extra-model variability (the prediction errors: nature minus model).
4. Model validation is not binary. Passing a validation test does not mean that the model can be used as a surrogate for nature.
5. Model validation experiments should be designed and conducted in ways that permit a realistic estimate of prediction errors, or extra-model variability, in application environments.
6. Code uncertainty-propagation analyses do not (and cannot) characterize prediction error (nature vs. computational prediction).
7. There are trade-offs between model complexity and the ability to measure a computer model's predictive capability that need to be addressed in any particular application.
8. Adequate quantification of predictive capability, even in greatly simplified situations, can require a substantial number of model-validation experiments.

Table of Contents

Executive Summary 6
Introduction 6
Framework 6
Statistical Model 7
Conclusions 8
Path Forward 12
Introduction
Introduction 13
Problem Statement and Schematic 14
Terminology 16
Contrast with Test-Based Inference 17
Mathematical Representation 17
Issues and Discussion 21
Introduction 21
Model Validation - The Concept 21
Model Validation as Hypothesis Testing 21
Experimental Design 23
Experimental Objectives 24
Constraints 26
Experiment-Model Compatibility 27
Data Analysis 28
Metrics 28
Choice of Analysis Variables 28
Inference 29
Distributional Predictions 30
Code Uncertainty Propagation 32
Estimation Uncertainty 33
Diagnostic Analyses 33
Model Tuning 34
Integrating Suites of Experiments and Calculations 35
Summary 36
Illustration: Linear Model 37
Introduction 37
Prediction Uncertainty Quantification: Point Prediction 37
Example 38
Test Results 39
Theoretical Model, Homogeneous Errors 40
Theoretical Model, Nonhomogeneous Errors 41
Tuned Model 42
Tuned Model; Homogeneous Errors 42
Tuned Model; Nonhomogeneous Errors 43
Predicting a Probability 44
Theoretical Model 44
Tuned Model 46
Example Continued: Probability Prediction 46

Theoretical Model 46
Tuned Model 47
Analysis: Unmeasured x 48
Measurement-Error Adjustment 50
Comparison to Strictly Test-Based Prediction 51
Where's the Inference: 51
Integration 52
Experimental Design and Statistical Power 53
Summary 55
Concluding Comments 56
Conclusions 56
Programmatic Implications 56
References 57

List of Tables

Table 1. Model Validation Test Results, Predictions, and Prediction Errors 39
Table 2. Model Validation Test Results, Predictions, and Prediction Errors: Nominal Predictions 49
Table 3. Summary of Analysis: Unmeasured Up 49

List of Figures

Figure S-1. Quantifying the Uncertainty of Computational Predictions 7
Figure S-2. The Inference Problem 9
Figure 1. Quantifying the Uncertainty of Computational Predictions 15
Figure 2. The Inference Problem 24
Figure 3. Predictions and 95% Prediction Limits: Theoretical Model, Assumed Homogeneous Variance 41
Figure 4. Predictions and 95% Prediction Limits: Tuned-Model, Assumed Homogeneous Variance 43
Figure 5. Actual Failure Prob. Vs. Predicted 46

Acknowledgment

This report benefited greatly from extensive discussions with and careful review by Bill Oberkampf, Tim Trucano, Marty Pilch, Kevin Dowding, and Tom Paez (all Sandia) and Rich Hills (New Mexico State University). My sincere appreciation also goes to the Sandia's Nuclear Weapons Program Integration and Studies Center (9800) which provided me the opportunity and support to pursue this study.

Executive Summary

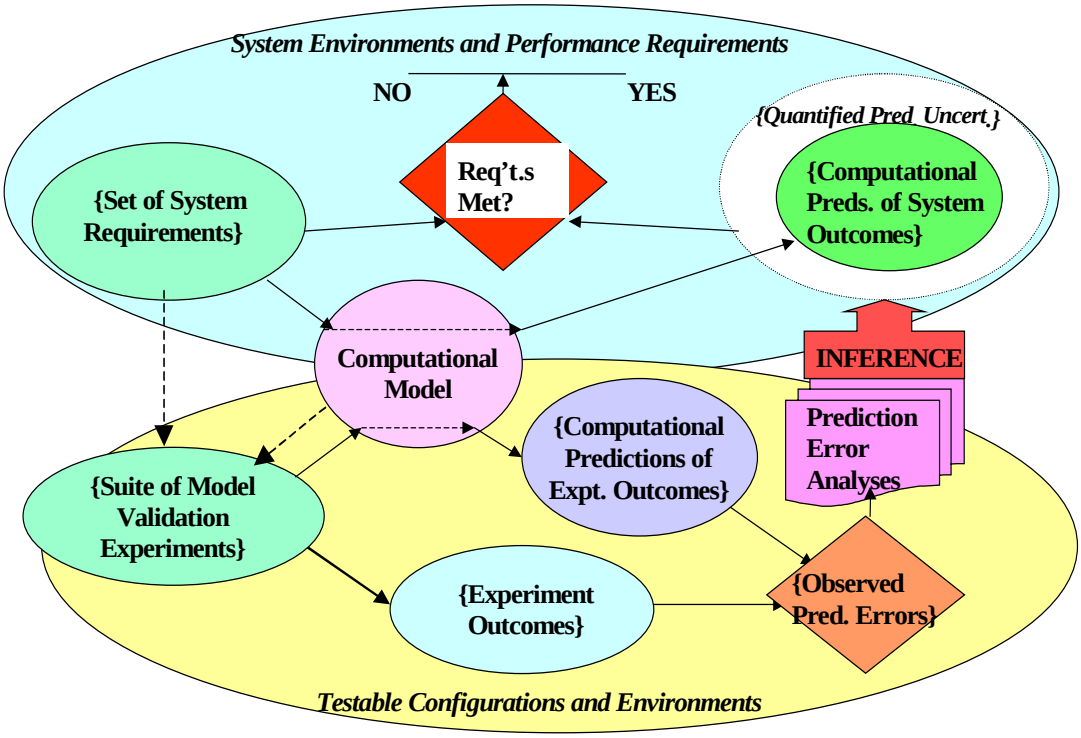
Introduction. Users of computational predictions, from designers to decision-makers, need to be provided with information on how accurate the prediction is and on what basis. E.g., "Based on our understanding of the underlying physics, our ability to translate that understanding to a computational code, 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 20%." Such prediction uncertainty limits define predictive capability and provide the necessary yardstick against which a computational prediction can be compared to a requirement. In this report I propose and illustrate methods for the determination of such limits and identify and discuss attendant issues.

Framework. Confidence in computational predictions comes (in large part) from comparisons with data. The term model-validation 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.

A common view of the model-validation process is that the outcome is binary – either the computational model is validated or it is not (for specific applications). More important, however, is the fact that these comparisons provide the raw material for determining bounds on prediction error for predictions in situations that cannot or will not be tested. In fact, I argue that the full purpose of model-validation experiments should be the measurement of predictive capability. Figure S-1 is my view of the whole process, set in the context of comparing a computational prediction to a system requirement.

The top ellipse in Fig. S-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 the specified environments. Comparing the prediction to the requirement requires an uncertainty yardstick, 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 requirements 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 upper ellipse.

Figure S-1. Quantifying the Uncertainty of Computational Predictions



Statistical Model. All computational models, no matter how extensively the underlying processes are modeled, are approximations to nature. They are based on assumed homogeneities and symmetries that do not necessarily hold in nature. To mathematically describe the relationship between computation and nature, let x be a vector of variables in the computational model that represent the item being tested and the environment to which it is subjected. These variables can be thought of as code inputs. Let y*(x) denote the computational result at x. Next let y(x) denote the corresponding outcome of an experiment conducted at x. I represent the relationship between y(x) and y*(x) by the following statistical model:

$$y(x) = y^*(x) + e_x, \qquad \text{(S1)}$$

where e_x is a random variable with an unknown probability distribution that possibly depends on x. At any particular x-point, various aspects of the computational model can introduce bias in e_x and unmodeled effects and variables, not controlled in the experiment, introduce variability. Both the mean and variance of e_x can change from point to point in the x-space.

With this set-up, measuring predictive capability becomes first the characterization (estimation) of the probability distribution of e_x at the x-points at which computations and

experiments are conducted, then the estimation of the distribution of e_x at x-points pertaining to system applications. From this estimated distribution one can (statistically) bound the difference between computational prediction and nature. In general, estimates of the probability distribution of e_x will be based on limited data, so estimation imprecision will be appreciable. Our goal is both to measure predictive capability and to measure the imprecision with which that measurement is done. (Note: What we have to work with in practice are measurements of nature, so observed e_x contains measurement error. It is possible to remove this source of variation in quantifying prediction uncertainty.)

Conclusions. Putting the concepts and processes represented in Fig. S-1 into practice can be a major undertaking. The body of this report addresses issues that arise and sets forth methods and approaches that can be used. A general discussion is followed by the application of the proposed methods in the specific case of a linear model, which is illustrated in this report by validation experiments for the CTH shock physics code. The primary messages I wish to convey about measuring the predictive capability of computational models are the following.

1. **The only way to measure predictive capability is via suites of experiments and corresponding computations in testable environments and configurations.**

Once it is realized that the uncertainty of importance to users of computational predictions is the (unknown) difference between nature and computation, this conclusion is obvious. You have to have data which means that you have to conduct or to have conducted experiments or tests that provide nature the opportunity to function and provide good data to compare to computational predictions. As discussed below (conclusion 6), there are other popular varieties of uncertainty quantification that are strictly analytical -- they are code-based and require no physical experimentation -- so it is important to stress the necessity of experimentation in quantifying *prediction* uncertainty or *predictive* capability.

2. **Any measurement of predictive capability is a function of experimental data and hence is statistical in nature.**

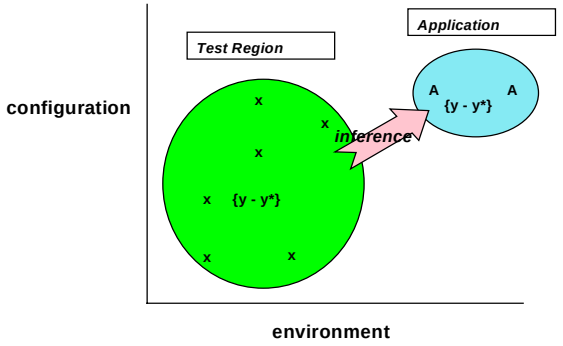
Experimental data are stochastic in nature because of instrumentation measurement error and because of the effects of uncontrolled sources of variation in the experiment. Thus, any data-based measure of predictive capability, such as the standard deviation of the observed prediction errors, is a statistical estimate. To make proper use of this estimate we need to know something of its 'reliability' (accuracy and precision). E.g., in conventional situations a sample standard deviation is characterized by its "degrees of freedom" (which reflects the amount of data going into the estimate) and bounds calculated from the standard deviation are a function of the degrees of freedom on which it is based. This relationship between a measure of predictive capability and its reliability is also important in designing experiments and in deciding when enough experimentation has been done.

3. A critical inferential link is required to connect observed prediction errors in experimental contexts to bounds on prediction errors in untested applications. Such a connection may require extrapolating both the computational model and the observed extra-model variability (the prediction errors: nature minus model).

The general inference problem is illustrated in Fig. S-2 in which the space in which experiments and applications exist are defined by two meta-variables, configuration and environment. Because of treaty, regulatory, or economic reasons it may not be possible to test hardware configurations in their required environments. (For this reason, I depict an extrapolation situation; interpolation should be easier.) Thus, we have to extend what we can learn about predictive capability (represented by the prediction errors, {y-y*}, in Fig. S-2) where we can evaluate it to an inference about predictive capability where we cannot. Note that the final 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 spaced along and between the axes that will provide the data necessary to make such extensions. Statistical data analysis methods are required to evaluate the uncertainty associated with such extensions. Statistical design of experiments ideas are required to assure an adequate and efficient basis for these extensions.

It must be recognized that a 'hard' inferential link is not always achievable. The inference may rest on untestable assumptions. The scientific basis or the data may be lacking. There may be no clear way to merge information on the predictive capability, e.g., in separate single-phenomena contexts to an inference about multiple-phenomena predictive capability. The remedy to this problem may be heavier reliance on testing and testing at a higher level in order to provide confidence that requirements are met. Graphically, this means shortening the inferential arrow in Fig. S-2. The resumption of underground nuclear tests would be an extreme example. Absent an expansion of testing, softer, expert opinion-based inferences may be required. Confidence becomes more elusive in this arena.

Figure S-2. The Inference Problem



4. Model validation is not binary. Passing a validation test does not mean that the model can be used as a surrogate for nature.

In terms of the statistical model above (eq. S1), model-validation often amounts to a test of the hypothesis that the expected value of e_x is zero, either at single or multiple x-points. Hypothesis tests have a binary outcome – accept or reject. But accepting the validity hypothesis does not mean that prediction error is negligible. In fact, the noisier y – y* is, the more likely the hypothesis is to be accepted. That is, the worse the model's predictive capability, the greater the chance of concluding that it provides 'valid' predictions. This property invalidates treating model-validation as a hypothesis test.

Further, a conclusion of negligible prediction error (in terms of both bias and variance) in the validation experiments does not imply negligible prediction error at untested x-points. The inference link in Fig. S-2 must be bridged, both in terms of predicting performance and in characterizing prediction error.

Model-validation should be thought of as an estimation problem – estimating the prediction error distribution in an application – rather than a hypothesis-testing problem – zero expected error for testable situations. This distinction has major implications for determining the nature and extent of validation experiments.

5. Model validation tests should be designed and conducted in ways that permit a realistic estimate of prediction errors, or extra-model variability, in application environments.

Having the goal of Fig. S-1, which is the quantification of uncertainty for computational predictions in untested system applications, provides a necessary focus for the model-validation effort. While tightly controlled, 'clean' experiments may be appropriate in early stages of model-validation, they may not provide an adequate basis for inferring prediction uncertainty in much less controlled situations. The experimental goal is not just to show that the code is doing approximately the right thing, but to also get the error structure right. This means giving the unmodeled variables and effects in e_x the chance to vary in an experiment as they would in the application of interest. Alternatively, one would need a means of extrapolation from lab-scale errors to application-scale errors. This could be achieved if e_x were characterized as a function of variables that are controlled in the experiment. Then, the estimated variability of these variables in the application could be propagated through the model for e_x in order to estimate the effects of uncontrolled variables in the application.

6. Code uncertainty-propagation analyses do not (and cannot) characterize prediction error (nature vs. computational prediction).

Code uncertainty-propagation analyses, which conventionally involve the propagation of assumed input probability distributions through a computational model, address one or both of the following:

1. the transmission of assumed stochastic variability in x into variability of y*

2. the transmission of uncertainty about assumed or estimated constants in the model, such as material properties, transfer coefficients, or equations of state, into uncertainty about y*.

(In some discussions, these two analyses correspond to treatments of irreducible and reducible uncertainties.) These analyses are conducted via the computational model and thus are conditional on the model and cannot measure the difference between computational prediction and nature. The first analysis estimates only the variability of the computational output variable, y*, not the variability of nature's y. It neglects the extra-model variability, represented by e_x, and thus can be misleadingly optimistic if interpreted as an estimate of nature's variability. The second analysis can provide bounds on a calculation based on unknown 'true values' of physical constants – but not bounds on nature. These uncertainty propagation analyses provide valid statements about nature only if it can be assumed that the extra-model variability is negligible. Justifying this assumption requires the experimentation and data comparisons discussed in this report. A fortunate result is the finding that e_x is negligible; it cannot be assumed to be negligible a priori.

7. There are trade-offs between model complexity and fidelity vs. model prediction-uncertainty quantifiablity that need to be addressed in any particular application.

The suite of model-validation experiments in Fig. S-1 consists of experiments at a selected set of x-points, as in Fig. S-2. Covering the x-space, in some sense, is required to provide a basis for inference about prediction error at untested x-points. Thus, the higher the dimension of x, the more experimentation is generally required. Furthermore, if x, the modeler's set of variables, contains variables that cannot be measured or controlled in an experiment, the job of characterizing predictive capability becomes more difficult and predictions become more uncertain. Compatibility of computational model and experiment is a goal of both model design and experimental design. Model simplification, which primarily means reducing the dimensionality of x and capturing the effect of the set-aside x's experimentally, via e_x, will alleviate these problems. Just as design for testability is a criterion for hardware design, design for validation should be a criterion for computational model design. Collaboration among modelers, experimentalists, and analysts is clearly called for in order to make this process work efficiently and effectively.

8. Adequate quantification of prediction errors, even in greatly simplified situations, can require a substantial number of experiments.

In the illustrative example in this report, six model-validation experiments are used to estimate the prediction error distribution in the case of a linear model in a single x-variable. The analysis shows that the resulting inferences are quite uncertain. Such uncertainty can be tolerable when adequate margin exists. Also, the limited amount of data in the example does not unambiguously detect model bias that is quite evident in the

larger set of data from which the six points were obtained. The expanded goal of measuring predictive capability, on which this report is based, requires more data, in general, than conventional model-validation, which often boils down to overlaying experimental and computational results from a small number of experiments.

Path Forward. Implementing the general approach presented in this report 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 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 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.

Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations

Introduction

Introduction. The ability to make credible declarations about the ability of nuclear weapons to meet their requirements in normal, abnormal, and hostile environments is key to successful stockpile stewardship and management. Historically credibility, or confidence, in these declarations has been derived in large part from a program of realistic testing – both in development and in stockpile surveillance. Currently, improved computational capabilities offer the opportunity for computational predictions of weapon performance to take on a portion of the confidence burden by supplanting, where necessary and appropriate, or supplementing, physical testing. Improved computational tools also have the potential to reduce design, development, and manufacturing time and cost by providing better predictions and more complete explorations of pertinent parameter spaces, thereby reducing the number of design-build-test cycles required to advance from concept to stockpile weapon. In fact, it is probably the case that improved computational tools will 'earn their spurs' as design tools as a preliminary to gaining acceptance as a system certification tool. Further, it should be noted that although nuclear weapons applications and science-based stockpile stewardship are the motivation for this work, there is a broad need to establish the credibility of computational predictions in many areas of application.

Alert decision-makers and others affected by computational predictions will ask: How much error might there be in the prediction (relative to the physical event being approximated by the computation)? and, How do you know? Answers to these questions generally have to come from comparisons of computations to data, where such comparisons can be made. Good answers, I claim, will come from applying methods of statistical design of experiments and statistical data analysis to complement the scientific and computational knowledge underlying the computational models.

The need to establish credibility in computational predictions is recognized by Sandia's requirement of formal verification and validation plans for ASCI (Accelerated Strategic Computing Initiative) codes (Pilch et al. 2001) and by the experiments and analyses sponsored by Sandia's MAVEN (Model Accreditation Via Experimental Sciences for Nuclear Weapons) program (Moya 1998). Model validation requires the computational prediction of outcomes for situations in which physical experiments can also be performed. The comparison of computational predictions to experiment outcomes over some meaningful suite of experiments, if satisfactory, provides confidence in the predictive capability of the computational model, at least where such comparisons have been made. It should be clear, though, that "V&V (verification and validation) processes do not [permit one to] directly make claims about the accuracy of *predictions*" (AIAA 1998, my emphasis). That is, measures of predictive capability, for situations in which

such measures can be obtained, do not necessarily or automatically apply to predictions made for other situations. But, it is precisely these 'other situations,' wherein testing is precluded or prohibitively expensive, for which we most critically need to be able to say something about the accuracy of a computational prediction. The hope, which this report pursues, is that the right set of model-validation experiments, the science-based linkage between the experimental conditions and the conditions for which subsequent predictions are required, and the appropriate analysis of the validation results will provide useful insight into the accuracy of these predictions.

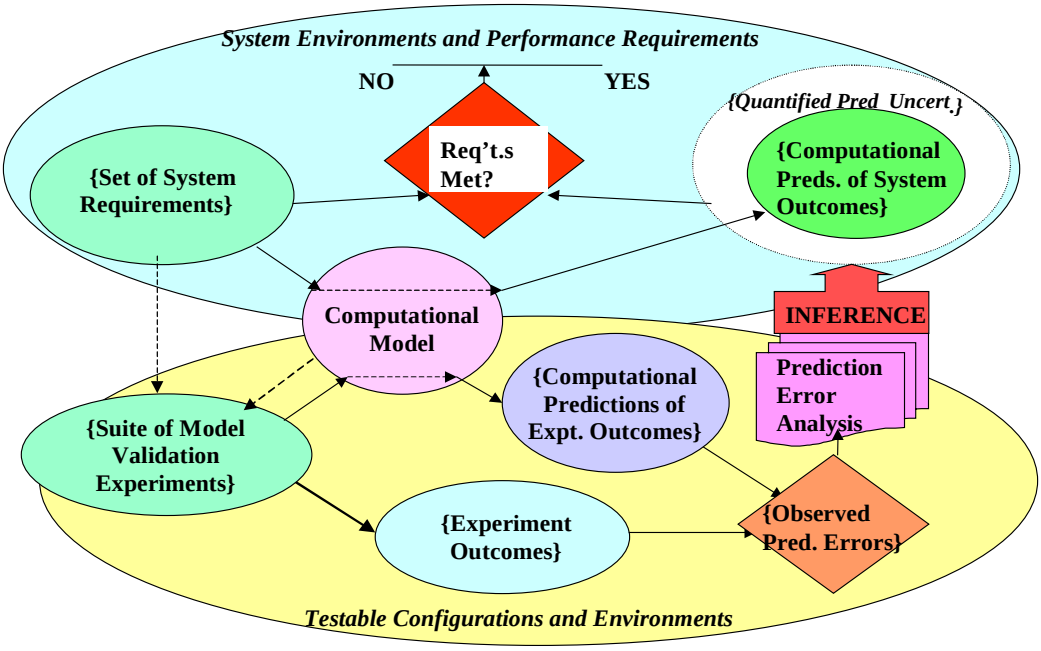
Thus, this report expands the purpose of model-validation experiments to include estimating bounds on the uncertainty associated with computational predictions. The goal of the analysis is to credibly make statements such as: "Based on our understanding of the underlying physics, our ability to translate that understanding into a computational code, and our comparative 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 20%." Such statements of predictive capability permit us to claim, e.g., that if the computational prediction plus P% is less than the failure threshold, we can confidently declare that the system meets requirements. (Terminology note. The general problem addressed in this report is the measurement of predictive capability; one particular measurement is a bound on prediction uncertainty.) My ultimate research goal, in conjunction with other Sandians working in this area, is to characterize the prerequisites and develop methodology for making such statements, including technical definitions of 'quite likely' and 'confidently declare' and derivation of the error-limit, P%. This report illustrates both issues and methods for measuring predictive capability by application to a simple physics model.

Problem Statement and Schematic. Figure 1 displays the basic elements of the problem of measuring the predictive capability of computational models of weapon performance. It also depicts the subsequent comparison of a computational prediction to a requirement against a backdrop of quantified prediction uncertainty. It should be recognized, though, that decisions such as the certification of nuclear weapon systems or components, will in general be based on a mix of computations and tests and will not be strictly 'computation-based.' Defining the mix is a matter for subsequent research.

The top large ellipse in Fig. 1 depicts the problem of interest: A weapon has performance requirements to meet. Within this overall context, particular applications are of interest, such as a particular class of accidents, e.g., crash-and-burn, and the requirement is a specified low probability of accidental detonation in these scenarios. A computational model will be used to predict the outcomes of these scenarios. Thus, the upper arrow passing through the computational model can represent a suite of computations, as, for example, is done in a Monte Carlo analysis, not just a single one. That is, the model itself is deterministic, but it can be nested in a stochastic shell. It can also be run at various conditions to explore a requirement space. For simplicity, I first discuss single predictions.

To judge these computational predictions against requirements requires the frame of reference of a prediction uncertainty limit. To state the problem mathematically, nature's unknown outcome in a particular application will be y. The computational prediction is y*. Our goal is to credibly estimate, or infer, limits on the magnitude of the unknown difference, y – y*. These limits, depicted by the uncertainty cloud around the computational prediction in Fig. 1, provide a frame of reference against which to judge y* vs. the system requirement, as in the above example.

Figure 1. Quantifying the Uncertainty of Computational Predictions



The only way to gauge the difference between nature and computational prediction is to conduct experiments in testable configurations and environments and measure the differences between nature and computational predictions in these conditions. The large bottom ellipse in Fig. 1 depicts this process. To generate this database of differences a suite of experiments will be designed and conducted, possibly ranging from single phenomenon to system level tests, from sub-scale to full-scale, and in single or combined environments. The design of these experiments should be driven by the system requirements and the computational model structure, as indicated by the dashed arrows in Fig. 1.

For a given experiment, (1) the computational model can be configured and run to predict the outcome and (2) the experiment can be carried out and its outcome(s) observed**.** (In some situations it may be appropriate to run the experiment first to obtain data for the initial and boundary conditions that are required as input to the computation. Such a protocol would be appropriate for cases in which system requirements were stated in terms of initial and boundary conditions. For example: the system shall safely survive a fire characterized by the following conditions: … .) Then, the computational and

experimental outcomes are compared to each other, as is represented in Fig. 1 by the set of observed prediction errors, {y – y*}, represented by the diamond in Fig. 1. Typical current model-validation analyses would end with a qualitative assessment of the goodness of the agreement. To reiterate, the goal here is to go further, to synthesize these data into a (statistical) statement about prediction error associated with computational predictions in the application arena. It is not enough to get good agreement in a model-validation exercise. That effort needs to provide the building blocks from which to infer predictive capability in untested situations.

[Note. An experiment actually provides only measured y, not nature's y. This distinction and its consideration in the analysis of the observed differences, {y – y*}, will be discussed below. At this stage of the discussion, I treat measured and true y interchangeably.]

The process by which we extend what we learn about prediction error in the suite of model-validation calculations and experiments to quantifying the uncertainty in computational predictions for applications of interest is represented in Fig. 1 by the prediction-error analysis. This analysis results in an inference that connects the upper and lower ellipses. That is, it is the analysis of the observed differences between computational predictions and experimental outcomes in the model-validation arena that provides the basis for quantifying the uncertainty associated with computational predictions in the system applications arena. Making this process work for the complex computational models used to predict weapon performance will be difficult. Such inference requires both a scientific basis and statistical inference methods adapted to the situation. Litmus tests need to be developed to define when the inference connection can and cannot be made. For example, if only single-phenomenon model-validation experiments are conducted, it may be impossible to credibly quantify the error associated with combined-phenomena computational predictions.

When the prediction uncertainty analysis can be successfully and believably carried out, the analysis output, as indicated by the inference arrow, is prediction uncertainty limits pertaining to the computational prediction of weapon application outcomes. As noted, this output depends critically on the scope, nature, and information content of the suite of experiments. For example, the more closely experiments can be made to resemble weapon applications, the 'shorter' the inferential arrow becomes, but the tests become more complex, costly, and difficult to instrument. Thus, there are trade-offs to analyze and resolve.

Terminology. Several terms used in the preceding discussion have different interpretations in different contexts, so it is necessary to define my usage. What I call the observed prediction errors, {y – y*}, have a variety of sources, or contributors, some of which are "uncertainties" and some of which are "errors," in the terminology of the AIAA Guide (AIAA 1998). The primary AIAA distinction is whether modeling-activity deficiencies are due to lack of knowledge (uncertainty) or not (error). The metrology and experimental communities would characterize the effects of various sources of the observed prediction errors as random and systematic errors (but not necessarily in a 1-1 correspondence with the AIAA's uncertainties and errors). Uncertainty would refer to

limits calculated to express the magnitude of these errors. On the other hand, outside of these particular fields, uncertainty is loosely used as a catch-all term that includes stochastic variability, the imprecision of estimated constants, and bias. Where it is tedious to be more explicit, I will succumb to this usage. I will, however, use the term, prediction-uncertainty limits to pertain to limits on computational predictions for untested situations. The analysis leading to these limits, which is comprised of the analysis of observed prediction errors plus the subsequent inference, will be called a prediction-uncertainty analysis. I also distinguish between prediction-uncertainty analysis, which deals with data, and code uncertainty-propagation analysis, which deals with propagating assumed probability distributions through a computational model. The general problem addressed in this report, as signaled by its title, is the problem of measuring predictive capability, one such measure of which is prediction uncertainty limits.

Contrast with Test-Based Inference. Decision-making based on traditional testing also rests on a prediction: the system will perform in a real-world application in the same way it performed in a test configuration and under test conditions. (Alternatively, if the test is an over-test, the prediction is that the system will perform no worse than it did in the test.) One relies on the ability of cognizant parties to scrutinize the test set-up, execution, and results, then render a decision on whether the tests are legitimate tests of the system's ability to meet requirements; any uncertainty pertaining to this inference is assumed to be negligible. More problematical is inference from test results at a few points in the requirement space to the conclusion that requirements are met throughout the space. This problem is particularly acute for abnormal environments where few tests can be done in what is essentially an unbounded space. Thus, test-based prediction and decision-making also involve uncertain inferences, but these are not usually explicitly quantified. (One exception: statistical inference may be involved as when we infer, with 95% confidence, e.g., that the number of defective units in a production lot is no more than D, based on random sample results of d defectives in n tests.) Both a computer model and a physical test are models – approximations to reality – so both require an inference link to use conditions and requirements – to reality. The linkage for test-based inference is often more transparent, but subtleties and unwarranted assumptions can contaminate inferences in this case also.

Mathematical Representation. For the discussion in the next section of further issues pertaining to Fig. 1, I expand the previous mathematical depiction of model-validation experiments and computations as follows:

Computation. (1a)

$y^*(x) = M(x:\phi)$, where

$M(x:\phi)$ is the computational model of the phenomenon of interest,

x = model input

ϕ = model parameters

y^* = model output, a prediction of a characteristic, y

Nature. (1b)

$y(x) = N(x,w:\varphi)$, where

$N(x,w:\varphi)$ is nature's function (unknown),

w = additional variables that influence nature's outcome,

φ = nature's parameters

y = nature's outcome at x,w.

All the quantities, y, x, w, ϕ, and φ, are possibly vectors or fields. The w's are in general unknown or unmeasured, so they are not shown as arguments of y(x). The following paragraphs elaborate on this representation of computational and nature outcomes.

First, with respect to the computation, the model's input vector x describes, in general, a physical entity and the environment to which it is subjected. This vector will include dimensions, materials, and environmental variables and will also include initial and boundary conditions. The computer model, $M(x:\phi)$, is an operator that transforms input x into the predicted result, y*. This is a deterministic calculation (though, as discussed later, multiple calculations might be done in a stochastic simulation). The governing equations in the model include parameters (user-specified constants), such as material properties, transfer coefficients, and equations of state. These parameters, denoted by ϕ, are part of the definition of M, but I express them explicitly because imprecision in their specification is a contributor to prediction uncertainty that may need to be addressed. The variables, x, and parameters, ϕ, can be linked. For example, an x-variable might be a material and a corresponding ϕ-parameter might be a property of that material such as tensile strength. In short, the x-variables describe the situation being simulated; the ϕ-parameters are part of the mathematical machinery by which that situation is simulated.

I further assume that all numerical aspects of M, such as grid size, time steps, and convergence criteria, are fully specified and thus also part of the definition of M (see Oberkampf et al. 1999 for a discussion of the effects of code numerics). For this discussion, I also assume that the code has been verified – it is validation-ready. Model-validation, however, is a process, as is model-building. It may be the case that early cycles through the validation process may actually be late cycles in the model-building process. For example, model-validation results could lead to changes in the values of the parameters, ϕ, used in the model.

Now, consider an experiment conducted at a specified x. That is, we subject a physical entity to the environment, as jointly defined by x, and observe the outcome, y(x). (Note the underlying assumption that the x-variables in the computational model are meaningful in defining an experiment. Achieving this compatibility requires modeler/experimentalist collaboration.) Nature's outcome at x will generally be influenced by variables beyond what the modeler has chosen. These are the w's in the

above representation of nature's outcome. For example, the w's may represent three-dimensional characteristics of a physical entity, while the computational model is two-dimensional based on an assumption of symmetry. Because of the unmodeled variables or effects present in nature, the function, $N(x,w:\varphi)$, is consequently different from $M(x:\phi)$ and might be different even if there were no w's. For example, nature may be nonlinear in x where linearity is assumed in $M(x:\phi)$. Also, nature's relationship might not even involve some x's the modeler has chosen. The two sets of model parameters, ϕ and φ, could be disjoint or overlapping. Where they overlap, in terms of their definition, e.g., the tensile strength of a particular metal alloy, they could have different values.

"All models are wrong, but some are useful," (Box 1979) is a statement by George Box, University of Wisconsin, that succinctly captures the essence of the question of confidence in computational predictions. All models are simplifications of or approximations to nature, thus 'wrong.' The different functions, $N(x, w:\varphi)$ and $M(x:\phi)$, represent this difference. Useful models are those for which the prediction error, the difference between nature and computation, is tolerable in the context in which the model is to be used. The problem is how to establish 'usefulness."

Statistical Characterization of Prediction Error. Conceptually, in repeated experiments at a fixed x, as nature's w's are allowed to vary randomly over these replications, a probability distribution of nature's outcomes would be generated at that x. Consequently, a probability distribution of differences between nature and the fixed computational prediction, i.e., a distribution of prediction errors, is also generated. Thus, my approach to model-validation experimental design and data analysis and the subsequent quantification of prediction uncertainty will be via the statistical model:

$$y(x) = y^*(x) + e_x, \qquad (2)$$

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

That is, in (2) nature's outcome at the situation described by x is modeled as the sum of the deterministic prediction at x plus a random error. This error has systematic and random components. The random component is the cumulative effect of nature's w's not included in the computational model but allowed to vary in the experiments. Functional differences between nature and computational model will be manifested as bias in e_x. This bias could depend on x. Similarly, the variability of e_x, in both its nature and its magnitude, could also depend on x. For example, one might expect prediction to become more difficult (uncertain) as the situation represented by x becomes more complex.

One further aspect of the model is to recognize that nature's outcome at x is observed or measured with error, so in all that follows I will let y(x) in (2) denote the measured experimental outcome. Thus, y(x) and $y^*(x)$ are both observable. Below, I discuss backing out the measurement error variance from an estimate of overall variance of prediction error, but for simplicity at this point, just assume that measurement error is negligible. Leaving it in is conservative in terms of leading to larger prediction uncertainties than if it is removed.

The conventional statistical approach of (2) also has engineering precedent. For example, in bridge design, civil engineers have developed a mathematical model for 'scour' – the erosion of soil around a bridge's foundation due to flooding (Johnson 1995). This model is a function of soil type, river-flow volume and velocity, and other pertinent variables. For predictions bridge analysts incorporate an additional 'modeling factor' (multiplicative in the case of the scour model, which is additive on a log scale) to represent the deviation of actual scour depths from the theoretical. This modeling factor is e_x in (2).

A key aspect of the statistical model (2) and the analysis based on it is that the probability distribution of prediction error is unknown. It's not something we can assume or analytically derive from ad hoc assumptions. We will have to estimate aspects of the error distribution based on possibly quite limited data from the error distributions at different x-points. Thus, as is generally the case for any measurement, measuring predictive capability is "fundamentally … approximate and … nondeterministic" (Trucano 2000), that is to say, fundamentally statistical.

In terms of (2), the predictive-capability measurement problem is first the estimation of the probability distribution of e_x at the x-points at which computations and experiments are conducted, then the estimation of the distribution of e_x at x-points pertaining to physical entities and environments, such as a system subjected to a threat environment, that cannot or will not be tested. From this estimated distribution, when it can be obtained, one can (statistically) bound the difference between computational prediction and nature. Furthermore, because the measurement of predictive capability must be an estimate, one should also characterize the "reliability" of the estimate. This is the essence of statistical analysis – to estimate quantities of interest and to characterize the precision of that estimate. As more well-designed model-validation experiments are conducted and analyzed, the precision with which predictive capability is known should improve, whether or not estimated prediction capability itself improves. The goal of the analysis must be to capture both aspects of prediction capability.

Note further that while model-validation is often motivated by a desire 'to get the physics right,' for the objective of predictive-capability measurement, the desire should be 'to get the error distribution right.' This contrasting goal has major implications for the design of model-validation experiments.

It is convenient to characterize the prediction-error distribution by its mean, say δ_x, and its standard deviation, σ_x. Ideally, prediction error would be normally distributed, δ_x would be zero, and σ_x would be usefully small, uniformly over x. This is not likely to be the situation. Both δ_x and σ_x could depend on x in complex ways. Model-validation experiments need to be conducted at a set of x-points and sized to permit this dependence to be characterized. One begins to sense the scope of experimentation that may be required to meet the goal of meaningful measurement of predictive capability. One test each at a few x-points, e.g., does not provide a very good estimated map of σ_x as a function of x.

Issues and Discussion

Introduction. The Fig. 1 schematic embodies a variety of issues that must be addressed in order to quantify prediction uncertainty successfully. Several of these issues are discussed here and then explored via a sample problem in the next section.

Model Validation – the Concept. The term validation invites the interpretation: affirm the validity (truthfulness, accuracy, genuineness – Microsoft 95 Dictionary) of. From this interpretation it might be concluded that predictions using a validated model can be taken as substitutes for actually observing nature's outcome in the situations modeled, at least in some x-region where the model is deemed valid. But this is an unwarranted stretch (see Popper 1969 on the philosophical untenability of validation of scientific theories, in general, and Oreskes et al. 1994 for a more recent discussion). At best we can measure the difference between prediction and nature at some number of selected points in the x-region. This process is in line with the following definition:

Validation: The process of determining the degree to which a computer model is an accurate representation of the real world from the perspective of the intended uses of the model (AIAA 1998).

As discussed in the AIAA Guide, one can determine (actually, estimate) accuracy only at points at which model output and the real world can be compared – e.g., the pairs of experiments and computations in the bottom ellipse of Fig. 1. Accuracy at untested x-points, which could correspond to "intended uses of the model," can at best only be predicted or inferred. But it is at points that cannot or have not been tested that we most want to be able to make credible statements of degree of accuracy. Otherwise, why develop a model? Common statements to the effect that 'we will base decisions on computational predictions from a validated computational model' rest on an inference that may not be justified or even recognized. The problem that must be worked in order to establish confidence in computational predictions beyond available data is the inference problem in Fig. 1. There is no guarantee of success. Having this objective, though, will be useful in identifying barriers to obtaining credible limits of prediction-uncertainty for computational predictions.

Model Validation as Hypothesis Testing. Model validation is often approached as a hypothesis-testing problem, e.g., Hills and Trucano (1999, 2000) and Coleman and Stern (1997). The statistical set-up for the test is to model the experimental result, y, for an experiment conducted at a particular x-point, as a random selection from a distribution that has expectation μ_e, which is unknown, and standard deviation σ_e, treated as known but in general estimated, say by s_e. This standard deviation represents measurement variability. Similarly, the computational prediction at x, namely y* (which is the computational prediction using all the nominal model parameter estimates), is modeled as a random selection from a distribution that has expectation μ_c (which is unknown) and standard deviation σ_c, treated as known but in general estimated, say by s_c. This standard deviation reflects uncertainty in the parameter estimates used in the model.

In terms of this statistical model, the hypothesis tested is $\mu_e = \mu_c$. The question asked, via a hypothesis test, is whether the observed difference, $y - y^*$, is satisfactorily within the combined uncertainties. (For scalars, the test statistic for testing the hypothesis is $(y - y^*)/s$, where $s = \sqrt{(s_c^2 + s_e^2)}$, and for vectors, the test statistic is a vector analog of this quantity.) If the agreement is satisfactory, experimental and computational expectations are regarded as equal (at least the data don't rule out equality of μ_e and μ_c). Presumably, the model is therefore deemed valid at the x point at which the comparison is made. An extension, as in Hills and Trucano (2000), is to test whether an ensemble of differences, $\{y-y^*\}$, over different x's are within their combined uncertainties. If not, the model is declared invalid. The remedy, then, is probably to try to improve the model. In this sense, model-validation continues the model-improvement process.

One flaw in this analysis is that the combined standard deviation, s, does not capture what may be a major contributor to the difference, $y - y^*$. That contributor is the combined effects of the w's that influence nature's outcome but are not in the model. The effect of this omission is to increase the probability of declaring the two expectations to be significantly different.

On the other hand, the combined standard deviation includes s_c, the standard deviation that reflects parameter-estimate uncertainty associated with the parameters in the model. As discussed below, our goal is to characterize the predictive capability of the model with the current best parameter values, not try to capture the unknown prediction that would be obtained if we knew the true parameter values. Thus, s_c is not a component of the current model's prediction error. An exception, as is also discussed below, is that if validation data are used to 'tune' the parameter estimates, the effect of tuning needs to be accounted for in using the same data to evaluate the predictive capability of the tuned model.

Rejecting the hypothesis of equal underlying expectations does not mean that the model is not useful for predictive purposes. An ensemble of differences might show that y^* predicts y consistently within 15%, which might be perfectly tolerable in the context of interest, even though the combined separate uncertainties were only 5% in magnitude. Conversely, accepting the hypothesis of equal expectations does not guarantee that useful predictions of y are provided by y^*. In fact, the more uncertain y^* is, the more likely it is to pass the hypothesis test of equal expectations and be accepted as a valid prediction. Similarly, the noisier the data, the easier it is to pass the hypothesis test. These properties do not encourage improved models or experiments. Thus, in my view, hypothesis-test pass/fail results, or refinements such as P-values (see the following Note) associated with the test, are inadequate and inappropriate tools for characterizing predictive capability and justifying conclusions based on computational predictions.

[Note. Classical hypothesis testing requires the prior determination of α, the probability of falsely rejecting the null hypothesis, and then an acceptance criterion is established that achieves this α. Rather than the binary pass/fail outcome, an alternative is to summarize the test by finding the largest α-value for which the test would fail. This threshold value is termed the P-value and is continuous on the [0, 1] interval (Gibbons

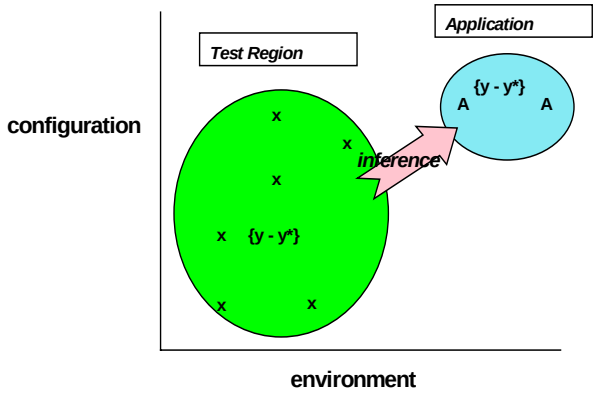
and Pratt 1975). Stated another way, a P-value is the probability of observing a result that contradicts the hypothesis by as much or more than does the observed result.]

For the set-up proposed here, one could use an ensemble of differences, $\{y - y^*\}$, obtained from multiple experiments at a given x-point, experiments at different x-points, or both to test the hypothesis of an expected difference of zero (after first determining that there were no systematic effects associated with x). The standard deviation of the differences would, in contrast to the above pooling of the separate estimated variances, provide the appropriate measure of the variability of $y - y^*$. The purpose of the hypothesis test in this case is to see whether the data support simplifying the error model by setting δ_x equal to zero. Rejection of that hypothesis would be a prompt to consider further model improvement in order to reduce or eliminate the bias in the predictions.

Experimental Design. The ability to credibly characterize predictive capability depends heavily on the suite of experiments and computations that will provide the y and y* data for subsequent analysis. In broad terms, 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 test plans that specify the test hardware, methods, conditions, instrumentation – extent, type, and location – and data collection and post-processing techniques used to obtain information required for subsequent data analyses. For example, it may be necessary to measure boundary and initial conditions, quantities that may not normally be easily measured. It is important to recognize that measuring predictive capability has profound implications for the experimental sciences, not just the analytic.

Figure 2 depicts the inference problem. The goal is to characterize predictive capability, represented by $\{y - y^*\}$, for an application based on observed predictive capability in selected experimental contexts. Both the experimental and application contexts are defined in terms of two meta-variables, configuration and environment. Different test entities (configurations) are to be subjected to different environments (the left ellipse). Before we can characterize predictive capability for the application (the right ellipse, which denotes the system or component configurations of interest and the environmental range in which it is to function), model-validation experiments and computations will be conducted for different hardware configurations and environments, generally less complex than the application. For the sake of generality, Fig. 2 shows the inference problem as extrapolation; interpolation should be easier. The experimental design issue is the choice of points, denoted by x's, in the left ellipse.

Figure 2. The Inference Problem



As an example, the application of interest could be a new-design neutron generator subjected to some range of radiation environments. The Comprehensive Test Ban Treaty precludes the historical means of testing a neutron generator in a threat-like environment via an underground nuclear explosion. Lesser environments, however, can be obtained from various above-ground radiation sources. On the configuration axis, which is probably easier thought of as discrete rather than continuous, initial computational predictions and experiments may be done for simple geometries, such as a flat plate or an aluminum cylinder, rather than for a neutron generator with its complex assembly of diverse parts. For evaluating prediction error for more complex assemblies, historical data might provide performance data for earlier generation neutron generators in underground tests. Or, the neutron generator could be tested in above-ground radiation environments, where feasible. Corresponding computational predictions would need to be run for all of these cases in order to evaluate prediction capability.

In choosing the experimental design and conducting the experiment numerous issues will arise. The following paragraphs discuss several of these issues.

a. Experimental Objectives. Meaningful (as opposed to haphazard) experiments are designed to meet explicit objectives. The goal of model-validation experiments, as I advocate here, is to measure predictive capability, first at selected situations in which predictions can be compared to experimental results, then to infer, or estimate, predictive capability for situations for which experimental results will not be obtained. Thus, in general, the experiments conducted (1) should provide an adequate measure of predictive capability for the selected experimental situations and (2) the collective set of experiments and computational predictions should provide a basis for the desired inference to untested situations.

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 set the objective of being able 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 experiment should also provide the data with which to test that assumption). Another measure of predictive capability is the expected value of e_x, call it the bias, at x. Ideally, the bias would be zero, but there is no guarantee. One could set the objective of being able to detect a serious bias with a high probability and then derive the number of experiments required to have that capability (termed statistical power; discussed in detail and illustrated in the Illustration: Linear Model section below).

Selecting the suite of x-points at which to experiment can have several objectives. At early stages in the validation cycle, the mode is exploratory: one wants to assure that portions of the code can adequately predict selected phenomena. Conversely, one would want to detect poor predictive capability at this stage, find its cause, and either fix the model or the experiment before proceeding to more complex situations. Modelers and experimenters will need to select suitable x-points for this sort of experimentation and prediction, based on situation-specific conditions.

Another experimental objective might be 'model-busting.' Pick x-points for which there is reason to expect that the computational tool will be sorely tested. Such testing would also be justified by a bounding approach to characterizing prediction capability. Rather than attempt to characterize prediction capability at each of several x-points, pick a situation for which it can be convincingly argued that prediction capability elsewhere (within reason) could be no worse. This approach is one way to limit the amount of required testing.

Ultimately, the goal is to synthesize information obtained about predictive capability at the selected x-points into an estimate of predictive capability for an untested application. Let $x_1, x_2, \ldots, x_n$ denote the x-points at which the experiments are to be run and let x_a denote the x-point for the application. The set of x_i points need to be chosen to support inference to x_a. Ideally, for the sake of confident inference, the x_i would surround or even include x_a. If that is not possible, we would want some x_i as close as possible to x_a in order to make the inference stretch as short as possible. Of course, this means higher-level, more expensive and complex testing and less ability to take high-quality measurements, so there are major trade-offs to analyze and resolve.

Statistical experimental design methods can be used to select an efficient design (suite of x-points) corresponding to a contemplated model. For high-dimensional x, because of the cost of experimentation and, in some cases, computational model set-up and running, the model may have to be quite simple. For example, suppose the objective was to model the bias in predictions at x by the linear model, $\delta_x = \Sigma d_i x_i$, where the x_is are individual x-variables in the input vector, x. Then, a highly fractionated two-level factorial experiment could be run to support fitting this error model (fractional factorial experiments are described, e.g., in Box, Hunter, and Hunter 1978). If the goal was to

detect serious bias, then statistical power calculations could be used to characterize a design's ability to detect nonzero d_is.

The conduct of the experiment also influences how well predictive capability can be measured. As mentioned, a variety of random and systematic factors can come between computational prediction and nature. The experiments need to be conducted in ways that allow these factors to be manifested in the same way they would be in an application of interest. This means that predictive capability measured in a tightly-controlled, pristine lab environment may not be appropriate for inferring predictive capability in a much less controlled, noisier application environment. (One possible approach to this problem: model predictive capability as a function of lab-controllable variables, then propagate the application's estimated uncontrolled distribution of these variables through the prediction-error model.) The objective of measuring predictive capability in a specific application affects experimental design in terms of both what is controlled and what is not controlled in the experiments.

b. Constraints. 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, holding the others fixed at nominal, or perhaps bounding, values. The subset selected should be those variables that provide a meaningful basis for estimating predictive capability in the application of interest. Another way to simplify the problem is to consider slices through the x-space. For example, radiation effects experiments could be done at different radiation levels for an aluminum cylinder and for a neutron generator mock-up. The differing radiation environments for the two configurations are 'slices' through the variable space. Though an aluminum cylinder and a neutron generator mock-up may be representable as two points in a high-dimensional x-space, one would not undertake experiments that, in some sense, fill that x-space.

Because of cost, treaty conditions, or test-facility limitations, system-level model-validation testing may not be possible. Hence, the extrapolation indicated in Fig. 2. Also, for complex multi-phenomena models, the fully combined-phenomena tests may not be feasible, while single phenomenon or some double-phenomena tests may be. In stress-strength situations, one portion of a model may predict the stress put on a system; another portion may predict its strength; separate validation tests may be feasible while combined tests are not. Thus, in general, it is recognized, see Oberkampf and Trucano (2000) and Trucano and Moya (1999), that model validation needs to occur as a series of building blocks as first parts, then combinations of parts of the model are evaluated. At a minimum, validation tests on the building blocks provide a final test on portions of the computational model, a chance to continue the model-improvement process. They also provide practice in the process of designing, conducting, and analyzing the results of experiments and computations to characterize prediction error for the x-sub-regions represented by the sub-problems. At a maximum, though, the building blocks will be designed so that their results can be integrated to provide the system application level inference of prediction uncertainty.

In general, limitations of time, resources, and experimental capability substantially influence experimental design and conduct. Such constraints will have to be balanced against experimental objectives in arriving at a plan for model-validation experimentation. A decision will have to be made as to whether a meaningful evaluation of predictive capability is possible under existing constraints in any given situation.

c. Experiment-Model Compatibility. The computational and experimental sides of the model-validation process depicted in Fig. 1 cannot be done in isolation. The vector x needs to be meaningful to both the experimentalist and modeler in order that both computational predictions and experiments at selected x-points can be run and compared. 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. The computational prediction will have to be based on assumed nominal values of the unmeasurable x's, which means that deviations of the actual x's in the experiment from the assumed nominals become, in essence, nature's w's and thereby add variability and increase prediction error relative to what would be obtained if the x's could be measured and predictions calculated based on the measured values. The assumed nominals are also a source of bias that may be difficult to ferret out. That is, if bias was observed between experimental results and computational predictions based on assumed nominals, it could not be determined whether the source of the bias was the computational model or the assumed nominals.

There is a natural tendency to attempt to put more and more science into computational models as computers increase in capability. That's progress; that's science. In terms of the mathematical model in the previous section for computational model and nature (1a and 1b), this modeling progress means identifying some of the w's in nature's function and moving them into the model, $M(x:\phi)$. Replacing a 2-D code by a 3-D code is an example of this sort of evolution. However, this expansion complicates and expands the effort required to cover the pertinent x-space experimentally. It also increases the dimensionality of ϕ, which means more parameter estimates are required. On the other hand, the experimentalist or validation test designer would just as soon leave the w's where they are and perhaps move some of the x's, especially the unmeasurable or difficult-to-measure ones, out of $M(x,\phi)$ into e_x so they don't have to be measured or controlled in the experiment. The philosophy would be to let nature reveal the effect of the uncontrolled x's and w's experimentally, via the prediction errors, $\{y - y^*\}$, rather than try to include them in the mathematical model.

I characterize these opposing tendencies between modeler and experimentalist as 'the battle of the x's and w's.' From the perspective of model-validation and prediction-uncertainty quantification, it is not necessarily the case that more (model sophistication) is better or cost-effective. The cost of developing, then experimentally validating, the more sophisticated model compared to conducting tests that characterize the prediction error for a less sophisticated model may not justify doing the former.

In sum, the ability to measure predictive capability of a computational model can and should influence computational model design and development, just as testability and the ability to analyze safety influence hardware design. The modeler cannot just toss the model over the transom for validation. The experimenter cannot insist on conducting and instrumenting tests in ways that were developed for other purposes. Collaboration and trade-offs between model and experiment will be necessary to achieve the compatibility that will provide an informative comparison of computational predictions and experimental results at a meaningful set of x-points. Regardless of computing resources, model simplification is in general a necessary prerequisite to successful prediction uncertainty quantification.

Data Analysis. After conducting a suite of experiments and computational predictions the next task is to analyze the resulting data, $\{x_i, y^*(x_i), y(x_i): i = 1, 2, \ldots n\}$, where the subscript denotes the different x-points at which experiments and computations were conducted. (Recall that x and y(x) can both be vectors or fields. The subscript i represents different experiments, not different data points within one experiment.) The objective of the analysis is to estimate predictive capability. The following subsections address issues that arise in this analysis.

a. Metrics. Predictive capability at an x-point can be measured by a variety of characteristics of the probability distribution of e_x. The expected value and the standard deviation of e_x are two possibilities mentioned above. Others might be the square root of the expected squared error ($[E(e_x^2)]^{1/2}$, where E(.) denotes expectation), the expected absolute error times three, the 99^{th} percentile of the distribution of absolute error, the lower and upper 95^{th} percentiles on the distribution of e_x, etc. If the computational model was set up to be conservative on the high-side (i.e., $y - y^*$ is intended to be negative), the metric of interest might be $Prob(e_x > 0)$. When e_x has a normal distribution all of these distributional characteristics are functions of the two parameters that characterize a normal distribution, the expected value (or mean) and the standard deviation.

None of these measures of predictive capability are known; they cannot just be assumed or analytically derived fom ad hoc assumptions; they must be estimated from the experimental and computational results. That is why the tests are conducted. With limited data, estimation uncertainty will be appreciable. Statistical methods account for estimation uncertainty by methods such as confidence limits. E.g., with 90% confidence the upper 95^{th} percentile of the distribution of e_x is no more than $U_{90/95}$. Other methods may also be candidates for evaluating the reliability (accuracy and precision) of estimated measures of predictive capability. The essential point is that any predictive-capability 'metric' derived from the model-validation process will be an estimate and the reliability of that estimate must also be considered and communicated.

b. Choice of Analysis Variables. In both experiments and computations there are a large number of performance variables that can be observed, computed, and compared. Making the analysis manageable and the results meaningful and communicable requires a judicious selection of variables for which to evaluate predictive capability.

The selection of variables should first be requirements-driven. If the requirement is that peak strain at a given location should not exceed some value, then the model-validation objective is to estimate predictive capability pertaining to calculated peak strain at that location. Similarly, system and component requirements should drive both modeling and experimentation to assure that, e.g., both provide peak strain results at that location. While it would add confidence in the computational model to know that the complete strain vs. time response at various locations in the test device can all 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 may be traces of responses such as acceleration, strain, or temperature in time and space, to a small number of 'integral' variables such as peak acceleration, peak-to-peak strain, the 'area-under-the-curve,' or the time to reach critical temperatures at selected points in a system or component. For analyzing a vector of performance variables, vector analogs of analyses described here can be carried out, as in Hasselman and Anderson (1999).

c. Inference. Suppose now that at n selected x-points, $x_1, x_2, \ldots, x_n$, we have conducted and compared experiments to computations and have estimated the standard deviation of the prediction error distribution at each point: $s_1, s_2, \ldots, s_n$. Suppose further that we have measures of the precision of these estimated standard deviations. (In a conventional statistical setting, these measures would be the 'degrees of freedom' (df) associated with the estimates.) The application of interest is defined by the point, x_a. The inference problem, in terms of this metric, is how to use the $\{x_i: s_i\}$ set of results to obtain s_a, the estimated prediction-error standard deviation at x_a, and to obtain a measure of the precision of that estimate.

This, in general, is a difficult and philosophically deep problem. What do imperfectly known knowns tell us about an unknown? The ability to satisfactorily, not perfectly, solve the inference problem depends on a number of considerations. First, the definition of the x-space is critical in order that x_a and the x_i be comparable. Again, the definition of the variables in the x-vector is not just a modeling issue. The experimenter, the specifier of requirements, and the decision-maker have to be able to operate and communicate in terms of this x-space. Next, the ability to draw inference depends on the location of x_a relative to that of the x_i. If x_a is, in some sense, surrounded by the x_i, the problem is one of interpolation. If x_a is beyond the x_i, then inference requires extrapolation. It is important to note that even if the underlying scientific relationships (e.g., linear differential equations) on which the computational model is based are known to extend over a region containing both the x_i and x_a, there is no such basis for extrapolating the error distribution which, after all, reflects factors in nature not captured by the scientific model. Inference about predictive capability will have to depend on empirical trends and expert interpretation of those trends. To satisfactorily solve the inference problem, it is clearly important to test at some x-points that are as close as possible to x_a, in order to minimize the inferential stretch. This means that some system-level testing will be highly desirable.

As an example, suppose that radiation effects testing of aluminum cylinders in various above-ground radiation environments and the corresponding computational predictions indicate that peak stress at various locations and orientations can be computationally predicted in these conditions with a relative standard deviation of about 10%. Suppose that above-ground radiation tests of and computations for a sub-scale partial mock-up of a weapon component indicate a prediction-error relative standard deviation of about 20%. We now want to make a credible, defensible statement about predictive capability in terms of calculated peak stress at critical points in a full-up component subjected to a radiation environment that is different from those achievable above-ground. The solution depends on our ability to link the test configurations and environments, $\{x_i\}$, to the application, x_a. Such a linkage may have to be judgmental, not mathematical: e.g., "In our judgment, the additional complexity in the application should cause the prediction-error standard deviation in the application to be no more than a factor of two greater than it was for the mock-up." While I have tried to define and lay out a technical/scientific approach for working the problem of measuring predictive capability, I have no illusions of eliminating subjectivity from inferences of predictive capability.

To repeat, achieving credible inference is obviously going to be a difficult problem to solve. The situation, though, is not unlike the inference required from tests alone. From component and system performance observed in various selected test environments we infer that similar performance will occur in the use environment. The confidence in such inferences derives from the credibility of the test program. It is not unknown to have incorrect test-based inferences when test environments have erroneously been thought to be adequate approximations to reality. For computation-based inference, we will have to infer that performance of the computational tool in test environments is indicative of its performance in the application environment. The credibility of that inference will depend on the relationship of the test environments, the x_i, to the application environment, x_a, and our ability to understand that relationship and its implications for predictive capability.

In passing, I would note that the spatial representation of the experimental design and inference problems suggests that spatial statistical methods, such as kriging (see, e.g., Chiles and Delfiner 1999), 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.

d. 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. That is, systems are not identical and delivery and target conditions, such as impact angle, impact velocity, and frozen-soil depth and composition, 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 such as its mean, its upper two-sigma point, or the probability of exceeding a failure threshold. The B61-11 analysis of Field et al. (1999), in which two variables, the angle of attack and the depth of

frozen soil at the target, were treated as random while other x-variables in the model were held constant at nominal values, gave rise to this sort of distributional prediction. Methods for propagating random variation through a computational model have been and continue to be extensively researched (see, e.g., Red-Horse et al. 2000) and practiced. The issue here, though, is: How can the results of model-validation experiments be used to characterize the uncertainty of such distributional predictions?

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. (For some requirements, e.g., performance over an assumed annual distribution of weather variables, this assumption may be appropriate. In other cases, the assumed distribution may be an uncertain estimate and this additional uncertainty will need to be accounted for in the final analysis.) Suppose further that the relationship between nature and computation as a function of x, as specified above, is

$$y(x) = y^*(x) + e_x; \quad e_x \text{ is random with mean } \delta_x \text{ and standard deviation } \sigma_x.$$

The law of total variance (see, e.g., Parzen 1962) says that

$$\text{var}(y) = \text{var}_x[E(y|x)] + E_x[\text{var}(y|x)], \tag{3}$$

where var(.) denotes variance, E(.) denotes expectation, and | denotes conditioning. The subscript indicates the random variable over which these moments are calculated. In words, (3) 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:

$$\text{var}_r(y) = \text{var}_r[y^*(x) + \delta_x] + E_r[\text{var}(e_x)], \tag{4}$$

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 (4) becomes

$$\text{var}_r(y) = \text{var}_r(y^*) + E_r[\sigma_x^2]. \tag{5}$$

Propagation of the assumed distribution of x_r through $M(x:\phi)$ provides an estimate of the first right-hand term in (5). Model-validation experiments and data analysis, if successful, provide an estimate of σ_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 (5). In the ideal situation in which σ_x is independent of x in the region of interest, the second right-hand term is simply σ_e^2, the variance of the difference between nature and computation. In either case I call σ_x^2 the 'extra-model' variability induced by the variation of the w's, which are nature's variables, not in the computational model. Similarly to (5), 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 (5) 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 (5), underestimates nature's variation of 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.

As mentioned, traditional code uncertainty-propagation analyses work the first right-hand term, 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, δ_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 Oberkampf (1997) for discussions and illustrations on this point in the context of fluid dynamics.

A variety of issues arise in implementing the indicated analysis in this section. If there is uncertainty about the assumed distribution of x_r, this would need to be addressed, either by a sensitivity analysis in which the analysis was repeated for different assumed distributions or, as described in the next section, by a nested propagation. If code realizations are expensive, then it may be possible to make only a few runs to provide an estimate of $var_r(y^*)$. The statistical uncertainty of this estimate would need to be accounted for when this estimate is combined with the model-validation based estimate of $E_r[var(e_x)]$, which itself may be based on limited data and thus have appreciable statistical uncertainty. If methods other than simple random sampling from the assumed distribution of x_r are used, such as constrained Monte Carlo methods or response surface methods, the uncertainty of the resulting variance estimates would also needed to be accounted for in the final analysis. This accounting can be considerably more difficult, technically, than it is for random sampling.

e. Code Uncertainty Propagation. The values of some of the parameters in ϕ, used in calculating $y^* = M(x:\phi)$, are often estimates obtained from calibration or model-identification experiments, rather than theory-provided constants. In code uncertainty analyses, this estimation-uncertainty is often expressed by an analyst as a probability

distribution, which is then propagated through the computer model in an evaluation of the uncertainty of model predictions. (This is a fundamentally different level of propagation than the propagation of assumed random variable distributions discussed in the previous section; more on this distinction in the next paragraph.) The interval between selected estimated percentiles of the distribution generated by this propagation can be thought of as a confidence interval on the (unknown) computational prediction that would be obtained if we knew the 'true values' of pertinent parameters. This confidence interval is not a prediction interval for nature's outcome in that particular situation. Code uncertainty analyses are conditional on the computational model and thus do not and cannot address the difference between nature and computational prediction.

In some code-uncertainty analyses both estimation uncertainty and assumed random variability are simultaneously (probabilistically) propagated through $M(x:\phi)$. It is appropriate, however, to separate estimation uncertainty and random variation (see, e.g., Kaplan and Garrick 1981 and Ferson and Ginzbirg 1996). This can be done through nested propagations. In the outer loop a value of ϕ, call it ϕ_i, is drawn from the 'uncertainty distribution' of ϕ. In the inner loop, the assumed (or estimated) probability distribution of x_r is propagated through $M(x:\phi_i)$ and parameters of interest, such as the upper two-sigma point on y*, given ϕ_i, are obtained. Repeating the outer loop, then the inner, some number of times provides an estimated distribution of estimated upper two-sigma points,say. From this estimated distribution a nominal estimate and an confidence interval on the upper two-sigma point on y* can be obtained. This analysis, however, still omits the extra-model variation and so could be misleading if it is interpreted as bounding nature's upper two-sigma point.

f. Estimation Uncertainty. My proposed analysis focuses on the observed prediction errors, $\{y - y^*\}$. When y* is based on $M(x:\phi^*)$, where ϕ^* is an estimate of the parameter ϕ, with recognized estimation uncertainty, it may seem appropriate to incorporate this estimation uncertainty into the analysis. However, the purpose of the suite of model-validation experiments is to characterize how well $M(x:\phi^*)$, with our current ϕ^*, predicts nature, not characterize how well $M(x:\phi^*)$ with a random ϕ^* predicts nature. Thus, it is not appropriate to incorporate estimation-uncertainty, with respect to ϕ^*, into the analysis. If the observed prediction errors have patterns that suggest problems due to ϕ^*, this could lead to model improvement via improvement or updating of the estimate ϕ^*. The source of an improved estimate could be additional, independent experimentation, or ϕ^* could be calibrated/tuned to the available model-validation data in order to improve some characteristic of the observed prediction errors, such as their root sum of squares (RSS). Such an analysis is done by Blackwell et al. (2000) in order to estimate thermal conductivity of an experimental specimen of stainless steel. Tuning is discussed below.

g. Diagnostic Analyses. The analysis of predictive capability may indicate that predictive capability is not satisfactory. If so, then it is desirable to diagnose the problem and correct it, if possible. For example, unless the computational model was set-up to be conservative, a finding of bias may indicate either a problem in the model or in the experiment.

Biases in the experiment could result from inability to conduct the experiment at the x-point for which the computational prediction is obtained. The way in which x is either measured or controlled in the experiment may be the source. Biases in the computation could result from the numerical set-up of the code or from errors in estimating the model parameters, represented by ϕ in the mathematical representation (1). Analysis is required to sort out these potential sources of bias and point to appropriate fixes.

Excessive variability of prediction error can also be a problem to be diagnosed and alleviated. Measurement error is one possible source. The difference between measured nature and the computational prediction contains measurement error. If the measurement error variance is known or well-estimated, it can be subtracted out of the observed prediction error variance. When x-values, such as boundary conditions, are experimentally measured, the errors in these measurements transmit error into y*. Propagation of an assumed/estimated x-measurement error distribution through y* could provide an estimate of this component of variation in y-y* which could then be subtracted out of the observed prediction error variance. A similar sort of measurement error occurs when the computational prediction, y*(x), is calculated at nominal x-values and the experiments are run at x-values that may differ from the nominal, but which cannot be measured. If the application of interest would not have this source of variability in its prediction error, it is appropriate to subtract it out. With information about the variance of actual x and the relationship between y and x, this could be accomplished.

Another source of variability, of course, is 'unmodeled physics.' This is exactly the variability that the model-validation experiments are designed to capture. Any attempt to reduce this source of variability by expanding the model needs to be balanced against the cost of further model development and the required verification and validation efforts that this would entail.

h. Model Tuning. When the analysis of prediction error data shows evidence of a bias, one can 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 ϕ parameters, which, as noted, may often be uncertain estimates. 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 α and β 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 α and β 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 α and β 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 ϕ 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 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.

For the case of a single estimated parameter, suppose that theory and subject-matter knowledge support an assumption that a particular material property, such as a thermal conductivity, is the appropriate tuning parameter. Suppose that one experiment is done, then the parameter, call it ϕ, is adjusted so that $||y - y^*||$ is minimized (for vector outputs such as atemperature/time history). Call the tuned parameter value $\hat{\phi}_1$. From this one experiment there is no way to assess the imprecision of $\hat{\phi}_1$. Suppose a second experiment was done and the model again tuned to the data; call the tuned parameter value $\hat{\phi}_2$. Then, it might be appropriate to treat these two estimates as independent estimates of the unknown parameter. Thus, a combined estimate of the parameter would be the average of the two estimates, and confidence intervals, quite broad, could be calculated for the unknown parameter value. If more tests were done and more estimates of ϕ obtained, the precision could be improved. Further analysis of the set of results could then yield an estimate of prediction-error variance, say, that accounts for tuning.

i. Integrating Suites of Experiments and Calculations. The ultimate goal in measuring predictive capability, as indicated in Figs. 1 and 2, is to integrate the results pertaining to prediction error in testable configurations and environments to provide useful information about predictive capability for computational predictions in the application context. This is not a problem that has been worked, in general. The ability to do so will depend on how the suite of experiments relate to the computational model as it is configured to provide a prediction for an application of interest.

This analysis problem is analogous to the problem of estimating system reliability from component reliability test data. If there is a mathematical model linking system reliability to the ensemble of component reliabilities, then it is possible to obtain such inferences as statistical confidence limits on system reliability based on component data and the system reliability model (Easterling and Spencer 1986). This inference is conditional on the assumed reliability model. If, for example, there are appreciable component interface unreliabilities not included in the model, the system inference will not be valid. In the model-validation context, if one can link physics sub-models to form the system application model and if one can obtain prediction uncertainty data for the

sub-models, then a similar analysis can be done. For situations in which the output of one subroutine is the input to another, this integration seems feasible. When there are complex, iterative feedback loops between phenomenon models, it may not be.

To be more explicit, suppose that the phenomenon of interest in the application can be expressed as,

$$y_A^* = M_A(y_1, y_2, \ldots, y_k \mid \phi_A),$$

where the y_i represent various phenomena that combine according to the model, M_A. Suppose that computational models exist for each of these phenomena,

$$y_i^* = m_i(x_i : \phi_i)$$

and that model-validation experiments (the x-points in Fig. 2) have been conducted and compared to the computational predictions to estimate the prediction error distributions for each of these sub-models. These estimated distributions could be propagated through the M_A model to estimate the prediction error associated with y_A^*. A very simple example of this sort of analysis is a stress-strength situation for which there are separate models for y_1 = stress; y_2 = strength (at the application scale) and the application variable of interest is the margin, $y_A = y_2 - y_1$. Combining the estimated prediction error distributions for y_1^* and y_2^* is then straightforward. The ability to work this problem in much more complex situations remains to be determined. Response surface approximations to the M_A function may be a workable approach.

Summary. Systematically demonstrating and documenting predictive capability for ASCI-level computational predictions is an extremely challenging problem. There will be situations in which it will be impossible to obtain enough of the right data or to extend the obtainable data to make the required inferences. There are also situations in which, as Romero (2000) documents, not all physical phenomena pertinent to certification of nuclear weapon performance can be mathematically modeled at present. Knowing when and where such barriers are encountered, though, is valuable information. Approximations, simplifications, work-arounds, etc. will be required, all the while maintaining scientific credibility for the process. Softer methods, such as those based on quantification of expert opinion ("We feel strongly, based on the evidence we were able to accumulate in model-validation experimentation, that the computational prediction cannot be off by more than a factor of two.") may also come into play.

Where prediction error limits are obtainable, it is evident that an appreciable amount of testing will be required for the sorts of complex, high-dimensional computational models that may be used for predictions of nuclear weapon performance. While the task is daunting, as this section has outlined, it is not impossible, as the example in the following section illustrates.

Illustration: Linear Model.

Introduction. While the general inference problem in Fig. 1 is quite formidable, there are situations in which it can be worked, so hope need not be abandoned.

Suppose theory says that y is a linear function of x, both scalars. Thus, the computational model is simply:

$$y^* = M(x:\phi) = \alpha + \beta x. \qquad (6)$$

Such a model can be arrived at directly, perhaps after taking logarithms or some other transformation, or as a first-order approximation to a more complicated relation within a suitable region. Another source for a linear model is a situation in which computer-model exploration in a region of interest suggests that a linear model is an adequate approximation for a more complex relationship. This is the case for the CTH code (McGlaun, Thompson, and Elrick 1990) calculations of shock wave velocity as a function of particle velocity (which is one-half of impact velocity) in Hills and Trucano (2000) for a particular class of pellet on plate impacts. That is, the shock physics finite-volume code, CTH, in the context considered, predicts shock wave velocity, to a very close approximation, by a linear function of impact velocity. Thus it is not unrealistic to use such a simple model to illustrate the measurement of predictive capability. A clear understanding of issues and methods in this situation will provide a basis for tackling more complex situations. Furthermore, such simplifying approximations will probably be necessary for measuring predictive capability in complex situations.

For a linear prediction model the slope (β) and intercept (α) may be provided by theory or by a source such as a handbook or laboratory testing, or these parameters may have to be estimated (or re-estimated) from the data obtained in model-validation experiments. This section illustrates the analyses and issues pertaining to quantifying the uncertainty of computational predictions based on this simple linear model in these different contexts.

Prediction Uncertainty Quantification: Point Prediction. Validation experiments for the linear model consist of specifying a set of x values, then conducting some number of experiments under appropriate conditions at the selected x values. The outcome of these experiments is a set of pairs, {x, y}. For a given slope and intercept, the model prediction, y*, and the prediction errors at each x, namely $\{y_x - y_x^*\}$, can be obtained. Ideally, these errors would look like random samples from a normal distribution with mean zero and a standard deviation, σ, that is constant across x. In general, though, as in Hills and Trucano (2000), there would be some pattern in the prediction errors and this pattern could be modeled and then used to adjust, or calibrate, future predictions. More likely, though, the observation of a pattern would be seen as evidence that the assumed model parameters, α and β, or perhaps the assumption of linearity, were inappropriate for this suite of experiments, so either the model parameters or the model form would be adjusted to improve the fit.

One such adjustment, still assuming linearity, would be to fit a straight line by least squares to the observed {x,y} data; call the result y*^ = a + bx. This line, the estimated variability about it, and the statistical uncertainty of the estimates, a and b, could be used to predict future y's at selected x values and, furthermore, valid prediction error-limits at those x values can be calculated that account for the fact that the model has been fitted, or 'tuned,' to the data. The numerical example below gives an example and shows how prediction uncertainty increases as the x-value at which the prediction is made moves further and further from the center of the test data. See Draper and Smith (1966) or other statistics texts for details. One theory-based assumption required by the analysis is that linearity holds over the range of the data and extends, if need be, to the point at which a prediction is desired. This assumption must be subject-matter-based, not statistical. For the standard analysis, the additional assumption of a constant variance of y over the x-range of interest is also required. This key assumption is the basis for the extrapolation of extra-model variability discussed previously. The data themselves can be used to assess this assumption over the range of experimentation and the analysis can be modified to reflect some other relationship between the variance and x over the x-range of interest. The additional assumption of normality is required in order to associate confidence levels with prediction intervals.

As discussed above, at least three experiments are required for the statistical prediction limits to exist because the residual standard deviation is based on n – 2 degrees of freedom, where n is the number of data points.) The 'inferential penalty' for tuning the linear model to the data used to estimate its parameters is the loss of two degrees of freedom (df), reflecting the estimation of two parameters, in estimating the residual variance – the variance about the fitted line. This adjustment scales up the residual variability and means that the multiplier (the t-value) for prediction uncertainty intervals is larger than in the case of a well-fitting external model.

Example. To illustrate the concepts and methods of prediction uncertainty quantification as applied to the linear model situation, I consider the model and a portion of the data considered in Hills and Trucano (2000). 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 to predict shock wave velocity in the aluminum plate as a function of particle velocity, which is one-half the pellet's impact velocity. Internal to the code is the Sesame equation-of-state (EOS) model that relates shock wave velocity to particle 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:

$$Us^* = 5263 + 1.368Up, \qquad (7)$$

for Up between roughly 300 and 4000 m/s, where Us is shock wave velocity and Up is particle velocity, both in m/s. (Obviously this linear approximation cannot be extended to Up = 0.) 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 for with this prediction,

relative to nature's outcome for such an impact? We need some model-validation experimentation and data.

For this illustration, suppose that only a limited number of experiments are possible and that they are constrained to be at particle velocities, Up, no higher than around 3000 m/s. In particular, I will use six of the 232 tests reported in Hills and Trucano (2000) 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, the required inference will require extrapolation, as in Fig. 2. 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 first carry out the analysis based on the assumption, not strongly contradicted by the data at conventional significance levels, that prediction errors at specified values of Up are normally distributed with a mean of zero and a constant standard deviation σ, for Up ranging from 2000 – 3000 m/s. With this set-up, σ is estimated by the root mean square (RMS = square root of average squared-error) of the six observed errors, namely s = 146 m/s.

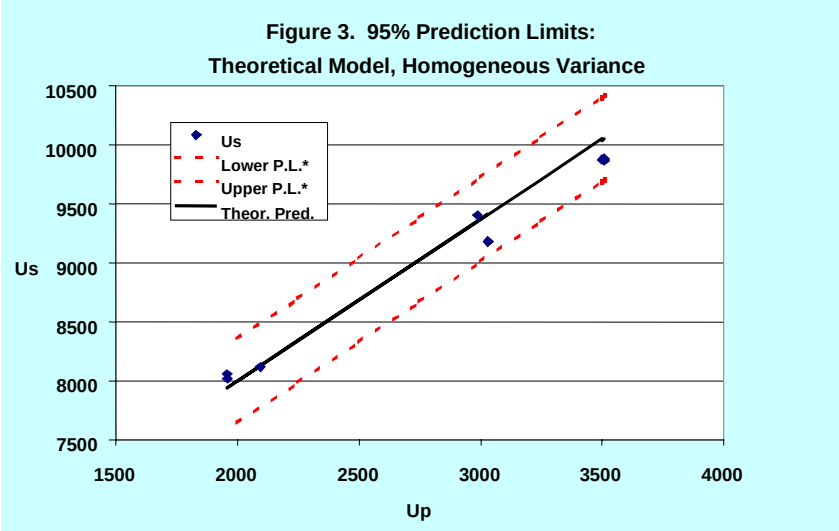
Next I suppose that (experts say that) both the physics model and the statistical model extrapolate on Up from 3000 to 3500 m/s. For the statistical model, this means that the environment for which we need predictions is similar (in terms of the w's at play) to that under which the experiments were conducted, with the exception of the achieved particle velocity. Without this strong assumption, we cannot go beyond the range of the observed data.

Before proceeding with the prediction error calculations, it is pertinent to discuss the possible sources of the observed prediction-error variability. 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 (2000), 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 could subtract it out of the observed variability. This adjustment is illustrated in a later subsection. A third possible contributor to prediction error is the effects of variables not included in the model – the w's. 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. If the prediction uncertainty is deemed 'too large,' it would be appropriate to try to quantify and eliminate some of these sources of variability. On the other hand, we do not want to eliminate sources of variability that would be present in an application for which predictions are desired.

Analyses. The limited amount of data available in this example do not dictate any particular analysis. Thus, I will illustrate a variety of analyses. The options considered include whether predictions are based on the theoretical (approximate) model or on a model tuned to the Table 1 data and whether the prediction errors are modeled as having a standard deviation that is constant over the Up-range of interest or that depends on Up.

a. Theoretical Model, Homogeneous Errors. Under the assumptions that the linear model (7) predicts Us without bias and that prediction errors at Up's in the 2000 – 3500 m/s range are normally distributed, with mean zero and standard deviation, σ, 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 +/- $t_{.025}(6)*s$ = +/- 2.447*146 = +/- 356 m/s. (The quantity, $t_{.025}(6)$, is the .025 quantile on the t-distribution with six df.) Thus, at Up = 3500 m/s, the predicted shock wave velocity for a single future test, at the 95% confidence level, is Us* = 10,051 +/- 356 m/s = (9695, 10,407) m/s. (Another way of expressing this inference is that to be consistent with the available data, at the 95% level, and the assumed physics model, a single future Us at Up = 3500 would have to fall in this interval.) Figure 3 displays the theoretical model, the data, and the calculated prediction limits. By eyeball (ocular) test, one can see some evidence that the theoretical model has a slope a little steeper than the data would suggest. Fig. 3 also shows three additional data points near Up = 3500 m/s (overlapping on the scale in Fig. 3). These points, which provide some validation information on the validation analysis based on the six points in Table 1, are within the prediction limits, but they do provide additional evidence of a lesser slope than is postulated by the theoretical model.



Probably of more interest than a bound on single observations is a bound on the distribution of errors in future predictions. For example, an upper 95% confidence limit on the upper 99th 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. This multiplier of s is obtained as follows: The 99th percentile of a normal distribution centered at zero is 2.33σ. The upper 95% confidence limit on σ, based on 6 df, is s*sqrt(6/χ^2(.05, 6)), where χ^2(.05, 6) is the 5th percentile on the chi-squared distribution with 6 df. This percentile is equal to 1.64 (see standard statistical texts such as Ostle and Mensing 1975). Thus the multiplier of s that provides the upper 95% confidence limit on σ is sqrt(6/1.64) = 1.91. Multiplying this by the factor of 2.33 for the 99th percentile gives the upper 95/99 limit of 4.45s. Thus, at Up = 3500, the inference is that with 95% confidence, 99% of the distribution of Us would fall below 10,701 m/s. If, e.g., failure was defined as Us > 11,000 m/s, these results would tell us that there is good reason to conclude that the failure probability at Up = 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.

b. Theoretical Model, Nonhomogeneous Errors. Because there is some evidence based on the six validation experiments considered here that the error variance may not be constant over the range of Up of interest, I repeated the analysis assuming that σ_{Up}, which is the standard deviation of the prediction errors (around the theoretical model) at Up, is not a constant, but instead is proportional to Up. (I also have the luxury of knowing that the full set of data indicate quite strongly that there is such a pattern.) By methods too tedious to describe in this report, I obtain (trust me) that the estimated σ_{Up} at Up = 3500 m/s is 205 m/s on approximately three df (the estimate depends primarily on the estimated sigma at 3000 m/s, at which point there are three validation data points). 95% prediction limits on a future single observation become Us* +/- 3.182x205 = 10,051 +/- 652 m/s. The upper 95/99 tolerance limit becomes Us* + 6.8x205 = 11,445 m/s. This more conservative, and perhaps prudent, analysis provides substantially more

conservative inferences about future outcomes than does the previous analysis based on the assumption of a constant error-variance over the Up range of interest. With limited validation data, it is difficult to discriminate between different statistical models that could lead to markedly different conclusions. Multiple analyses are required to convey this aspect of total uncertainty. Additional data would be required to provide better discrimination among possible models for the data.

c. Tuned Model. The limited validation data considered (Table 1) hint of bias in the CTH predictions and hence bias in the linear approximation. (As mentioned above, the analysis of the full set of 232 tests reported in Hills and Trucano (2000) shows this bias decisively.) Thus, an alternative analysis is to use the validation test data to fit the parameters, and then carry out an analysis that accounts appropriately for this tuning.

The least-squares fit to the six points in Table 1 is

$$\hat{Us} = 5727 + 1.17*Up, \qquad (8)$$

where the caret denotes an estimate. The residual standard deviation about this fit is 117 m/s, based on four df. Thus, tuning, or updating, improves the fit as measured by the residual standard deviation (reduced from 146 m/s), but the precision of that estimate is reduced (now based on four df rather than six df). As before, a comparison of the three residuals at the low Up values to those at the high Up values suggests that the variance may not be constant. With such limited data, though, the evidence is not conclusive, so I will again do the analysis both ways.

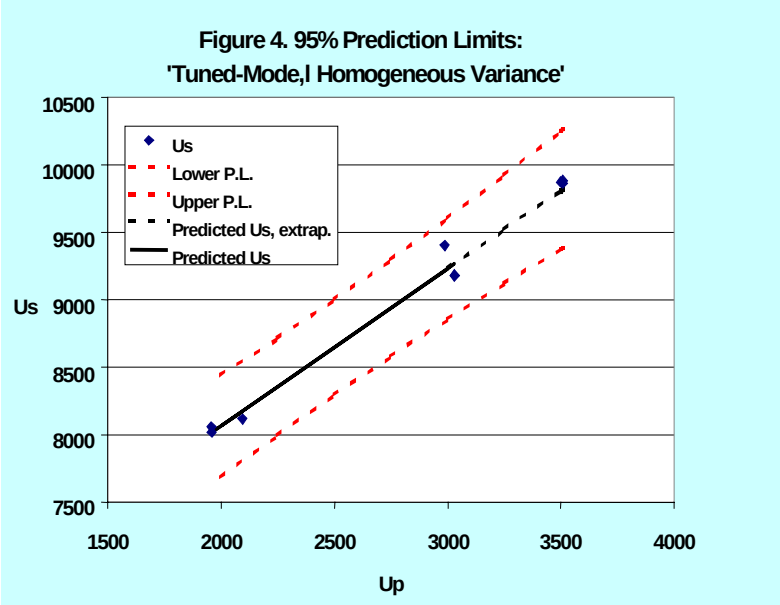
[Note. After tuning, one might be tempted to RMS the six residuals to estimate the residual standard deviation. This would yield $\tilde{\sigma}$ = 96 m/s, but that would be wrong – an under-estimate. The cost of fitting the model to the data is the loss of two degrees of freedom on which to base the estimate of the residual variance.]

Equation (8) is a re-estimation of the approximate linear CTH model (7). Hills and Trucano (2000) use 112 of the 232 (Up, Us) pairs to fit Us as a linear function of Up and this linear function is then used as the EOS internal to CTH. Running CTH with this new EOS results in predictions very much like the fitted EOS.

d. Tuned Model, Homogeneous Errors. At Up = 3500 m/s, the predicted shock wave velocity becomes $\hat{Us}$ = 9813 m/s and the upper and lower prediction-error limits at Up = 3500 m/s, by conventional regression/prediction methodology (Hahn and Meeker 1991), are 9813 +/- 3.74*s = 9813 +/- 437 m/s = (9376, 10,250). This multiplier of 3.74, vs. the 2.447 multiplier for the case of the validated model, reflects the 'tuning penalty.' This penalty is due to the uncertainty of the fitted coefficients, the extent of the extrapolation, and the loss of degrees of freedom in estimating the residual variance. The penalty is partially offset by the reduced estimate of the error-variance.

Figure 4 shows the prediction limits for a single future observation as a function of Up. The curvature in the prediction limits reflects the increasing uncertainty as predictions are

made further from the data used to fit the model, which were the six points for Up in the 2000 – 3000 m/s range. By comparing Fig. 4 to Fig. 3 one can see the steeper slope in the theoretical model vs. that of the fitted model. The three data points (overlapping on this scale) near Up = 3500 m/s are clearly more consistent with the fitted model than the original model. As mentioned, based on the full set of data, Hills and Trucano (2000) rejected the original EOS model and replaced it with a fitted one based on about half of the data.



Consider now the determination of tolerance limits in the 'tuned-model' case. Methods given in Hahn and Meeker (1991) provide the 95/99 upper tolerance limit at Up = 3500 m/s as Us(95/99) = 9813 + 6.4*s = 9813 + 749 = 10,562 m/s. By way of contrast, the multiplier of s in the theoretical-model case is 4.45. Estimation uncertainty, extrapolation, and small residual df all combine to give the larger multiplier of s, namely 6.4. Trying to estimate extreme percentiles by extrapolating a small amount of data, even when facilitated by normality and homogeneous variance assumptions, is an imprecise undertaking. Statistical methods quantify that imprecision appropriately. The difference between these two multipliers shows the benefit of having an acceptable theoretical model, supplemented by a limited amount of data, on which to base predictions and predictive capability, as opposed to an empirical model based on limited data that is also used to measure predictive capability. I.e., when the slope and intercept can be treated as given, more precise predictions are obtained than when the model parameters have to be estimated from the same data used to evaluate prediction error.

e. Tuned Model, Nonhomogeneous Errors. The residuals about the fitted model also suggest that the prediction-error variance increases with increasing Up. Fitting a model that assumes that the error standard deviation is proportional to Up leads to an estimated standard deviation of s = 155 m/s, on 2 df, at Up = 3500. The resulting 95% prediction limits for a single future outcome are equal to 9813 +/- 725. The upper 95/99 tolerance limit is given by 9813 + 10.7x155 = 11,470 m/s. The projected increased variance at Up

= 3500 and the loss of degrees of freedom in estimating that variance lead to substantially more conservative results than the assumed-homogeneous errors model.

Predicting a Probability. Suppose now, for the general case of a linear model, that in a scenario of interest x is a random variable. For example, x could be the B61-11 angle of attack and the scenario of interest would allow this variable to vary over some distribution of weapon-delivery environments. [Note. I'm not assuming the B61-11 computational model is linear; I'm just using some of its variables as illustrations.] Suppose failure occurs if $y > y_f$. To continue the B61-11 example, y might be the shock on a key component, known or assumed to fail at the threshold, y_f. Thus, under the assumed model with specified slope and intercept,

$$\text{Prob(failure)} = \text{Prob}(\alpha + \beta x > y_f) = \text{Prob}(x > (y_f - \alpha)/\beta).$$

Suppose that x is assumed to follow a normal distribution with mean μ_x and standard deviation σ_x (the basis for this assumption could be weapon-delivery simulations, field tests, requirements, or convenience). Then, the computational model's prediction of the above failure probability is:

$$P_f^* = 1 - \Phi(z_f^*), \text{ where} \qquad (9)$$

$$z_f^* = [(y_f - \alpha)/\beta - \mu_x]/\sigma_x \text{ and}$$

$\Phi[z]$ is the cumulative standard normal distribution function.

How good is this prediction?

a. Theoretical Model. Suppose first that n validation experiments over selected values of x support the (fortuitous) conclusions that the data generally agree with the specified model and that the prediction-error, e_x, has a Normal distribution with mean zero, standard deviation σ, constant over all pertinent x. Then, the statistical model for nature is

$$y = \alpha + \beta x + e; \quad e \sim \text{NID}(0, \sigma_e),$$

where $\text{NID}(\mu,\sigma)$ is an abbreviation for "normally and independently distributed with the indicated mean and standard deviation." This means that the deviations from the line, from test-to-test, are (treated as) independent samples from the $N(0, \sigma_e)$ distribution. Given the assumed distribution of x, nature's failure probability is

$$P_f = \text{Prob}(\alpha + \beta x + e > y_f) = \text{Prob}(\beta x + e > y_f - \alpha)$$

$$= 1 - \Phi(z_f), \text{ where} \qquad (10)$$

$$z_f = [(y_f - \alpha)/\beta - \mu_x)]/\sigma_x(1 + \sigma_e^2/\beta^2\sigma_x^2)^{1/2}$$

$$= z_f^*/(1 + \sigma_e^2/\beta^2\sigma_x^2)^{1/2}.$$

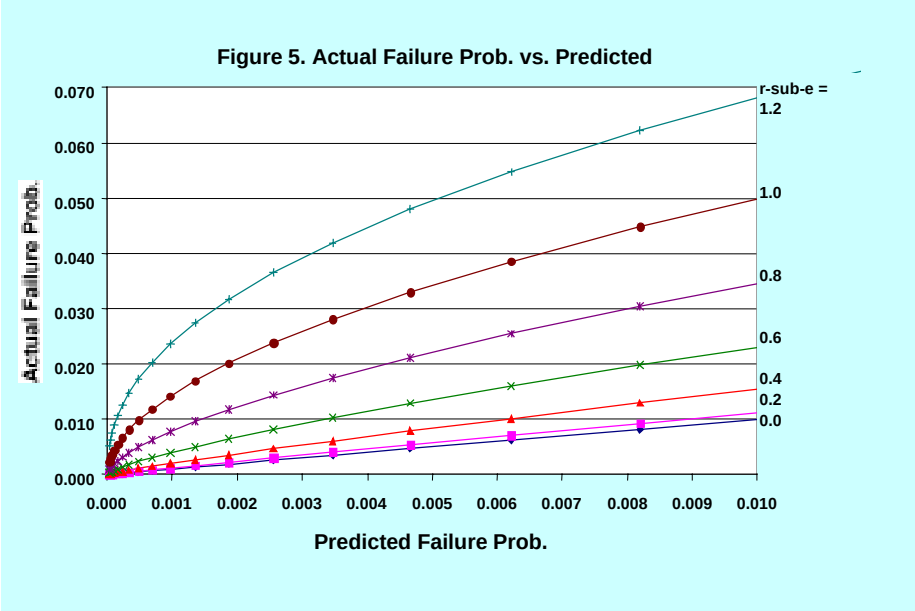
This result is obtained by deriving the distribution of $\beta x + e$ from the assumed distributions of x and e. Replacing σ_e^2 in (10) by its estimate from the validation test data, say s^2, provides a point estimate of P_f and replacing σ_e^2 by upper and lower statistical confidence limits on σ_e^2 provides confidence limits on P_f.

The last equality in (10) shows how z_f^* should be scaled downward in magnitude in order to adjust, or calibrate, the computational prediction to obtain nature's actual probability of failure, given the assumed distribution of x. In reality, nature will have its own distribution of x and the goodness of the assumed distribution as an approximation is another source of prediction error that could be addressed. To limit the scope of this report, I make the analysis conditional on the assumed distribution of x. In the broader context, the probability prediction model is a combination of the computational model and the assumed probability model for nature's distribution of x. Model-validation and prediction-uncertainty quantification would need to address this combined 'super-model.' One can readily imagine situations in which biased predictions were obtained but one could not determine whether the cause was in the scientific model or in the assumed probability model.

For predicted failure probabilities less than 0.50, z_f^* is positive, so dividing z_f^* by a positive number greater than 1.0 leads to a smaller z-value and hence a greater predicted failure probability. This result agrees with intuition because the computational-model prediction, P_f^*, omits the extra-model variation (that is, the variation associated with all the w's) which, when included, results in greater variability than the model captures, which results in a higher probability of exceeding y_f. Thus, the result of incorporating extra-model variability is not a bound on the model-based prediction, but rather a shift in the prediction in the direction of a higher failure probability. (This conclusion pertains to the case of unbiased computational predictions. If e_x has a nonzero expectation, the shift could be in the opposite direction.) Equation (10) also shows that the larger the extra-model, or residual variation, σ_e^2, is relative to the variation accounted for by the model, namely $\beta^2\sigma_x^2$, the greater the adjustment. Conversely, the smaller σ_e^2 is relative to the variation accounted for by the model, the better the model-prediction, P_f^*, is. These results again are intuitive but this example provides a quantification of the effect of unmodeled sources of variability.

Figure 4 plots P_f vs. P_f^* as a function of r_e (written r-sub-e in Fig. 4) = $\sigma_e/|\beta|\sigma_x$ and shows the magnitude by which P_f^* can underestimate P_f. For example, when the model-based prediction is $P_f^* = .005$ and $r_e = .6$, the actual P_f is about .014; at $r_e = 1.0$, P_f is about .035, which means that P_f^* underestimates the failure probability by a factor of seven in this case. In general, Fig. 4 shows that if the residual standard deviation is less than 20% of the standard deviation of the variability captured by the model, the extra-model variability can be ignored. This result can be generalized to other models. If the residual standard deviation is less than 20% of the standard deviation obtained by propagating the distribution of x_r through M, the residual variability can be ignored. However, one first needs to do validation-experimentation in a way that provides a

legitimate estimate of the residual variability in the application of interest in order to see whether it is negligible or not. As discussed, one cannot dismiss residual variability on the basis of a validation hypothesis-test for zero bias.



b. Tuned Model. Suppose now that the model is "tuned" to the validation-test data by a least squares fit:

$$\hat{y} = a + bx.$$

Incorporating the extra-model variability leads to estimating the failure probability by

$$\hat{P_f} = \text{Prob}(a + bx + e > y_f) = 1 - \Phi(\hat{z_f}), \text{ where} \qquad (11)$$

$$\hat{z_f} = [(y_f - a)/b - \mu_x)]/\sigma_x(1 + s_e^2/b^2\sigma_x^2)^{1/2}.$$

By accounting for the uncertainty of the estimates, a, b, and s_e^2, in this expression, confidence bounds on P_f can be obtained. Taylor's series, bootstrap, Bonferroni, or bounding calculations are methods that are candidates for this analysis. (See Hahn and Meeker 1991.)

Example Continued: Probability Prediction. Now, suppose for the sake of illustration that Up in a pellet/plate impact scenario of interest is assumed to be random, with a Normal distribution with mean μ_p = 3500 m/s, and standard deviation σ_p = 100 m/s. Suppose further that the failure threshold is Us =10,500 m/s.

a. Theoretical Model. Based strictly on the original model (which, as discussed, might plausibly be claimed to be valid in the sense that it passes roughly through the center of the data, as shown in Table 1), by propagating the assumed distribution of Up through the

(approximate CTH) model, Us* = 5263 + 1.368Up, the failure probability would be predicted via (9) as

$$P_f^* = 1 - \Phi(z_f^*),$$

where

$$z_f^* = [y_f - (\alpha + \beta\mu_p)]/\beta\sigma_p$$

$$= [(10{,}500 - (5263 + 1.368\text{x}3500)]/1.368\text{x}100$$

$$= 3.28.$$

Thus the failure limit is 3.28 sigmas away from the mean Us, in which case, on the strong assumption of a Normal distribution, the predicted failure probability is .0005. This is just the sort of prediction that is developed from conventional code uncertainty analyses, such as that by Field et al. (1999). But, for the case at hand, this prediction turns out to be quite optimistic; substantial extra-model variability was shown to exist in the validation tests and its effect will be quite significant.

When the estimated error variability of $\hat{\sigma}_e = 146$ m/s at Up = 3500 m/s, for the case of assumed homogeneous residual variability, is added to the model-based variance of $\beta^2\sigma_p^2 = 136.8^2$, the z_f value, eq. (11), becomes $\hat{z}_f = 2.24$. The corresponding 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 σ_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!

For the analysis based on σ_e being proportional to Up, the estimated residual standard deviation is $\hat{\sigma}_e = 205$ m/s and the nominal predicted failure probability is .034 and the upper 95% confidence limit is .28. Clearly, the model-based predicted failure probability can be wildly optimistic when appreciable extra-model variability is present and when there are only limited data available to estimate that extra-model variability. My contention in the previous section that propagating assumed distributions of code inputs through a model does not and cannot measure the uncertainty of using that code to predict nature, even if the code is deemed valid, is not just academic.

b. Tuned Model. For the fitted model, $\hat{Us} = 5727 + 1.17Up$, and the assumption of homogeneous error, the total variance of Us, given the assumed distribution of Up (namely, the Normal distribution with mean 3500 and standard deviation 100 m/s), is estimated by

$$\text{var}(Us) = [1.17^2 100^2 + 117^2] = 165.5^2.$$

The first term in this expression is $b^2\sigma_p^2$ and the second is the estimated error variance. (The equality of the first and second terms was totally accidental.) The predicted mean of Us at Up = 3500 is 9813 m/s. Thus, the failure limit of 10,500 m/s corresponds to $\hat{z}_f$ = (10,500 – 9813)/165.5 = 4.15, which corresponds to a nominal estimated failure probability of 2×10^{-5}. Accounting for the limited data on which the model is fitted (estimates of slope, intercept, and residual variance), by an analysis to tedious for this report, leads to an upper 95% confidence limit on the failure probability of .02, three orders of magnitude greater than the nominal estimate. When this analysis is repeated assuming nonhomogeneous error, the result is a nominal estimated failure probability of 2×10^{-4} with an upper 95% limit of about .20. It is clear in this example that if we want good assurance of a low failure probability, more validation data are needed.

Analysis: Unmeasured x. In the discussion of model/experiment compatibility, I described how prediction uncertainty might be increased if some variables in the model could not be measured in an experiment. The present example can be used to illustrate this situation.

Suppose, hypothetically, that the experimenter said, "I can set up the experiment to provide nominal particle velocities of Up = 2000 m/s and 3000 m/s, but I can't measure the actual velocity achieved in a given test." This situation is analogous to one in which an experiment can be designed to achieve nominal boundary conditions, but the actual conditions cannot be measured. The prediction one would have to use, as discussed above, is to plug the nominal Up's into the model, then analyze the deviations of the observed shock velocities from the nominal prediction. In general one would expect these deviations to be more variable than those observed when Up can be measured because the variability of actual Up's around the nominal would lead to more variation of Us. In terms of the model,

$$y = \alpha + \beta x + e_x,$$

when x is measured, the $\{y - y^*\}$ data provide an estimate of the variance of e_x. When x is not measured, y* becomes a nominal prediction, not a test-specific prediction, and the $\{y - y^*\}$ data provide an estimate of the variance of $\beta x + e_x$, which is $\beta^2\sigma_x^2 + \sigma_e^2$, where σ_x^2 is the variance of the actual x's about the nominal setting. Hence, greater uncertainty is incurred when x-variables in the model cannot be measured in an experiment.

The data in Table 1 can be re-analyzed to illustrate the analysis when some x's are not measured. At nominal values of Up = 2000 and 3000 m/s, under the model Us* = 5263 + 1.368Up, the results in Table 2 are obtained. Note the difference between these Up-nominal prediction errors and those for the Up-specific predictions in Table 1.

Table 2. . Model Validation Test Results, Predictions, and Prediction Errors: Nominal Predictions (All values are in units of m/s.)

Up,nom	Us	Us*	Us – Us*
2000	8054	7999	55
2000	8015	7999	16
2000	8114	7999	115
3000	9401	9367	34
3000	9177	9367	-190
3000	9180	9367	-187

To develop an error model in this case I first do separate analyses at each Up, calculating the mean and standard deviation of each set of three prediction errors, Us – Us*. I also assess the consistency of the errors with a mean of zero by calculating the t-statistic for testing that the expected error is zero. These calculations are summarized in Table 3.

Table 3. Summary of Analysis: Unmeasured Up.

Up,nom	ave.	std. dev	t(0)	sig. level
2000	62	50	2.15	.08
3000	-114	129	-1.54	.13

Table 3 shows some evidence of nonzero means, but not enough to rule out the assumption of zero means at both nominal Up values. Also, as in the previous analysis, there is some evidence of different standard deviations at the two nominal Up values, but with the small sample size, the apparent difference could easily be due to chance. The errors in Table 2 also provide some evidence that the expected errors for the two Up groups of tests are not equal: note the all positive errors at Up = 2000, the mostly negative errors at Up = 3000. An analysis of variance shows that the two average errors are different at about the .09 level of significance, so again there is not strong evidence of a difference.

Thus, one tenable conclusion that could be reached from these data is that the prediction errors are distributed about zero with a common sigma. This error standard deviation is estimated by the RMS of the six errors, namely $\hat{\sigma}_e$ = 122 m/s. Note that although, for reasons discussed above, the *expected* error variance for the unmeasured-x case is larger than that for x-measured, the sample result, in this case, did not turn out that way: the RMS for the six errors in Table 1 was 146 m/s. With the small sample sizes considered here, this reversal is not surprising. The evidence thus is that the variability of actual Up in these tests (all considered to be nominally either 2000 m/s or 3000 m/s) is not a appreciable source of prediction uncertainty. One would not want to spend much money measuring actual Up; nominal is accurate enough.

Now, alternatively, suppose that it was concluded that there is enough evidence of nonzero means that that possibility should be addressed. This conclusion means that, relative to the data, the theoretical model, Us* = 5263 + 1.368*Up-nom, is biased. But this conclusion leads to an indeterminancy: either the nominals are biased (apparently one in one direction, the other in the other) or the model is in error – or both. That is, either the model is bad or the experimenter is unable to deliver the claimed nominal velocities – or some combination. Let the blame-games begin. Additional experimentation would likely be required to resolve the issue. The present limited data cannot.

If the experiment was absolved, then, from the conclusion of bias in the model, it would be appropriate to tune the model to the data by fitting Us as a function of Up-nom. Least squares regression leads to the model,

$$Us\hat{} = 5678 + 1.19\ Up_{nom},$$

which is negligibly different from the model fitted to the (Up, Us) data in Table 1:

$$Us\hat{} = 5727 + 1.17*Up.$$

Prediction uncertainty limits could be obtained by the methods used above in conjunction with this 'tuned' model.

In general, unmeasured x's can lead to inflated prediction uncertainty and ambiguity in interpreting and acting on the results. These can be serious barriers to meaningful quantification of prediction uncertainty. Avoiding or minimizing the problem of unmeasured x's requires the efforts of both experimentalist and modeler.

Measurement-Error Adjustment. The illustrative analyses in this section all were based on the conservative assumption that all of the extra-model variability was due to unmodeled variables and effects in the experiments. In fact, measurement variability is a component of the observed {y – y*} data 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 or known, 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 Us 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 Up) would be $\text{sqrt}(146^2 - 90^2) = 115$ 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 Up (the x-variable in the model), then this would translate into error in Us* with a standard deviation of 1.368*25 = 34 m/s, leading to a further adjustment to a prediction error standard deviation of $\text{sqrt}(115^2 - 34^2) = 110$ 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 – the model can be used as a surrogate for nature.

Comparison to Strictly Test-Based Prediction. Suppose no computational model existed and that it was possible to test near Up = 3500. To be more specific, suppose we do impact tests structured so that nature can deal random Up's from its distribution, which fortuitously, in the best of all worlds, turns out to be the same one as assumed in the preceding analysis. Suppose six such tests were conducted resulting in ave(Us) = 10,051; standard deviation $=(146^2 + \beta^2\sigma_{Up}{}^2)^{1/2}$ = 200m/s (the same numerical results as the original model plus validation data predict for the case of homogeneous errors). Then, for a sample standard deviation of 200, based on 6 observations, the 95% confidence interval on nature's sigma is (124, 490). By way of contrast, given the model and 6 validation tests, substituting the upper and lower end-points of a 95% confidence interval on $\sigma_e{}^2$ into $(\sigma_e{}^2 + \beta^2\sigma_{Up}{}^2)^{1/2}$, obtained from the combined physics and statistical models, yields (166, 349). This latter model-plus-validation-experiments result is substantially more precise than the strictly test-based result. In fact, it would take 16 tests to provide the same precision for test-based estimation. Thus, in this case the model was 'worth' 10 tests, in addition to the 6 validation tests. One can use examples like this to make the economic case for model-based prediction, even including the cost of validation testing.

Where's the Inference? Inference is involved at various stages in this example. First, the finding that the deviations of the experimental data from the model predictions are, e.g., consistent with the model of a Normal distribution with mean zero, and constant standard deviation σ_x is the basis for the inference of prediction error limits associated with subsequent predictions. Other findings about the error distribution would lead to other inferences pertaining to prediction error. The inferences at Up = 3500 m/s are also based on the assumptions of linearity outside the range of the data and that the error variability observed in the validation tests also applies when predictions are made for subsequent experiments outside the range of the data. In terms of the generic situation depicted in Fig. 2 this problem required only a modest extrapolation along the environment axis, as represented by Up, but no inference is claimed for configurations other than the aluminum pellet/plate impacts in the experiments. Not being versed in the physics involved, I will not attempt to justify any further inferences to other contexts.

Note that extrapolation along one axis, as in this example, is probably the simplest form likely to be required. In some situations, as discussed earlier, e.g., radiation effects testing, we may need to extrapolate from tests of aluminum cylinders exposed to various reactor-generated radiation fields to performance of complex electro-mechanical assemblies, such as a neutron generator, when exposed to radiation fields such as would be generated by an underground nuclear explosion. This inferential connection would have to be based on a comparison of the physics in the test situation to the system application of interest and a conclusion that the same predictions would apply. In the B61-11 illustration, the sort of inference that might be required is that prediction errors observed in a series of sled track impact tests with a dummy B61-11 and corresponding computational predictions are representative of the prediction errors for real weapons on real targets. If a satisfactory linkage cannot be established, the sled track tests may leave

us with only the proverbial warm feeling, but no quantification of predictive capability in a system application. One might make ad hoc assumptions, such as that the 'real' residual variability is some multiple of that of the test-environment residual variability, and compute the consequences of such assumptions. Some such subjective scale-up of the prediction uncertainty is better than ignoring it or assuming it is negligible. Another alternative is to conduct more realistic tests and calculate the corresponding computational predictions.

Integration. Suppose for the sake of illustration that α and β in the linear model were provided by two separate sub-models:

$$\alpha^* = A(x_a:\phi_a) \qquad \beta^* = B(x_b:\phi_b).$$

Suppose further that model-validation experimentation for these two sub-models has led to the models for nature's α and β:

$$\alpha = \alpha^* + e_a; \quad e_a \sim N(0, \sigma_a) \qquad \beta = \beta^* + e_b: \quad e_b \sim N(0, \sigma_b),$$

along with estimates of the two sigmas. Then, under the assumption that full-model prediction is simply the combination of sub-model predictions, a combined model for nature's y, based on these two sub-models, would be:

$$y_c = \alpha^* + e_a + (\beta^* + e_b)x = \alpha^* + \beta^* x + e_a + e_b x.$$

Thus, the combined prediction-error distribution at x would have variance,

$$\sigma_c^2 = \sigma_a^2 + x^2\sigma_b^2.$$

From the estimates of σ_a and σ_b, and their corresponding degrees of freedom, σ_c would be estimated from this relationship and the effective degrees of freedom associated with this combined estimate could be obtained by Satterthwaite's method (see, e.g., Ostle and Mensing 1975).

Full-model validation experimentation, as described above, would lead to the model,

$$y_s = \alpha^* + \beta^* x + e_x; \quad e_x \sim N(0, \sigma_x)$$

and σ_x would be estimated directly from these experiments. The assumption under which the sub-model results were integrated is that $e_a + e_b x = e_x$. If there are important sources of variability (w's) that affect nature's y only at the full-model level, and thus were not captured in either of the sub-model experiments, then the integration assumption is not justified and full-model validation experimentation would be required to provide a valid measurement of predictive capability. If it is unclear whether the assumption is justified or not, some full-model experimentation would be required to identify whether additional variation was present. If not, the estimated prediction-error variance from the full-model

experimentation could be pooled with that estimated from the sub-model experimentation.

Experimental Design and Statistical Power. I treated experimental design rather casually in the pellet/plate example, in contrast to its actual importance. Basically, the situation being mimicked was that, for economic reasons and test facility constraints, only limited testing could be done over a Up region that did not extend to the Up of interest. I supposed that six tests could be afforded and selected a design in which three tests were done near each end of the assumed testable Up range. Unstated was the motivation that such a design provides the best (most precise) estimate of a slope, under the assumption of linearity. But, how good would the precision be in that design? Would it be adequate to detect an important error in the computational model's assumed slope of $\beta = 1.368$?

Another design criterion might be the ability to detect nonlinearity. That is, if nature is nonlinear in a situation for which the computational model predicts a linear relationship, a linear extrapolation to Up = 3500 m/s might be seriously in error. The two-point design has no ability to detect nonlinearity. Tests at some mid-range Up points would be required to address this concern. How many? Such questions can be addressed via statistical power analyses.

Consider the slope estimation. If the experimental design is n tests at Up = 2000 m/s and n tests at Up = 3000 m/s, the slope estimate from the resulting data would be

$$b = (\text{Y-bar}_{3000} - \text{Y-bar}_{2000})/1000,$$

where Y-bar_{Up} denotes the average measured Us at the indicated value of Up. Suppose that the standard deviation of measured Us for repeated tests at a fixed Up value is σ. Then, the variance of b is equal to

$$\text{var}(b) = (\sigma^2/1000^2) \text{ x } (2/n).$$

To gauge the agreement of the data with the computational model's slope, we would use the statistic,

$$z = (b - 1.368)/\text{sd}(b),$$

where sd(b) denotes the standard deviation of b, which is the square root of the variance. (Note. For planning purposes, I will treat σ as known. In reality σ will have to be estimated from the 2n tests and this estimate substituted for σ. Similar, but more complicated calculations to what follows can be done for this case. A useful starting point is to work the σ-known case, using what is assumed to be a conservative σ value.)

Of most concern would be a situation in which the model understated nature's slope because the extrapolation to Up = 3500 m/s might be a serious underestimate of the actual Us distribution at that point. Evidence of such a modeling error would be provided by a significantly large positive z-value. Suppose we define such evidence as $z > z_{.05} =$

1.645, the upper 5th percentile of the standard normal distribution. With this criterion, if nature's slope is also equal to 1.368, there is only a 5% chance of wrongly concluding that the model's slope is in error. What we would like is a fairly high probability of concluding the model's slope is in error, when in fact it is. This leads to the statistical concept of power.

Suppose we decide that if that the actual slope is 15% greater than the model's, namely 1.368x1.15 = 1.573, that would lead to serious under-prediction by the model. To protect against failing to detect that great of a difference, suppose we select n so that the probability of detecting a difference this large will be at least .90. Thus, we need to solve for n in the equation

$$\text{Prob}(z > 1.645 \mid \beta = 1.573) = .90. \qquad (12)$$

Substituting for z in (12) gives the equation

$$.90 = \text{Prob}(b - 1.368)/\text{sd}(b) > 1.645 \mid \beta = 1.573).$$

By adding and subtracting 1.573 in the numerator of the left hand side of the inequality, the equation becomes

$$.90 = \text{Prob}(\hat{b} - 1.573 - 1.368 + 1.573)/\text{sd}(\hat{b}) > 1.645 \mid \beta = 1.573)$$

$$= \text{Prob}(z > 1.645 - (.205/\text{sd}(\hat{b}))), \qquad (13)$$

where

$$z = (\hat{b} - 1.573)/\text{sd}(\hat{b}).$$

The random variable z has a standard normal distribution when $\beta = 1.573$. The equality (13) requires that the right hand side of the inequality in (13) be equal to –1.282, the point on the standard normal distribution exceeded with .90 probability. Thus, the equation to solve becomes

$$.205/\text{sd}(\hat{b}) = 1.645 + 1.282.$$

Substituting $(\sigma/1000)\sqrt{(2/n)}$ for $\text{sd}(\hat{b})$, then solving for n yields

$$n = 2 \times [(\sigma/(.205 \times 1000)) \times (1.645 + 1.282)]^2$$

$$= .00041\sigma^2.$$

If we had with good prescience assumed for planning purposes that $\sigma = 150$ m/s (happily close to the estimate of 146 m/s), the solution is n = 9.2. Thus, 10 tests at each end of the Up range would be required to meet the desired statistical power.

By way of contrast, for the actual mimicked experiment with n = 3, the probability of detecting a 15% error in slope is

$$\text{Prob}(z > 1.645 - .205/(.150 \times \text{sqrt}(2/3))) = \text{Prob}(z > -.029) = .51.$$

Thus, there was roughly a 50-50 chance of detecting a 15% error in the original model's slope. Note that the fitted slope of 1.17 is different from the model's 1.368 by about 15% so it is not surprising that the limited illustrative data did not detect an error of this magnitude in the model's slope.

Power can be calculated as a function of n or $(\beta - \beta^*)/\sigma$ and the resulting power curves used to provide an overall picture of the effectiveness of various experimental designs, which means choice of n in this case. With this information trade-offs between experimental cost and effectiveness can be examined in arriving at an experimental design.

For an experimental design that will detect nonlinearity, consider running n tests each at Up = 2000, 2500, and 3000 m/s. A function of the resulting data that measures nonlinearity is

$$D = .5(\text{Y-bar}_{3000} + \text{Y-bar}_{2000}) - \text{Y-bar}_{2500}.$$

For a linear relationship, the expected value of D is zero and the standard deviation of c is $sd(D) = \sigma \times \text{sqrt}(1.5/n)$. Power calculations for the ability of the experiment to detect 'serious' nonlinearity could be conducted as in the preceding paragraphs.

Another experimental design consideration might be the precision with which the standard deviation, σ, of the extra-model variability can be estimated, either at individual x-values, or overall. Various measures of precision, such as the ratio of the upper and lower 95% confidence limits on σ, given n tests, can be analyzed as a function of n to reach a determination of test size.

A much more complex experimental design problem is the simultaneous design of a suite of experiments across different levels of complexity and fidelity, as depicted in Figs. 1 and 2. When there are ways to link the experiments, such as tests for the α-model, the β-model, and the combined $\alpha + \beta x$ model in the example discussed above, then it is possible to analyze trade-offs among various experimental suites. Extending this approach to more complex models and relationships is a research area and the results are apt to be very application-specific.

Summary. In this report I advocate the use of statistical methods to design model-validation experiments and analyze the results to statistically measure predictive capability of computational models. This section illustrates the application of these methods for a simple linear model in some detail. The detail was provided in order to convey the depth and breadth of methods available and to provide an indication of the sorts of methods that will need to be extended to deal with much more complex models.

Concluding Comments

Conclusions. The primary conclusions of this report are listed in the abstract and discussed in the Executive Summary, so they will not be repeated here. In brief, this report has advocated and illustrated the use of model-validation experiments and data to measure the predictive capability of computational models. Without such data, we would have no means of characterizing the uncertainty of a computational-model's point prediction. We would have no uncertainty frame of reference for comparing a computational prediction to a requirement. We would make probability predictions, obtained by propagating assumed input distributions through the computational model, that omit the extra-model variability and thus could consequently be seriously in error. It is critically important, for the sake of credible predictions, that model-validation experiments be designed, conducted, and analyzed in ways that provide for the meaningful measurement of predictive capability.

Programmatic Implications. Implementing the general approach presented in this report for complex codes will be difficult. Model-validation is not just the conduct of a few experiments followed by an overlaying of data. 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 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 in many applications. The required assumptions on which such inferences must be based may be difficult to justify. Solutions to the problems encountered will have to be application-specific, but the general direction must be toward simplification – reduced dimensionality, reasonable approximations. Where solution is not possible, we will have a clear understanding of what the barrier to successful inference is.

Implementing the proposed approach has significant implications for 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 approach presented here provides a framework for that collaboration.

The path forward is thus to 'just do it.' General guidelines can be provided, but progress will come through implementation, not abstract research. By testing the proposed approaches and methods on increasingly difficult problems, we will develop an understanding of their capabilities and limitations.

References

AIAA (1998). *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*, American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, Reston, VA.

Aeschliman, D. P., and Oberkampf, W. L. (1997). Experimental Methodology for Computational Fluid Dynamics Code Validation, Sandia National Laboratories report SAND95-1189, September 1997.

Blackwell, B. F., Gill, W., Dowding, K. J., and Easterling, R. G. (2000). Uncertainty Estimation in the Determination of Thermal Conductivity of 304 Stainless Steel, *Proceedings of IMECE '00*, July, 2000.

Box, G. E. P. (1979). Robustness in the Strategy of Scientific Model Building, *Robustness in Statistics*, ed. by Launer, R. L., and Wilkinson, G. N., 201-236. Academic, New York.

Box, G. E. P., Hunter, W.g., and Hunter, J. S. (1978). *Statistics for Experimenters*, John Wiley and Sons, Inc., New York.

Chiles, J. P., and Delfiner, P. (1999), *Geostatistics, Modeling Spatial Uncertainty*, John Wiley and Sons, Inc., New York

Coleman, H. W., and Stern, F. (1997). Uncertainties and CFD Code Validation, *Journal of Fluids Engineering*, v. 119, 795-803

Easterling, R. G., and Spencer, F. W. (1986) Lower Confidence Bounds on System Reliability Using Component Data: The Maximus Methodology, *Proceedings of the 1984 University of Missouri Conference on Reliability and Quality Control*.

Ferson, S., and Ginzbirg, R. (1996). Different Methods are Needed to Propagate Ignorance and Variability, *Reliability Engineering and System Safety*, v. 54, 133-144.

Field, R. V., Red-Horse, J. R., and Paez, T. L. (1999). Nondeterministic Analysis of the B61-11 for Internal Component Qualification Using Computational Probabilistic Methods, internal Sandia National Laboratories memorandum, Dec. 20, 1999.

Gibbons, J. D., and Pratt, J. W. (1975). P Values: Interpretation and Methodology, *The American Statistician*, v. 29, no. 1, 20 – 25, February, 1975.

Hahn, G. J., and Meeker, W. Q. (1991). *Statistical Intervals*, John Wiley & Sons, Inc., New York.

Hasselman, T., and Anderson, M. (1999). A MATLAB Toolbox for Nonlinear Model Validation and Calibration, presentation slides, August 19, 1999.

Hills, R. G., and Trucano, T. G. (1999). Statistical Validation of Engineering and Scientific Models: Background. Sandia Report SAND99-1256, May 1999.

Hills, R. G., and Trucano, T. G. (2000). Statistical Validation of Engineering and Scientific Models with Application to CTH. Draft Sandia National Laboratories Report.

Johnson, P.A. (1995). Comparison of Pier Scour Equations Using Field Data, *ASCE Journal of Hydraulic Engineering*, v. 121, 626-629.

Kaplan, S., and Garrick, B. J. (1981). On the Quantitative Definition of Risk, *Risk Analysis*, v.1, 11-27.

McGlaun, J. M., Thompson, S. L., and Elrick, M. G. (1990). CTH: A Three-Dimensional Shock Wave Physics Code, *Int. J. Impact Engng.* v. 10, 350-360.

Moya, J.L. (1998). Strategic Computing & Simulation Validation & Verification Program Draft Implementation Plan, version 1.0 (unpublished, available on Sandia internal restricted network web site).

Oberkampf., W. L. (2000). Verification and Validation in Computational Sciences, seminar presentation, Jan. 13, 2000.

Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F. (1999). A New Methodology for the Estimation of Total Uncertainty in Computational Simulation, presented at AIAA Non-deterministic Approaches Forum, paper no. AIAA-99-1612, April 12-15, 1999, 23 pages.

Oberkampf, W. L., and Trucano, T. G. (2000). Validation Methodology in Compuptational Fluid Dynamics, AIAA 2000-2549, presented at Fluids 2000 Conference, June, 2000.

Oreskes, N., Shrader-Frechette, K., and Belitz, K. (1994). Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences, *Science*, v. 263, Feb. 4, 1994, 641-646.

Ostle, B. O., and Mensing, R. W. (1975). *Statistics in Research (3rd ed.)*, Iowa State University Press, Ames.

Parzen, E. (1962). *Stochastic Processes*, Holden-Day, San Francisco.

Pilch, M., Trucano, T., Moya, J. L., Froehlich, G. Hodges, A., and Peercy, D. (2001). Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format: Version 2.0. Sandia National Laboratories Report SAND2000-3101

Popper, K. R., *Conjectures and Refutations: The Growth of Scientific Knowledge*, Routledge and Kegan, London, 1969.

Red-Horse, J. R., Paez, T. L., Field Jr., R. V., and Romero, V. (2000). Nondeterministic Analysis of Mechanical Systems, Sandia National Laboratories report SAND2000-0890.

Romero, C. A. (2000). The Increasing Role of Numerical Simulation in Stockpile Certification, draft SAND report, March 1999.

Trucano, T. G. (2000). Formalism for "Validation Metrics," presentation slides, December 5, 2000.