LA-UR- 03-8491

Approved for public release;
distribution is unlimited.

Title:

UNCERTAINTY QUANTIFICATION AND THE VERIFICATION AND VALIDATION OF COMPUTATIONAL MODELS (U)

Author(s):

François M. Hemez, Los Alamos National Laboratory, ESA-WR

Submitted to:

Damage Prognosis for Aerospace, Civil and Mechanical Systems,
Edited by D.J. Inman, C.R. Farrar, V. Lopes Jr., and V. Steffen Jr.,
John Wiley & Sons Ltd., London, United Kingdom,
December 2004

Los Alamos
NATIONAL LABORATORY

*Published in **Damage Prognosis for Aerospace, Civil and Mechanical Systems**, Edited by D.J. Inman, C.R. Farrar, V. Lopes Jr., and V. Steffen Jr., John Wiley & Sons Ltd., London, United Kingdom, December 2004.*

UNCERTAINTY QUANTIFICATION AND THE VERIFICATION AND VALIDATION OF COMPUTATIONAL MODELS

François M. Hemez[a]

Engineering Sciences and Applications, ESA-WR
Los Alamos National Laboratory, Mail Stop T001
Los Alamos, New Mexico 87545, U.S.A.

ABSTRACT

This chapter offers a brief overview of the technology developed at Los Alamos National Laboratory, among other places, in support of engineering verification and validation programs. The material presented is based to a large extent on a tutorial taught at the Los Alamos Dynamics Summer School [1]. The chapter overviews the concepts and introduces methods useful to assess the predictive accuracy of numerical simulations. The technology available for code verification, solution verification, test-analysis correlation, meta-modeling, and calibration is discussed. The quantification of uncertainty, both in the forward mode ("*What is the effect of uncertainty on predictions?*") and inverse mode ("*Where is an observed variability coming from?*") is also addressed. Rather than providing a detailed account of the state-of-the-art, techniques are illustrated using engineering applications.

1. INTRODUCTION

In computational physics and engineering, numerical models are developed to predict the behavior of a system whose response cannot be measured experimentally. A key aspect of science-based predictive modeling is to assess the ***credibility*** of predictions. Credibility, which is usually demonstrated through the activities of model Verification and Validation (V&V) refers to the extent to which numerical simulations can be analyzed with ***confidence*** to represent the phenomenon of interest [2].

One can argue, as it has been proposed in recent work [3], that the credibility of a mathematical or numerical model must combine three components: 1) An assessment of fidelity to test data; 2) An assessment of the robustness of prediction-based decisions to variability, uncertainty, and lack-of-knowledge; and 3) An assessment of the prediction accuracy of the models in situations where test measurements are not available. Unfortunately, the three goals are antagonistic. In Reference [3], a Theorem is proven that illustrates for a wide class of uncertainty models the irrevocable trade-off between robustness-to-uncertainty, fidelity-to-data, and confidence-in-prediction.

The three aforementioned assessments nevertheless require a similar technology in terms of model validation and quantification of uncertainty. Even though V&V in Structural Dynamics is rapidly evolving and open research to a great extent, the intent of this publication is to overview the technology developed at Los Alamos National Laboratory (LANL) in support of V&V activities for engineering applications. After a brief introduction of the main concepts of predictive accuracy, the discussion focuses on verification, validation, and the quantification of uncertainty for numerical simulations. The main references are provided and the techniques are illustrated with examples.

2. CONCEPTS

Even though the conventional activities of model V&V are generally restricted to improving the fidelity-to-data through the correlation of test and simulation results and the calibration of model parameters [4-5], the other two components are equally important. The main reason is that optimal models—in the sense of models that minimize the prediction errors with

[a] Technical Staff Member and ESA-WR Validation Methods Team Leader; Phone: 505-663-5204; E-mail: hemez@lanl.gov. This manuscript has been approved for unlimited, public release on October 16, 2003. LA-UR-03-8491. **Unclassified**.

respect to the available test data—possess exactly zero ***robustness*** to uncertainty and lack-of-knowledge [6]. This means that small variations in the setting of model parameters, or small errors in the knowledge of the functional form of the models, can lead to an actual fidelity that is significantly poorer than the one demonstrated through calibration.

Clearly, fidelity-to-data matters because no analyst will trust a numerical simulation that does not reproduce the measurements of past experiments or the information contained in historical databases. Robustness-to-uncertainty is equally critical to minimize the vulnerability of decisions to uncertainty and lack-of-knowledge. It may be argued, however, that the most important aspect of credibility is the assessment of confidence-in-prediction, which is generally not addressed in the literature. Assessing the confidence-in-prediction here refers to an assessment of prediction error away from settings where physical experiments have been performed, which must include a rigorous quantification of the sources of variability, uncertainty, and lack-of-knowledge, and their effects on model-based prediction. The concepts of fidelity-to-data, robustness-to-uncertainty, and prediction confidence are illustrated in Figure 1.

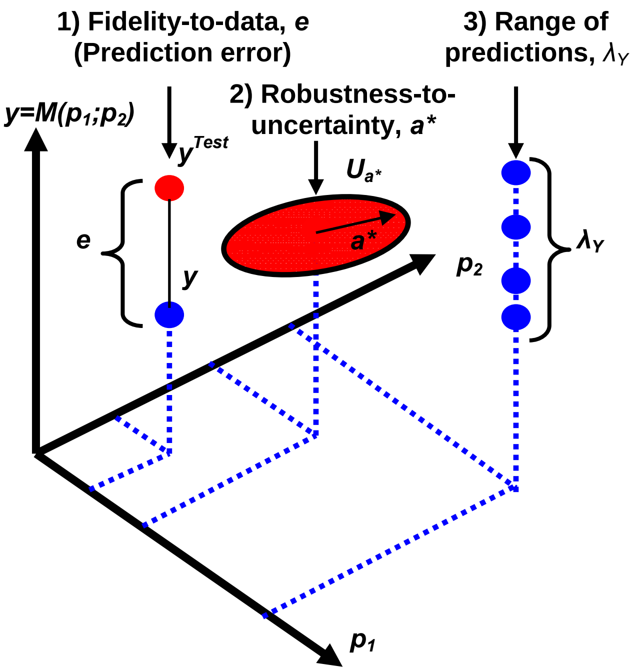


Figure 1. Illustration of the concepts of fidelity-to-data, robustness-to-uncertainty, and prediction accuracy (range, "looseness," or confidence).

It is emphasized that, because this is work-in-progress to a great extent, the concept of prediction accuracy denoted in Figure 1 by the symbol λ_Y is somewhat broad. It is analogous to a range of predictions, or "looseness." Clearly, predicting a range of values relates to the notion of confidence that one has in the ability to make accurate predictions. The notion of accuracy λ_Y is further discussed below. It is believed that future research will narrow down this definition, but a standard accepted throughout the scientific community is not, to the best of our knowledge, currently available.

Throughout the manuscript, the numerical simulation is represented conceptually as a "black-box" input-output relationship between inputs p_k and outputs y_k. In the case, for example, of the relationship between two inputs $(p_1;p_2)$ and a single output y, we write:

$$y = M(p_1;p_2) \tag{1}$$

A domain such as $[p_1^{(min)};p_1^{(max)}]$ x $[p_2^{(min)};p_2^{(max)}]$ represents the design space over which predictions must be obtained. Such requirement implies that the prediction accuracy must be established for all settings $(p_1;p_2)$ in the design domain $[p_1^{(min)};p_1^{(max)}]$ x $[p_2^{(min)};p_2^{(max)}]$.[1]

Fidelity-to-data represents the distance *e*—assessed with the appropriate metrics, possibly statistical tests if probabilistic information is involved—between physical measurements y^{Test} and simulation predictions *y* at a setting $(p_1;p_2)$:

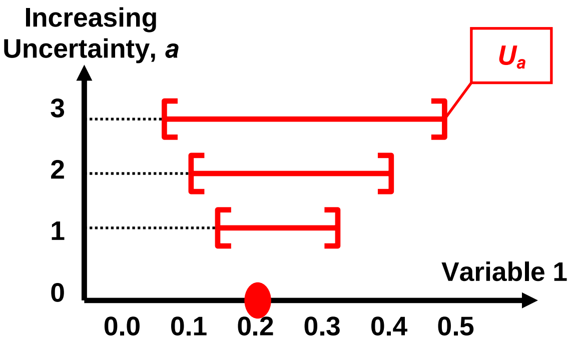
$$e = \left\| y^{Test} - y \right\| \tag{2}$$

Fidelity-to-data is pictured in Figure 1 as the vertical distance between a measurement y^{Test} and a prediction *y* for a physical experiment and a numerical simulation performed at the same setting $(p_1;p_2)$.

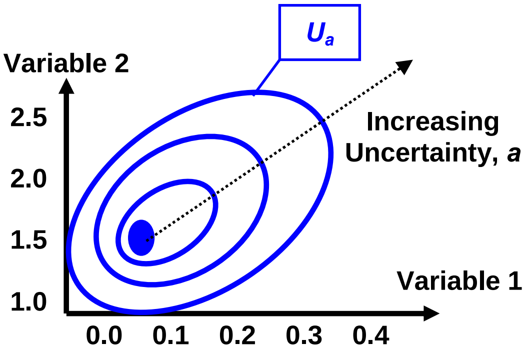
Robustness-to-uncertainty here refers to the range of settings $(p_1;p_2)$ that provide no more than a given level of prediction error e_{Max}. The concept of robustness is illustrated in Figure 1 by showing a subset U_a of the design domain $[p_1^{(min)};p_1^{(max)}]$ x $[p_2^{(min)};p_2^{(max)}]$. The significance of the concept of robustness-to-uncertainty is that

[1] The input parameters $(p_1;p_2)$ represent settings such as, for example, the angle of attack and flow velocity of an aero-elastic simulation that predicts a coefficient of lift $y=C_L$. Another example is the response of a building to Earthquakes. The input parameters might represent the amplitude and frequency contents of the excitation, and the output prediction might be a peak level of structural stress occurring in the structure.

all predictions made for settings *(p_1;p_2)* chosen inside the domain U_a are guaranteed not to exceed the error level e_{Max}. The *a*-parameter represents the "size" of the domain U_a. The definitions of the sizing parameter (*a*) and corresponding domain (U_a) are arbitrary at this point because the purpose of this discussion is to introduce concepts. The only constraint to satisfy is that increasing values of the sizing parameter *a* must define nested domains U_a, as shown in Figure 2. References [3, 7] define the families of domains U_a as convex sub-spaces. This choice allows the analyst to accommodate a wide variety of uncertainty and lack-of-knowledge models.[2]



(2-a) Illustration of 1D nested intervals.



(2-b) Illustration of 2D nested ellipsoids.
Figure 2. Concept of nested, convex subsets.

Clearly, a large robustness-to-uncertainty (*a*) is more desirable than a small one (*b*) because the former subspace will encompass all events defined in the latter one, or $U_b \subset U_a$. A large robustness indicates that potentially large uncertainty and lack-of-knowledge does not deteriorate the prediction error by more than e_{Max}. Generally, a trade-off must be decided upon between the robustness-to-uncertainty (*a*) and prediction error (e_{Max}), or fidelity-to-data. Studying such trade-off is the basic concept of the information-gap theory for decision-making under severe uncertainty [7].

Finally, the symbol λ_Y in Figure 1 refers to the range of predictions made by a family of potentially different "models." The importance of λ_Y stems from the fact that, to have confidence in predictions, there should be as much consistency as possible between the predictions provided by equally credible sources of information. Confidence is generally increased when different sources of evidence all reach the same conclusion. The concept of confidence-in-prediction is illustrated in Figure 1 by showing a range λ_Y of predictions obtained when different "models" are exercised to make predictions at a setting *(p_1;p_2)* where no test data are available.

The ultimate goal of model validation is to establish predictive confidence by estimating the range of predictions λ_Y (or, equivalently, the lack-of-consistency) provided by equally credible sources of information. The range of predictions is related to the notion of confidence through, for example, the use of statistical testing. Note that the terminology "model" is here defined in a broad sense. In any realistic application, sources of evidence include expert judgment, back-of-the-envelope calculations, measurements, and predictions obtained from phenomenological models or high-fidelity simulations. All available sources of information must be taken into account to assess the credibility of predictions. It is equally important to understand, quantify, and eventually combine the uncertainty associated with each source of information.

The remainder of this chapter is devoted to discussing where uncertainty arises in numerical simulations, how it may be identified, and how to quantify it. It will become clear that it may not be possible to always describe uncertainty using the theory of probability. Defining a framework for information integration within the Generalized

[2] A first example is a probabilistic model of variability where the values of coefficients in the covariance matrix are controlled by the parameter *a*. A second example is a possibility structure π defined to represent a lack-of-knowledge, where the size of intervals is proportional to the parameter *a*. A third example is a family of fuzzy membership functions defined to represent expert judgment and linguistic ambiguity, where the membership functions are parameterized by the uncertainty parameter *a*.

Information Theory is currently an area of active research [8-9], although not addressed here.

3. VERIFICATION ACTIVITIES

Because our intent is to substitute numerical simulations for information that cannot be obtained experimentally, the predictive accuracy of models upon which simulations rely must be established through V&V. ***Verification*** basically refers to the assessment that the equations implemented in the computer code are solved correctly, no programming error is present, and the discretization leads to converged solutions both in time and space. ***Validation***, on the other hand, refers to the adequacy of a model to describe a particular physical phenomenon.

Numerical results from, for example, finite element simulations, provide approximations to sets of coupled partial differential equations. Before the validity of the equations themselves can be assessed, verification must take place to guarantee the quality of the solution. Verification is formally defined as "*the process of determining that the implementation of a model accurately represents the conceptual description of the model and its solution*" [10]. The primary sources of errors in computational solutions are inappropriate spatial discretization, inappropriate temporal discretization, insufficient iterative convergence, computer round-off, and computer programming. Verification quantifies errors from these various sources, and demonstrates the stability, consistency, and accuracy of the numerical scheme. The main activities, namely, code verification and calculation verification, are briefly overviewed in this section.

3.1 Code Verification

Code verification can be segregated into two parts: numerical algorithm verification, which focuses on the design and underlying mathematical correctness of discrete algorithms for solving partial differential equations, and Software Quality Assurance (SQA), which focuses on the implementation of the algorithms.

Software Quality Assurance

SQA determines whether or not the code as a software system is reliable (implemented correctly) and produces repeatable results with a specified environment composed of hardware, compilers, and libraries. It focuses on the code as a software product that is sufficiently reliable and robust from the perspective of computer science and software engineering.

Analysis and testing basically rests in three techniques: static analysis, dynamic testing, and formal testing [11]. Static analysis techniques analyze the form, structure, and consistency of the code without executing the code. Examples of static analysis techniques are software reviews, complexity analysis, inspections, and analyses of data flows. Dynamic testing involves the execution of the code. Examples include regression testing, black-box testing, and glass-box testing. The results of executing the code help to diagnose coding errors or weaknesses in design that can cause coding errors. Formal testing methods are directed toward rigorously demonstrating that the code exactly represents the underlying conceptual model.

It is emphasized that successful SQA plans must be defined before and implemented during the development of the product, rather than being viewed as an activity that takes place after the software has been developed. Whether SQA is a legitimate V&V activity is still debated in the Structural Dynamics community. Our current opinion is that it is not because SQA activities cannot presume the intended purpose of the software. Code users should nevertheless be cognizant and enforce the implementation of sound SQA practices for the software they use.

Numerical Algorithm Verification

Numerical algorithm verification addresses the reliability of the implementation of all of the algorithms that affect the numerical accuracy and efficiency of the code. In other words, this verification process focuses on how correctly the numerical algorithms are programmed in the code. The major goal is to accumulate sufficient evidence to demonstrate that the numerical algorithms in the code are implemented correctly and functioning as intended.

The study of error estimation and numerical algorithm verification is fundamentally empirical. Numerical algorithm verification deals with careful investigations of topics such as spatial and temporal convergence rates, iterative convergence, independence of solutions to coordinate transformations, and symmetry tests related to various types of boundary conditions. Such error evaluation is clearly distinct from error estimation that deals with approximating the numerical error for particular applications when the correct solution is not known [12].

The principal components of this activity include the definition of appropriate test problems for evaluating solution accuracy, and the determination of satisfactory performance of the algorithms on the test problems. Numerical algorithm verification rests upon comparing computational solutions to the “correct” answer, which is provided by highly accurate solutions for well-chosen test problems. Figure 3 pictures a non-linear pendulum verification problem. The equation that governs the pendulum angle as a function of time is given by:

$$\frac{d^2\theta(t)}{dt^2}+\lambda^2\sin(\theta(t))=0 \qquad (3)$$

where $\lambda^2=(g/L)$, *L* is the length of the pendulum, and *g* is the gravitational acceleration constant. A highly accurate solution of equation (3) can be compared to the results obtained from a finite element analysis that, for example, represents the pendulum as a rigid body. Statistics of the difference between the two solutions establish the reliability of algorithms for this problem only.

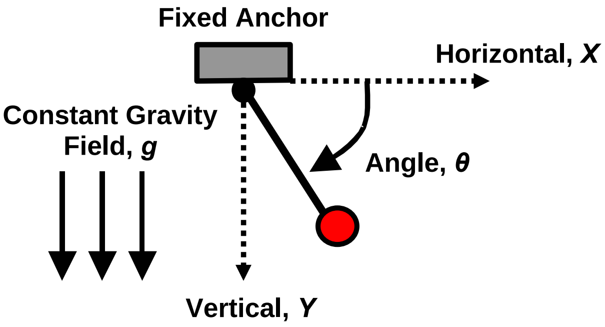


Figure 3. Pendulum verification problem.

It is important to understand that this activity provides evidence of the verification from the ***intended purpose*** of the code. “Intended purpose” draws a clear boundary between SQA and numerical algorithm verification. The suite of problems selected to verify the software should exercise all the important dynamics and solution procedures that will be put into play when solving the engineering application of interest.

The main challenge here is to develop test problems for which analytical or highly accurate solutions can be obtained.[3] A technique for developing a special type of analytical solutions is the Method of Manufactured Solutions (MMS) [13-14]. The MMS provides custom-designed verification test problems of wide applicability. Using the MMS in code verification requires that the computed source term and boundary conditions are programmed into the code, and that a numerical solution is computed. Although the intrusive character of the MMS can be viewed as a limitation, this technique nevertheless verifies a large number of numerical aspects in the code, such as the numerical method, differencing technique, spatial-transformation technique for grid generation, grid-spacing technique, and correctness of algorithm coding. Applications in Fluid Dynamics illustrate that the MSS can diagnose errors very efficiently, but cannot point to their sources, nor does it identify algorithm-efficiency mistakes [13].

3.2 Solution Verification

Solution verification basically deals with the quantitative estimation of numerical accuracy of a given solution. The primary goal is attempting to estimate the numerical accuracy of a given solution, typically for a non-linear system of partial differential equations with singularities or discontinuities. Numerical accuracy is critical in computations used for validation activities, where one should demonstrate that numerical errors are insignificant compared to test-analysis correlation errors.

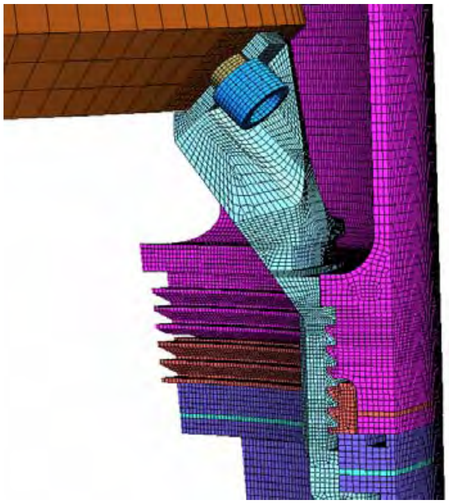


Figure 4. Detail of a geometry discretization.

Typical issues of computational grids are suggested in Figures 4 and 5. Figure 4 shows that the resolution provided by the grid must be

[3] The equation (3), for example, can only be solved analytically for angles $\theta\approx(\pi/2+/-\varepsilon)$, where ε is “small.”

appropriate to avoid stress concentrations at the "corners" of a complex geometry. Also, the peak wave speed that can be captured by a given mesh is $C_{max}=(h/\Delta t)$, where h and Δt represent a characteristic element size and time increment. Modeling any phenomenon that could propagate information at a velocity $C>C_{max}$ requires another mesh or time integration, to satisfy $(h/\Delta t)>C$. A third, typical issue is illustrated in Figure 5 that compares a distorted mesh to a more regular one. Elements with poor aspect ratios tend to provide low-quality approximations.

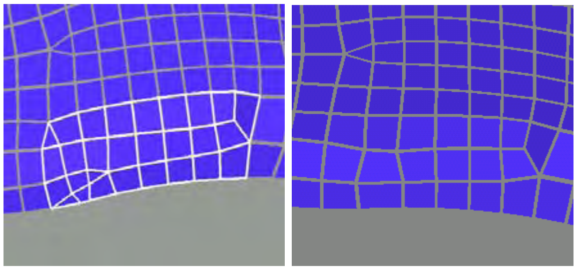


Figure 5. Comparison between a poor-quality mesh (left) and a good-quality mesh (right).

The two basic approaches for estimating the error in a numerical solution are a priori and a posteriori approaches. A priori approaches use only information about the numerical algorithm that approximates the partial differential operator and the given initial and boundary conditions. A priori error estimation is a significant element of numerical analysis for differential equations, especially those underlying the finite element and finite volume methods [15-16].

A posteriori approaches use all of the a priori information, plus computational results from numerical solutions using the same numerical algorithm on the same system of partial differential equations and initial and boundary data. Computational results are generally provided as sequences of solutions on consecutively finer grids. The framework upon which solution verification techniques rely is that the true-but-unknown solution of the continuous equations, or y_C, is equal to the solution $y(h)$ of the discretized equations, plus an error assumed to be proportional to the rate of convergence:

$$y_C = y(h) + \alpha h^p + O(h^{p+1}) \tag{4}$$

The discretization parameter h can represent a characteristic mesh size or a time step. Equation (4) forms the basis for estimating or verifying the order of convergence p, and verifying that the approximation $y(p)$ converges to the continuous solution y_C. The discussion below focuses on a posteriori error estimates because they provide quantitative assessments of numerical error in practical cases of non-linear equations.

The Richardson Extrapolation

A posteriori error estimation has primarily been approached through the use of Richardson extrapolation [13] or estimation techniques with finite element approximations [17]. Richardson's method can be applied to both finite difference and finite element methods. It computes error estimates of dependent variables at all grid points, as well as error estimates for solution functionals. It is emphasized that different dependent variables and functionals converge at different rates as a function of grid size or time step. Error estimation should be carried out mindful of the response outputs of interest.

Equation (4) features three unknowns: the continuous solution y_C; the order of convergence p; and a constant α. If the order of convergence cannot be assumed, three equations are needed at a minimum to estimate the triplet $(y_C;p;\alpha)$.[4] Therefore, the extrapolation starts by computing the three numerical solutions obtained with three resolutions h_C, h_M, and h_F:

$$\begin{aligned} y_C &\approx y(h_C) + \alpha h_C^p \\ y_C &\approx y(h_M) + \alpha h_M^p \\ y_C &\approx y(h_F) + \alpha h_F^p \end{aligned} \tag{5}$$

The subscripts identify the "coarse," "medium," and "fine" resolutions, respectively. The order of convergence p can then be estimated as:

$$p \approx \frac{\log\left(\dfrac{y(h_M)-y(h_C)}{y(h_F)-y(h_M)}\right)}{\log(r)} \tag{6}$$

where r denotes the ratio between successive refinements, $r = (h_C/h_M) = (h_M/h_F)$. It is important that r be kept constant for estimation (6) to be valid. Finally, the true-but-unknown solution y_C of the continuous partial differential equations can be approximated as:

$$y_C \approx \frac{r^p y(h_F) - y(h_M)}{r^p - 1} \tag{7}$$

[4] Note that, if the order of convergence p is known a priori, only two numerical solutions are required to estimate the continuous solution y_C. It is strongly recommended, however, to verify the actual order of convergence. Factors such as programming errors, stress concentrations, and non-linearity can severely deteriorate, at least locally, the order of convergence.

Posterior error indicators such as the grid convergence index discussed next can be calculated to estimate how far the numerical approximation is from the continuous solution.

The Grid Convergence Index

A Grid Convergence Index (GCI) based on Richardson's extrapolation has been developed to assist in the estimation of grid convergence error [13, 18]. The GCI converts error estimates that are obtained from any grid-refinement ratio into an equivalent grid-doubling estimate. Recent studies have established the reliability of the GCI method, even for solutions that are not asymptotically convergent [19-20].

The definition of the GCI in Reference [13] involves two solutions *y(h$_C$)* and *y(h$_F$)*, obtained with coarse and fine resolutions, respectively:

$$\mathrm{GCI} = 100 \left| \frac{y(h_C) - y(h_F)}{y(h_C)} \right| \left(\frac{\beta}{r^p - 1} \right) \qquad (8)$$

where β is a constant that adds conservatism to the formula, typically, $1 \leq \beta \leq 3$. Small values of the GCI—typically, less than 1%—indicate that the best approximation obtained from the two resolutions is close to the true-but-unknown continuous solution.

Applying the Richardson extrapolation and GCI concepts would typically start by performing three finite element calculations, then verify the order of convergence (7), and verify asymptotic convergence (8). Again, it is emphasized that the convergence study can verify the adequacy of a mesh (when *h* is defined as the grid size), just like it can be applied to time or frequency-domain convergence (when *h* is a time step or frequency increment). In the case of non-linear, transient dynamics simulations, mesh and time convergence should be verified independently of one another, which could lead to significant computational demands.

Table 1. Solution convergence results.

Criterion	**Value Estimated**	
Order (*p*)	1.0093	
GCI	**Coarse-to-medium**	**Medium-to-fine**
	5.25 10^{-2} %	5.13 10^{-3} %

Table 1 illustrates the results of a GCI study for the pendulum problem of Figure 3. An Euler forward time integration scheme is implemented to integrate equation (3) with three successive increments of *Δt*=10^{-2} second, *Δt*=10^{-3} second, and *Δt*=10^{-4} second.[5] It can be read that the estimated order of convergence is very close to one, its theoretical value. In addition, the GCI values demonstrate asymptotic convergence. A 10-fold refinement—going from *Δt* to (*Δt*/10)—combined with an order of convergence equal to one decreases the GCI by a factor of $(r)^p \approx 10$, which is what can be observed in Table 1.

3.3 What Can be Expected From Verification?

The rigorous verification of a code requires "proof" that the computational implementation accurately represents the conceptual model and its solutions. This, in turn, requires proof that the algorithms provide converged solutions of these equations in all circumstances under which the code will be applied. It is unlikely that such proofs will ever exist for general-purpose computational physics and engineering codes. Verification, in an operational sense, becomes the absence of proof that the code is incorrect.

In this definition, verification activities consist of accumulating evidence substantiating that the code does not have any apparent algorithmic or programming errors, and that it functions properly on the chosen hardware and system software. This evidence needs to be documented, accessible, repeatable, and capable of being referenced. The accumulation of evidence also serves to reduce the regimes of operation of the code where one might possibly find such errors.

In the absence of formal proof, what can be expected from the verification activities? Clearly, evidence must be provided that the code does not have any apparent error, and that it provides approximate solutions of acceptable accuracy, consistent with the purpose intended. How much evidence should be provided (in other words, "*How good is good enough?*") is not addressed here because such question directly relates to accreditation and the definition of a standard. It is also application-specific to a great extent.

4. VALIDATION ACTIVITIES

In this section, typical validation activities are presented. To illustrate the relationship between

[5] The finite element analysis simulates a rigid rod, therefore, the only aspect that needs to be verified is the time integration algorithm. Other aspects of the code, such as an element formulation or a contact algorithm, can be verified independently.

fidelity-to-data, uncertainty quantification, and predictive accuracy, these activities are put in the context of a material test instead of being discussed separately.

Material scientists commonly use an experiment known as the Taylor anvil impact to develop constitutive and equation-of-state models. Because of the regimes for which they are developed, such models generally include plasticity, high strain-rate and temperature dependency. Examples used in engineering mechanics include the Johnson-Cook and Zerilli-Amstrong models [21-22]. The Taylor anvil experiment consists of impacting a sample of material against a rigid wall and measuring its deformed profile. The measured profiles are compared to numerical predictions and parameters of the constitutive equations can be calibrated to improve the accuracy of the model.

A suite of validation experiments is designed, starting with Hopkinson bar testing and proceeding with the higher strain-rate Taylor tests, to assess the predictive accuracy of material models over different regions of the operational space. Based on test-analysis correlation and statistical meta-modeling, the predictive accuracy of a constitutive model is assessed in regions of the operational space where testing is not possible. The significance of this assessment is that the fidelity-to-data can be estimated, even before performing the calculation itself. The suite of tools developed for model validation helps analysts decide whether their models meet the accuracy requirement for a particular application.

4.1 The Validation Domain

Before proceeding with the description of the validation steps, the notion of validation domain must be introduced. Generally speaking, a numerical simulation is always developed to analyze a given operational domain because point-predictions, that is, models that cannot be parameterized or modified, are not very useful.

The constitutive model investigated here is developed to run numerical simulations at various combinations of strain-rates and temperatures. For our application, these two input variables define the operational space of interest. The validation domain is simply defined as the region of the operational space where the mathematical or numerical model provides acceptable accuracy for the application of interest. This concept is illustrated in Figure 6. Simply speaking, validation is achieved when the predictive accuracy of the model has been assessed within the operational domain, a consequence of which is the identification of the region—or validation domain—that provides sufficient accuracy.

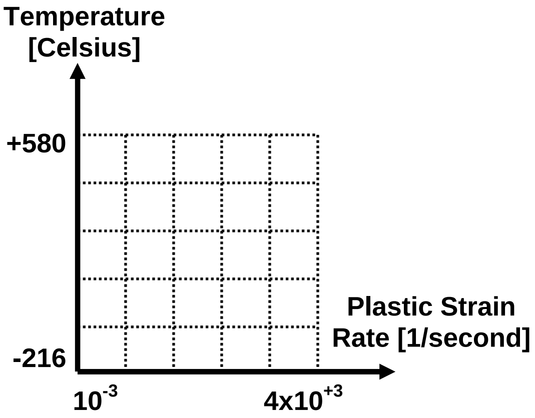


Figure 6. Definition of a validation domain.

4.2 Validation Experiments

A suite of validation experiments is first designed. The nature of a validation experiment is fundamentally different from the nature of a conventional experiment. The basic premise is that a validation experiment is a somewhat simpler procedure that isolates the phenomenon of interest. A suite of experiments provides increasing levels of understanding.

Static material testing comes first, which allows the identification of bulk mechanical properties such as the modulus of elasticity. Because such tests are static in nature, they must be augmented with Hopkinson bar tests. Hopkinson bar tests however do not provide sufficient resolution in the regime of interest, that is, at high strain-rates and varying temperatures.

To obtain more insight into the behavior of the plasticity throughout the validation domain, Taylor anvil tests are performed next. The validation experiments explore different regions of the validation domain while providing successive material models that are hopefully consistent with each other. The discussion in this work focuses on the definition of an error metric between inferences made from the

Hopkinson bar tests and inferences made from the Taylor anvil impact tests (see section 4.6).

4.3 Design vs. Calibration Parameters

The Zerilli-Amstrong model estimates the stress resulting from a plastic deformation as:

$$\sigma = C_o + C_1 e^{-C_3 T + C_4 T \log(\frac{d\varepsilon_P}{dt})} + C_5 \varepsilon_P^N \qquad (9)$$

where the symbol *T* represents temperature and ε_P denotes plastic strain. The six parameters C_0, C_1, C_3, C_4, C_5 and *N* are material-dependent constants that can be calibrated to improve the predictive accuracy of the model. Because of the large spread of strain-rates for which a validated model is sought (from the quasi-static rate of 10^{-3}/second to $4\text{x}10^{+3}$/second), another symbol S_R is introduced that defines the logarithm of the plastic strain rate:

$$S_R = \log_{10}\left(\frac{d\varepsilon_P}{dt}\right) \qquad (10)$$

It is important not to confuse the six calibration variables $(C_0;C_1;C_3;C_4;C_5;N)$ with the two input parameters $(T;S_R)$ that define the operational space or validation domain. The main difference between the two is that calibration variables are introduced by our particular choice of plasticity model. Should another physical model be adopted, the calibration variables would likely change. The dimensionality of the operational space, however, never changes and the plasticity models—whatever they are—must still be validated at various combinations of $(T;S_R)$.

4.4 Forward Propagation of Uncertainty

Two of the key technologies for V&V are the propagation and analysis of uncertainty. This is because model validation is essentially an exercise in the assessment and quantification of uncertainty, whether it originates from the model (lack-of-knowledge, modeling assumptions, and parametric variability), computations (mesh and convergence-related numerical errors), physical experiments (variability of the environment and measurement error), or judgments (vagueness and linguistic ambiguity).

In sections 4.4 and 4.5, some of the tools employed to propagate uncertainty are briefly illustrated. They include Monte Carlo sampling for propagating uncertainty through forward calculations and Bayesian calibration for backward inference. Other tools, that include the design of experiments and analysis-of-variance, are illustrated in section 5 [23-25].

Figure 7 shows several deformed profiles simulated with the finite element package HKS/Abaqus^{TM} [26]. In this numerical simulation, an axi-symmetric mesh is impacted against a perfectly rigid surface, which produces large deformations (over 260%) and significant plastic strain at the crushed end of the cylinder. The six profiles in Figure 7 are obtained by varying the material coefficients $(C_0;C_1;C_3;C_4;C_5;N)$.

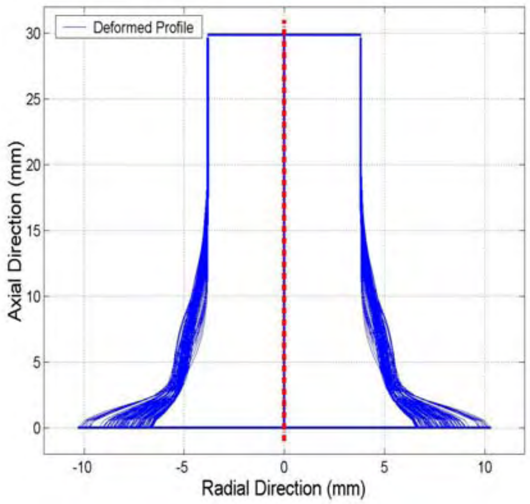


Figure 7. Deformation profiles randomly selected from a 1,000-run Monte Carlo analysis.

Each calibration variable is assumed to vary according to a Gaussian Probability Density Function (PDF). Because the calibration variables are assumed to be independent and uncorrelated, sampling the six individual PDF laws is straightforward. Random samples of coefficient values are drawn from the six distributions independently. A combination of variables $(C_0;C_1;C_3;C_4;C_5;N)$ defines a specific material model, and the impact simulation is repeated for each model. This procedure defines a Monte Carlo simulation, and a total of 1,000 finite element calculations are performed.[6]

In Figures 8 and 9, two features of the response are defined to characterize the deformed profiles, the ratios of final-to-initial lengths (L/L_o) and footprints (R/R_o). Each point

[6] The number of simulations is selected somewhat arbitrarily based on the time necessary to perform a single analysis and the available computing resource.

in Figure 8 corresponds to the result of an impact simulation for a particular material model. It can be observed that there is a significant correlation between the two output features, as one would expect because the shorter the cylinder, the larger its footprint. The histograms shown in Figure 9 are obtained by projecting the distribution of output features on the horizontal and vertical axes. Each axis is then discretized in twenty bins and the histograms simply show how many features are counted within each bin. The histograms approximate the output PDF. It can be observed from their asymmetries and long tails that the probability laws of response features *(L/L_o)* and *(R/R_o)* are not normal. It is well known that a Gaussian PDF propagated through a non-linear system such as this finite element simulation does not stay Gaussian.

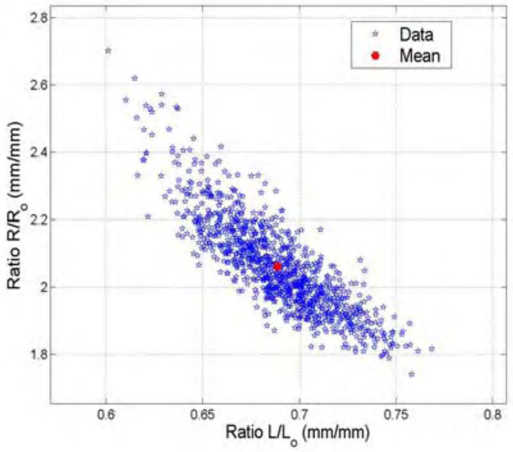


Figure 8. Distribution of features (L/L_o) and (R/R_o) obtained from the Monte Carlo simulation.

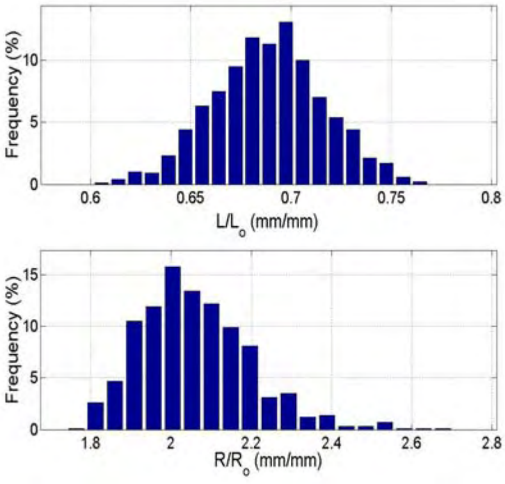


Figure 9. Histograms of (L/L_o) and (R/R_o).

Monte Carlo simulations are popular because of their simplicity and well-established convergence properties. Our simple illustration demonstrates the propagation of uncertainty from inputs to outputs and the estimation of the response's probability structure, from which statistics can be calculated. To guarantee convergence, however, large numbers of samples may be required in which case other sampling strategies—for example, stratified sampling such as the Latin Hypercube Sampling [27] or orthogonal arrays [28]—offer attractive alternatives. Screening experiments and analysis of variance can also be performed to understand which combinations of inputs are responsible for explaining a spread of responses such as the one pictured in Figure 8 [23, 25, 29].

4.5 Inverse Propagation of Uncertainty

The calibration of model parameters is a technique often employed to improve the fidelity-to-data. Calibration is generally formulated as a deterministic inverse problem. A cost function is defined as the "distance" in some sense between measurements and predictions. Model parameters are then optimized to minimize the cost function. In the context of statistics where it is recognized that both calibration parameters and response features are random variables, a mechanism must be found to propagate uncertainty from the measurements back to the inputs. This is here referred to as the inverse propagation of uncertainty.

Although many formulations are possible, the concept of Bayesian inference is illustrated here for the Taylor impact application [30]. Like in the deterministic case, a procedure for inverse propagation of uncertainty starts with the definition of a cost function. The main difference is to take into account the fact that the input parameters *p* and output features *y* are random variables, which generally implies that the cost function becomes a statistical test. In the case of Bayesian inference, the cost function defines the posterior probability that the model parameters *$p=(C_0;C_1;C_3;C_4;C_5;N)$* are correct given evidence provided by the physical measurements *y^{Test}*. The posterior PDF is the conditional probability law of the calibration variables *p*:

$$e^2 = -2\log(\mathrm{Prob}(p \mid y^{\mathrm{Test}})) \qquad (11)$$

The Bayes Theorem states that the posterior probability *(e^2)* is the product of the likelihood function—likelihood to predict the measurements based on a given model—multiplied by the prior probability of *p*. Under the assumption of Gaussian probability laws, the likelihood function becomes the mean square error between measurements and predictions. One advantage is that the cost function obtained is a closed-form expression:

$$e^2 = \sum_{k=1\ldots N} \left(y_k^{Test} - y_k(p)\right)^T \left(\Sigma_{y_k}^{Test}\right)^{-1} \left(y_k^{Test} - y_k(p)\right) + \left(p - p_o\right)^T \left(\Sigma_p\right)^{-1} \left(p - p_o\right) \qquad (12)$$

where the inverted matrices are formed from variance and covariance values, and the symbol p_o denotes the nominal material coefficients. The quantify $y=\{(L/L_o);\ (R/R_o)\}$ collects the output features. The summation aggregates information obtained from potentially different experiments (see section 6). When Gaussian probability distributions are assumed, the cost function (12) becomes the well-known chi-square statistical test that attempts to reject the null hypothesis that measurements y^{Test} and model predictions *y* are sampled from the same parent population.

The general procedure for calculating the chi-square statistics goes as follows. First, the simulation is analyzed for a given experimental configuration defined by the temperature and strain-rate parameters $(T;S_R)$, and given material coefficients (*p*). The features (L/L_o) and (R/R_o) are calculated from the final deformed profile. These two predictions are compared to the measurements. Figure 10 shows a typical test-analysis comparison. The procedure is repeated for each combination of temperature and strain-rate $(T;S_R)$ to accumulate the chi-square metric.

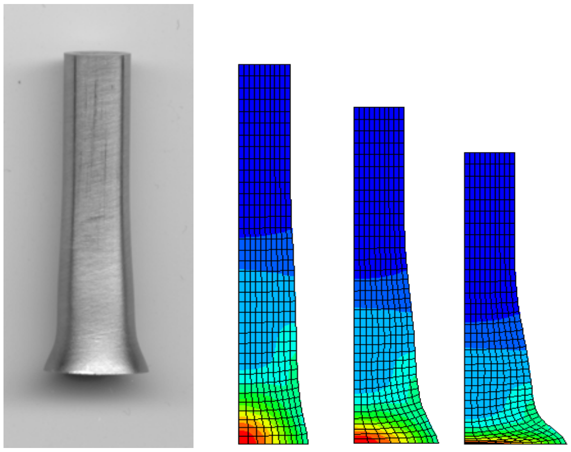


Figure 10. Test-analysis comparison (Left: measured profile; right: simulated profiles at 17 ms, 33 ms, and 50 ms after impact.)

Once a numerical procedure has been defined to compute the cost function, the calibration variables $p=(C_0;C_1;C_3;C_4;C_5;N)$ are optimized to search for the lowest possible chi-square value. Because the prior and posterior PDF laws have been assumed to be Gaussian, a deterministic optimization solver can be used to optimize the posterior mean and covariance. In the case where no evidence is available to suggest a particular distribution, the main difficulty becomes the estimation of a posterior PDF whose functional form is unknown. This can be resolved with a Markov Chain Monte Carlo optimizer that exhibits the attractive property of being able to sample an unknown probability law [25, 31].

Table 2: Values of the calibration variables.

	Prior Mean[a]	**Post. Mean**[b]	**Prior STD**[c]	**Post. STD**[d]
C_0	175.0	102.5	20.0%	32.1%
C_1	950.0	954.3	20.0%	9.6%
C_3	3.0x10^{-3}	4.1x10^{-3}	20.0%	14.6%
C_4	8.5x10^{-5}	11.7x10^{-5}	20.0%	24.8%
C_5	675.0	996.2	20.0%	2.2%
N	0.275	0.247	20.0%	8.5%

Legend: [a]Mean of the prior distribution; [b]Mean of the posterior distribution; [c]Standard deviation of the prior distribution, divided by the prior mean in column 2; [d]Standard deviation of the posterior distribution, divided by the posterior mean in column 3.

The calibration results presented in Table 2 are obtained by minimizing the cost function (12) with a gradient-based optimization solver. The statistical moments of the calibration variables prior and posterior Gaussian distributions are shown. Bayesian inference also updates the entire posterior covariance matrix, from which one can study the correlation between variables. It can be observed from Table 2 that calibration tends to reduce the standard deviation values, except for variables C_3 and C_4. This indicates that information is learned from the test-analysis correlation exercise.

In summary, the example illustrates how a set of calibration variables $p=(C_0;C_1;C_3;C_4;C_5;N)$ can be optimized to improve the fidelity-to-data of a numerical simulation. Prediction accuracy is improved using physical experiments performed at various settings of the design parameters, in our case the temperature and strain-rate $(T;S_R)$. The procedure relates measurement variability to input uncertainty, hence propagating the uncertainty backwards.

5. UNCERTAINTY QUANTIFICATION

In section 4, the propagation of uncertainty from input variables to output features, or vice-versa, is discussed—forward propagation and inverse propagation, respectively. Another type of analysis, namely the ***screening experiment***, is addressed here. Effect screening refers to the identification of interactions between inputs to explain the variability of outputs.

Effect screening addresses questions such as: "*Which inputs or combination(s) of inputs explain the variability of outputs?*" Screening is typically performed to truncate the list of input variables by determining which ones most influence the output features over the entire design domain. Identifying the effects most critical to an input-output relationship is also the basis for replacing physics-based models by fast-running surrogates (see section 5.2).

5.1 Effect Screening

Effect screening is typically achieved using the concept of analysis-of-variance [29], or other techniques such as Bayesian screening [25] (not discussed here). Our experience with these methods is that a successful screening of effects should always provide consistent results when techniques of different nature are employed [23].

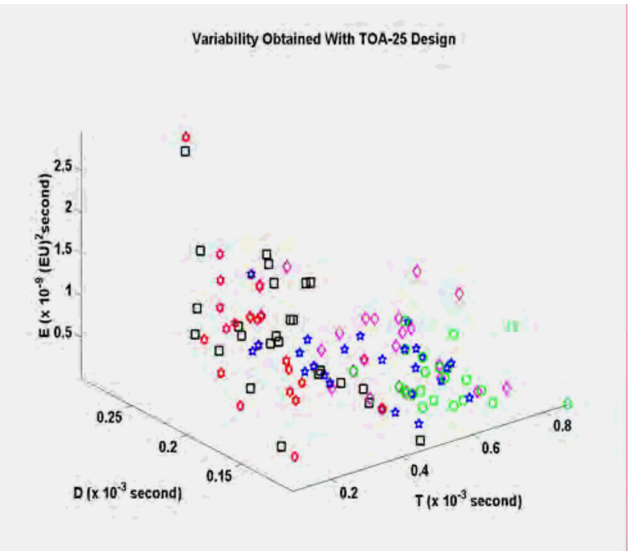


Figure 11. Variability of predictions obtained by varying the material coefficients.

Figure 11 shows features of the longitudinal stress wave that propagates through the material during Taylor impact simulations. The three features are temporal moments E, T, and D that characterize the energy and time-of-arrival of the waveform [32]. Figure 11 is shown to explain what screening can be useful for. Each datum results from a numerical simulation performed with a specific material model (9). The "spread" of predictions *(E;T;D)* comes from varying the five material coefficients C_1, C_3, C_4, C_5, and N. Figure 11, however, does not explain which input (C_1, C_3, C_4, C_5, or N) or combination of inputs (such as C_1C_3, C_1N, or N^2) explains the variability of E, T, and D. Screening answers such question, basically, by performing multiple regression analyses and estimating correlations between the input effects and output features.

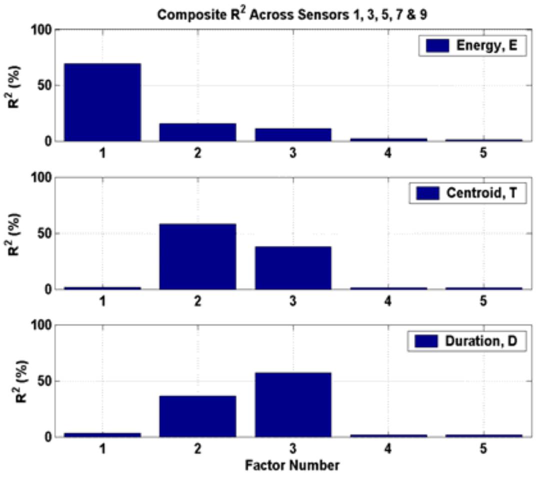


Figure 12. Screening of important variables through an analysis-of-variance.

Figure 12 pictures the results of an analysis-of-variance for main effects only. The screening of main effects (also known as linear screening) attempts to identify the inputs that control the output variability, without accounting for higher-order effects. The R^2 statistics, that estimate a coefficient of correlation, are shown for the three features E, T, and D. A large R^2 relative to the other values indicates that the corresponding input factor (either C_1 in position 1, C_3 in position 2, etc.) produces a significant variability of the output feature. The analysis demonstrates that the variability of features *(E;T;D)* is controlled for the most part by the material coefficients C_1, C_3, and C_4. Because the other factors do not produce significant output variability, they can be kept constant and equal to their nominal values. These results illustrate the potential of screening experiments to understand where an observed variability comes from, and how it can most effectively be reduced.

5.2 Surrogate Modeling

Many analyses (such as the propagation of uncertainty, parameter calibration, and reliability) become computational prohibitive when the number of inputs and outputs increases. Instead of arguing for approximate and less expensive solutions, the approach can be taken to replace physics-based models by surrogates.

Surrogate models (also known as meta-models, emulators, or response surfaces) capture the relationship between inputs and outputs without providing a detailed description of the physics, geometry, loading, etc. The advantage of surrogate models is that they can be analyzed at a fraction of the cost it would take to perform the physics-based simulations. Examples include polynomials, exponential decays, neural networks, principal component decomposition, and statistical inference models.[7]

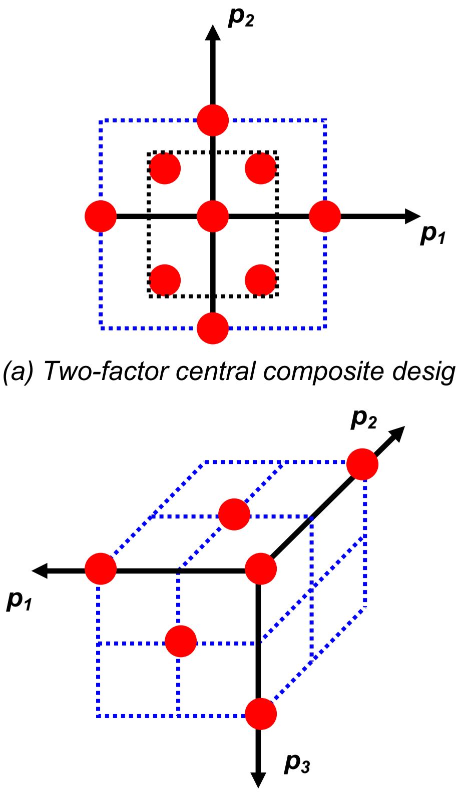
Meta-models must be trained, which refers to the identification of their unknown functional forms and coefficients. Their quality must be evaluated independently from the training step. Because analyzing finite element models at every combination of input variables is generally a combinatorial impossibility, training can be based on a subset of carefully selected runs that, statistically speaking, provide the same amount of information (see section 5.3).

5.3 Design of Computer Experiments

Design of Experiments (DOE) techniques have been developed for exploring large design spaces when performing complex, physical experiments. They can be brought to bear to select judicious subsets of finite element runs. Such simulations provide the input-output data used, for example, to fit surrogate models.

It is important to realize that meta-modeling and effect screening can both be performed with the same DOE because identifying which effects and interactions capture a particular input-output relationship essentially delivers the functional form of the surrogate model.

Examples of popular designs include the Morris method, fractional factorials, the Central Composite Design (CCD), and orthogonal arrays. Figure 13 illustrates a two-factor CCD and a three-factor orthogonal array. In both cases, finite element analyses are performed at the combinations *(p_1;p_2)* or *(p_1;p_2;p_3)* shown. It can be observed that the CCD and orthogonal array require nine and six finite element runs, respectively. Using the CCD, a fully quadratic polynomial can be identified. The orthogonal array shown in Figure 13 can only identify a few interactions (such as p_1p_2 or p_1p_3) but quadratic effects (such as p_1^2 or p_2^2) cannot be screened.



(a) Two-factor central composite design.

(b) Three-factor orthogonal array design.
Figure 13. Designs for computer experiments.

Characteristics of a good design are that it relies on as few runs as possible; provides screening capabilities; leads to high-quality surrogate models; and minimizes aliasing, which refers to the compounding of effects that cannot be captured by the design [33-34].

5.4 General Information Theory

The discussion of uncertainty quantification and propagation has so far been placed in the context of probability. It is our assessment that probability theory provides an appropriate model of uncertainty for most engineering applications.

[7] An example of meta-model is presented in section 6 for assessing the predictive accuracy of the Taylor impact simulation. Techniques for the design of experiments, screening, and meta-modeling are overviewed in References [33-34].

Analysts should nevertheless be cautious not feeling over-confident and assuming more than is really known. In cases of severe lack-of-knowledge, more appropriate theories may be available to represent uncertainty. Examples include interval arithmetic; fuzzy logic [35]; the theory of information-gap [6-7]; the Dempster-Shafer theory of evidence [36]; and possibility theory [37]. These are generally referred to as the Generalized Information Theory (GIT).

Information integration and aggregation is a major challenge that the current research is attempting to address. Although progress has mostly been made within the GIT and statistical sciences communities, work is in progress in the Structural Dynamics community as well [8-9].

6. PREDICTIVE ACCURACY

In section 5, the problem of calibrating the model's parameters under uncertainty has been illustrated. Calibration, however, is only a tool in support of predictive accuracy assessment. A calibrated model is likely to provide small prediction errors in the neighborhood of the points used for calibrating its parameters but the question of adequacy in other regions of the operational space remains. In the remainder, an assessment of predictive accuracy for the plasticity model is illustrated. The illustration is purposely simplified for the sake of clarity.

6.1 Assessment of Prediction Accuracy

The concept of operational space (or validation domain) introduced in section 4.1 is essential to the discussion. The operational domain, that is, the set of conditions for which a validated model of plasticity is sought, is defined by the combination of temperature *T* and strain-rate S_R, as shown in Figure 14. The red dots symbolize the settings *(T;S_R)* at which physical experiments have been performed. These are the same seven experiments used previously to calibrate the variables *(C_0;C_1;C_3;C_4;C_5;N)*, see section 4.5. The question we would like to answer is the following one: "*Can the predictive accuracy of the plasticity model be estimated throughout the operational domain?*"

Figure 15 pictures the results of the inverse propagation of uncertainty from section 4.5. The solid lines represent the strain-stress curves predicted by the calibrated Zerilli-Amstrong model for the seven configurations *(T;S_R)* shown in Figure 14. Predictions are compared to experimental measurements obtained with the Hopkinson bar tests. The vertical bars represent the experimental uncertainty. Simply speaking, Figure 15 compares the consistency obtained when material models are inferred from two different experimental procedures. Predictive accuracy of the Zerilli-Amstrong plasticity model is assessed based on this consistency.

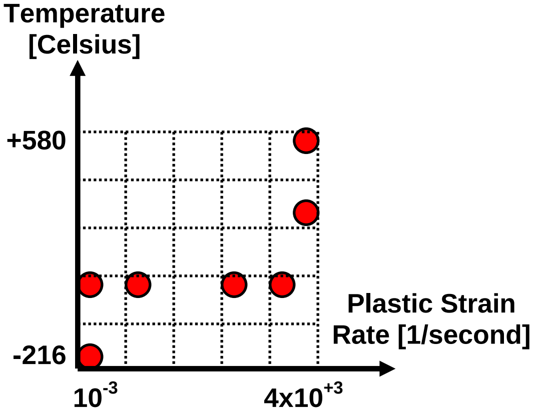


Figure 14. Location of the seven Taylor tests performed in the operational space.

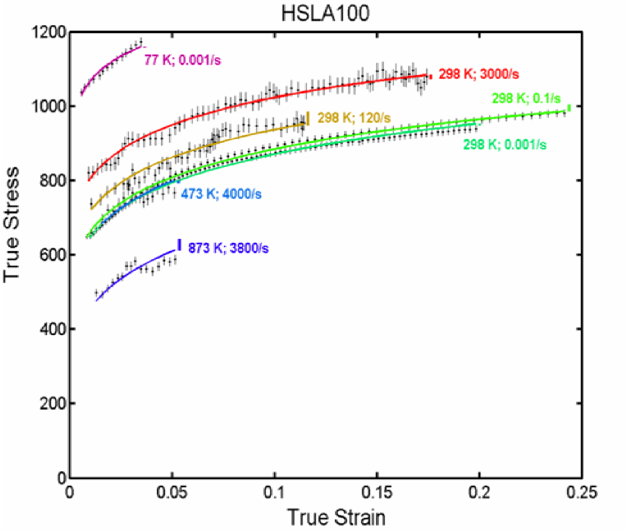


Figure 15. Comparison between Zerilli-Amstrong models and Hopkinson bar test data sets.

To define a quantitative metric of correlation, the Mahalanobis distance is calculated between measurements and predictions:

$$e^2 = \left(y^{Test} - y(p)\right)^T \left(\Sigma_y^{Test}\right)^{-1} \left(y^{Test} - y(p)\right) \tag{13}$$

where y^{Test} represents the mean of the measured stress-strain curves and *y(p)* is the prediction of the calibrated plasticity model. The covariance matrix is equal to a diagonal matrix—therefore neglecting the correlation between the data—initialized with the measurement variance. The Mahalanobis statistic is adopted here because it assesses the fidelity-to-data relative to the experimental variability. However, the choice of a metric for test-analysis comparison is generally application-specific. Figure 16 shows the Zerilli-Amstrong prediction errors computed at the seven settings in the two-dimensional space *($T;S_R$)* where Taylor anvil impacts have been performed. The question remains of knowing how well the model performs at locations in the operational space where no physical experiment is available.

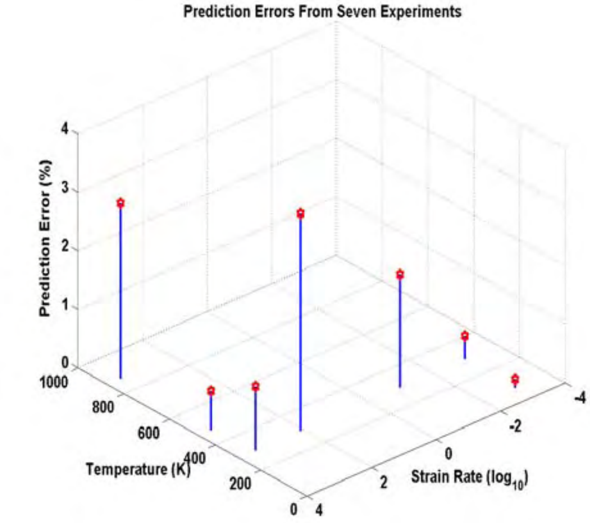


Figure 16. Mahalanobis error metrics for predictive accuracy assessment.

This question is addressed by developing a meta-model of predictive accuracy. A statistical model is developed based on the seven errors of Figure 16. The model is referred to as a meta-model because its development is not based on physical principles. Its only purpose is to capture an input-output relationship—between the input parameters *($T;S_R$)* and the prediction error e^2—and extrapolate beyond the available data with reasonable accuracy. A polynomial is selected for simplicity. Other choices might include neural networks or Krigging models. An arbitrary choice such as this one expresses lack-of-knowledge. Even though it is not shown, this uncertainty should be accounted for in the assessment of prediction accuracy. The estimation of predictive accuracy is illustrated graphically over the entire domain *($T;S_R$)* in Figure 17.

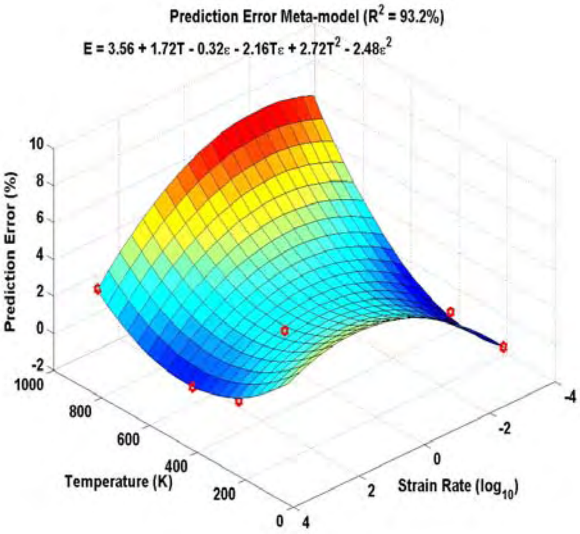


Figure 17. Meta-model of prediction accuracy.

The significance of Figure 17 is that it provides an estimation of the adequacy of the Zerilli-Amstrong model everywhere within the operational domain, without having to perform any additional physical experiment or simulation. It is emphasized that validation does not mean that the model should be "perfect" everywhere, which is a significant shift of paradigm compared to the model calibration approach. The model is validated because its prediction accuracy has been assessed throughout the design space, and especially away from testing conditions.

6.2 Discussion

Two key issues have not been discussed. They are the breakdown of total uncertainty and the quantification of modeling uncertainty. Total uncertainty refers to the aggregation of all potential sources of uncertainty, originating from measurements, models, or expert knowledge. In this work, examples have been given of how to account for independent sources such as measurement uncertainty, parametric variability, or solution error, but bringing them together has not been addressed. Information integration and total uncertainty assessment are currently being studied. The importance of these concepts is critical for validation and margin assessment.

Modeling lack-of-knowledge refers to the uncertainty that results from arbitrary choices such as modeling assumptions and functional forms. Any rigorous assessment of predictive accuracy should account for such uncertainty.

One roadblock is that many of these sources of uncertainty cannot be quantified probabilistically.

7. CONCLUSION

This chapter offers a brief overview of the technology developed at Los Alamos National Laboratory in support of engineering verification and validation programs. The material presented is based to a large extent on a tutorial taught at the Los Alamos Dynamics Summer School [1]. The tools developed are applied to the validation of a non-linear plasticity model. The technology includes code and solution verification, statistical sampling, design of experiments, screening, metrics for test-analysis correlation, surrogate modeling, and calibration.

The objective is to provide an estimation of prediction accuracy when the plasticity model is implemented to calculate strain-stress curves at any combination of input settings. Obtaining this information over an entire operational space is essential to answer questions such as: "*How appropriate is the model overall?*" "*Which one of several competing models is more appropriate for a particular application?*" "*Which physical tests would be useful to improve the predictive accuracy of the model?*"

8. REFERENCES

[1] The Los Alamos Dynamics Summer School, ext.lanl.gov/projects/dss/home.htm.

[2] Doebling, S.W., "Structural Dynamics Model Validation: Pushing the Envelope," *International Conference on Structural Dynamics Modeling: Test, Analysis, Correlation and Validation*, Madeira Island, Portugal, Jun. 3-5, 2002.

[3] Hemez, F.M., Ben-Haim, Y., "The Good, the Bad, and the Ugly of Predictive Science," *4th International Conference on Sensitivity Analysis of Model Output*, Santa Fe, New Mexico, Mar. 8-11, 2004.

[4] Mottershead, J.E., Friswell, M.I., "Model Updating in Structural Dynamics: A Survey," *Journal of Sound and Vibration*, Vol. 162, No. 2, 1993, pp. 347-375.

[5] Hemez, F.M., Doebling, S.W., "Inversion of Structural Dynamics Simulations: State-of-the-art and Orientations of the Research," *25th International Conference on Noise and Vibration Engineering*, Leuven, Belgium, Sep. 13-15, 2000, pp. 403-413.

[6] Ben-Haim, Y., "Robust Rationality and Decisions Under Severe Uncertainty," *Journal of the Franklin Institute*, Vol. 337, 2000, pp. 171-199.

[7] Ben-Haim, Y., **Information-Gap Decision Theory: Decisions Under Severe Uncertainty**, Academic Press, 2001.

[8] Booker, J., Ross, T., Hemez, F.M., Anderson, M.C., Joslyn, C., "Quantifying Total Uncertainty in a Validation Assessment Using Different Mathematical Theories," *14th Biennial Nuclear Explosives Design Physics Conference*, Los Alamos, New Mexico, Oct. 20-24, 2003.

[9] Ross, T., Booker, J., Hemez, F.M., "Quantifying Total Uncertainty and Performance Margin in Assessing the Reliability of Manufactured Systems," *5th Biennial Tri-lab. Engineering Conference*, Santa Fe, New Mexico, Oct. 21-23, 2003.

[10] AIAA, **Guide for the Verification and Validation of Computational Fluid Dynamics Simulations**, *American Institute of Aeronautics and Astronautics*, AIAA-G-077-1998, Reston, Virginia, 1998.

[11] Wallace, D.R., Ippolito, L.M., Cuthill, B.B., "Reference Information for the Software Verification and Validation Process," *Sandia National Laboratories*, Report 500-234, Albuquerque, New Mexico, 1996.

[12] Roache, P.J., "Code Verification by the Method of Manufactured Solutions," *Journal of Fluids Engineering*, Vol. 114, No. 1, 2002, pp. 4-10.

[13] Roache, P.J., **Verification and Validation in Computational Science and Engineering**, Hermosa Publishers, Albuquerque, New Mexico, 1998.

[14] Shih, T.M., "A Procedure to Debug Computer Programs," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 6, 1985, pp. 1027-1037.

[15] Ferziger, J.H., Peric, M., **Computational Methods for Fluid Dynamics**, Springer-Verlag, New York City, New York, 1996.

[16] Oden, J.T., "Error Estimation and Control in Computational Fluid Dynamics," in **The**

Mathematics of Finite Elements and Applications, Whiteman, J.R., Editor, John Wiley and Sons, New York, 1993, pp. 1-23.

[17] Ainsworth, M., Oden, J.T., **A Posteriori Error Estimation in Finite Element Analysis**, John Wiley, New York, 2000.

[18] Roache, P.J., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, 1994, pp. 405-413.

[19] Cadafalch, J., Perez-Segarra, C.C., Consul, R., Oliva, A., "Verification of Finite Volume Computations on Steady State Fluid Flow and Heat Transfer," *Journal of Fluids Engineering*, Vol. 124, 2002, pp. 11-21.

[20] Chen, C.-F., Lotz, R.D., Thompson, B.E., "Assessment of Numerical Uncertainty Around Shocks and Corners on Blunt Trailing-edge Supercritical Airfoils," *Computers and Fluids*, Vol. 31, No. 1, 2002, pp. 25-40.

[21] Adessio, F.L., Johnson, J.N., Maudlin, P.J., "The Effect of Void Growth on Taylor Cylinder Impact Experiments," *Journal of Applied Physics*, Vol. 73, No. 11, Jun. 1993, pp. 7288-7297.

[22] Zerilli, F.J., Amstrong, R.W., "Dislocation Mechanics-based Constitutive Relations for Material Dynamics Calculations," *Journal of Applied Physics*, Vol. 61, No. 5, Mar. 1987, pp. 1816-1825.

[23] Cundy, A.L., Schultze, J.F., Hemez, F.M., Doebling, S.W., Bingham, D., "Variable Screening in Metamodel Design for a Large Structural Dynamics Simulation," *20th International Modal Analysis Conference*, Los Angeles, California, Feb. 4-7, 2002, pp. 900-903.

[24] Hemez, F.M., Wilson, A.C., Doebling, S.W., "Design of Computer Experiments for Improving an Impact Test Simulation," *19th International Modal Analysis Conference*, Kissimmee, Florida, Feb. 5-8, 2001, pp. 977-985.

[25] Kerschen, G., Golinval, J.-C., Hemez, F.M., "Bayesian Model Screening for the Identification of Non-linear Mechanical Structures," *Journal of Vibration and Acoustics*, Vol. 125, Jul. 2003, pp. 389-397.

[26] **HKS/Abaqus™ Explicit User's Manual**, Version 5.8, Hibbitt, Karlsson & Sorensen, Pawtucket, Rhode Island, 1998.

[27] McKay, M.D., Beckman, R.J., Conover, W.J., "A Comparison of Three Methods For Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239-245.

[28] Hedayat, A.S., Sloane, N.J.A., Stufken, J., **Orthogonal Arrays: Theory and Applications**, Springer-Verlag, 1999.

[29] Saltelli, A., Chan, K., Scott, M., **Sensitivity Analysis**, John Wiley and Sons, 2000.

[30] Hanson, K.M., "A Framework for Assessing Uncertainties in Simulation Predictions," *Physica D*, Vol. 133, 1999, pp.179-188.

[31] Carlin, B.P., Chib, S., "Bayesian Model Choice via Markov Chain Monte Carlo," *Journal of the Royal Statistical Society Series B*, Vol. 77, 1995, pp. 473-484.

[32] Hemez, F.M., Doebling, S.W., "From Shock Response Spectrum to Temporal Moments and Vice-versa," *21st International Modal Analysis Conference*, Kissimmee, Florida, Feb. 3-6, 2003.

[33] Myers, R.H., Montgomery, D.C., **Response Surface Methodology: Process and Product Optimization Using Designed Experiments**, Wiley Inter-science, 1995.

[34] Wu, C.F.J., Hamada, M., **Experiments: Planning, Analysis, and Parameter Design Optimization**, John Wiley and Sons, 2000.

[35] Klir, G.J., Yuan, B., **Fuzzy Sets and Fuzzy Logic: Theory and Applications**, Prentice Hall, 1995.

[36] Shafer, G., **A Mathematical Theory of Evidence**, Princeton University Press, 1976.

[37] Ross, T.J., "Possibility Intervals in the Characterization of Dynamic Model Uncertainty," *19th International Conference of the North American Fuzzy Information Processing Society*, Atlanta, Georgia, Jul. 13-15, 2000, pp. 353-357.