SANDIA REPORT

SAND2002-0341
Unlimited Release
Printed March 2002

General Concepts for Experimental Validation of ASCI Code Applications

Timothy G. Trucano, Martin Pilch, and William L. Oberkampf

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

Sandia National Laboratories

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

Printed in the United States of America. This report has been reproduced directly from the best available copy.

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)576-5782
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

SAND2002-0341
Unlimited Release
Printed March 2002

General Concepts for Experimental Validation of ASCI Code Applications

Timothy G. Trucano
Optimization and Uncertainty Estimation

Martin Pilch and William L. Oberkampf
Validation and Uncertainty Quantification

Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0819

Abstract

This report presents general concepts in a broadly applicable methodology for validation of Accelerated Strategic Computing Initiative (ASCI) codes for Defense Programs applications at Sandia National Laboratories. The concepts are defined and analyzed within the context of their relative roles in an experimental validation process. Examples of applying the proposed methodology to three existing experimental validation activities are provided in appendices, using an appraisal technique recommended in this report.

Acknowledgements

The authors have interacted with many people about the contents of this report. We specifically acknowledge our colleagues Richard Klein, Roger Logan, and Cynthia Nitta at Lawrence Livermore National Laboratory; and Robert Benjamin, Joseph Kindel, James Kamm, Kenneth Koch, Douglass Post, William Rider, and David Tubbs at Los Alamos National Laboratory. The authors would like to specifically thank our fellow Sandians Tom Carne, T. Y. Chu, Kevin Dowding, Steve Lott, William Moffatt, Jaime Moya, Jim Nakos, Tom Paez, Vicente Romero, Brian Rutherford, and Angel Urbina for reading and commenting on various drafts of this manuscript before publication. Ashley Emery (University of Washington) also provided detailed comments on a draft of this report. Finally, we thank Rhonda Reinert for providing detailed editing of the final draft of this report.

Contents

Figures

Executive Summary

Accelerated Strategic Computing Initiative (ASCI) codes will be applied to high-consequence nuclear stockpile problems. In some cases, a premium will be placed on the credibility of the *predictive* application of these codes. The credibility of a code for use on a high-consequence problem is strongly dependent upon the success of a set of scientifically defensible and consequential verification and validation tasks. For computational science and engineering codes of the type being developed by ASCI, validation requires experimental activities that can serve as substantive benchmarks for assessing the fidelity of implemented physics and engineering models. We believe that a systematic, documented analysis of the critical elements of experimental validation projects that is broadly relevant to the Sandia ASCI program does not currently exist.

It is the purpose of this report to define and analyze a process methodology that can be used in planning, executing, and assessing experimental validation projects. Our perspective in writing this report has been to present a specific view of what useful experimental outcomes should be and how they can be achieved in experimental validation projects. The process methodology consists of eight key elements that embody important concepts distinguishing experimental validation. This process is schematically presented in Figure 1.2 as a serial movement through each of the eight key elements. In fact, the text also discusses the complex connections between those process elements that have not been suggested in Figure 1.2 and which act to further complicate the experimental validation methodology.

The main body of this report provides a critical discussion of the eight key elements in Figure 1.2. A brief description of the elements follows.

Element 1: Defense Programs (DP) Application

Validation activities must assess confidence in the use of the code for a specified DP application. The DP application at which a particular validation activity is directed is a critical planning element and must be defined before the performance of any specific validation work. Specific requirements associated with the selected DP application must be identified during planning (Element 2), as these requirements place serious constraints on the goals of the validation activity.

Element 2: Planning

Validation activities must be formally planned. The plan should define the key elements in the experimental activity, including the design of the validation experiments (Element 4), the validation metrics intended to be applied (Element 5), and the assessment criteria to be applied (Element 6). If some aspects of the validation activity involve research, such as definition of particular metrics, this should be spelled out in the plan. It is important to emphasize that this planning exists within a broader planning environment governing DP activities, experimental campaigns, and ASCI code development. An important component that strongly influences the development of the plan of a particular experimental validation effort is the Phenomena Identification and Ranking Table (PIRT), a deployment tool that crystallizes specific prioritized validation activities from the application requirements of the code. The PIRT is discussed in greater detail in Pilch et al. (2000a.).

Element 3A: Code Verification

A nominal level of code verification should be established as a requirement to conduct a validation activity. Typically, this element centers on gathering evidence that the code can produce calculations that can be fruitfully compared with the gathered experimental data. Although this evidence will likely exist independently within the code development project (for example, evidence of configuration management and testing performance), code verification may also require independent work on the part of the validation team. Bluntly speaking, we do not consider codes that do not run as being proper targets for the scarce resources associated with dedicated experimental-validation efforts.

Element 3B: Calculation Verification

Calculation verification seeks assessment of the accuracy of calculations performed during the course of the validation activity, for example, through convergence studies and *a posteriori* error estimation. Such calculations encompass additional verification studies that could be needed to accomplish satisfactory code verification (Element 3A), as well as predictive calculations applied in the design of validation experiments and postdictive calculations applied in postexperiment analysis (Element 4). The role of calculation verification should be defined in the planning element (Element 2).

Element 4: Experimental Design, Execution, and Analysis

Properly designed and executed validation experiments are an important deliverable of any experimental validation activity. Validation experiments should provide data that are accurate enough to fulfill the validation requirements defined by the underlying DP application. The design of validation experiments should provide quantification of the experimental uncertainty as part of the delivered data. The collected data should be comparable with code calculations in precise enough ways to conduct the metrics (Element 5) and assessment (Element 6) in the validation methodology. This is most possible when the code has participated in the definition and design of the validation experiments as well as in the postexperiment analysis. We thus place a high premium on usage of the code in the design phase of the planned experiments. All anticipated aspects of experimental design, execution, and analysis should be defined in the planning element (Element 2).

Element 5: Metrics

Conclusive comparisons of code calculations with experimental data are the most significant outcome of dedicated experimental validation. These comparisons must be quantitative, should encompass uncertainty in both the experimental data and the code calculations, and must be assessed (Element 6). The definition of metrics that accounts for uncertainty, say, within a probabilistic framework, is currently a research and development topic (Trucano et al. 2001). Anticipated metrics should be defined in the plan (Element 2).

Element 6: Assessment

Assessment is the methodological element that determines the increase or decrease in confidence in the code that results from the validation activity. Sharp assessment is dependent upon sharp metrics; hence, this element is very strongly coupled to the metrics (Element 5). An ideal situation is to define assessment criteria that define "passing" or "failing" or both for the test posed by comparison of the calculations with the gathered experimental data. The assessment criteria should be defined in the plan (Element 2).

Element 7: Prediction and Credibility

The original DP application and its requirements ultimately require code usage that may be predictive. Prediction in this case means that the required use of the code represents an extrapolation from the associated validation knowledge base to the larger domain of applicability of the code. Element 7 ensures that the question of credibility of the code for this intended application is *asked* at the completion of the specific validation activity or activities, and that some attempt is made to *answer* it. The desirable outcome of Element 7 is thus to specifically determine the contribution of the specific validation activity, especially through the results of the metrics and assessment elements (Elements 5 and 6), to improving our understanding of the credibility of the code. Formal thought should be devoted to the implications of the assessment activity (Element 6) on establishing credible confidence in the required predictive application of the code. It is desirable to define and apply a methodology for performing the extrapolation of credibility from experimental validation to the desired predictive application. For example, this could be defined as a technique for directly extrapolating observed metric performance in the assessment element from the validation data to the intended application domain using statistical techniques. Extrapolation of credibility gained in the assessment element (Element 6) to the prediction domain should be addressed in the plan for the experimental validation activity (Element 2).

Element 8: Documentation

Documentation should be sufficient to provide traceable, repeatable, and credible information about the conduct, results, and conclusions of the experimental validation activity needed for future evaluation. The validation activity under consideration is typically only one of several validation activities that are undertaken for a given code application. Therefore, detail and accuracy of the documented information are also required for proper integration of distinct validation activities.

While there are significant interactions between these elements as previously claimed, we have emphasized the serial character of this process in Figure 1.2 and our detailed treatment because this approach favors the practical constraint that specific experimental validation activities must *begin* at some point and *end* at some point. Thus, we have emphasized that experimental validation activities *begin* with the specification of the driving DP application, *end* with the delivery of the necessary information in documented form, and have a carefully defined series of steps that lead from beginning to end. One who is so inclined could probably just as productively read this text by starting at the end state (Element 8) and backtracking to the beginning.

In the more detailed discussion of the major process elements in the body of this report, we have developed a series of further concepts that follow from the general goals of these eight elements. The report focuses on the presentation of these more detailed concepts, a total of 33, which are also summarized in Section 11. In that section, a notional measurement approach has been defined that could that be useful for applying this methodology in planning experimental validation activities, as well as in assessing the outcomes of these activities. The suggested measurement approach is applied to three distinct experimental activities in Appendices A, B, and C. The appendices demonstrate that the concepts presented in this report and summarized in Section 11, along with a rather simple measurement process, develop useful information about the strengths and weaknesses of ongoing as well as historical experimental validation activities.

The ideas presented in this report support a view of experimental validation that contradicts the naively simple characterization that experimental validation activities simply consist of the following elements: conducting experiments, performing calculations, and comparing results. While the elements of the naïve characterization are, indeed, critical in any experimental validation activity, it is attention to their underlying details (how the elements fit together and how they are used) that leads to complexity. The success of experimental validation, in our view, requires painstaking attention to many details, particularly Elements 1 through 8 highlighted in this report, and also requires a process that specifies the manner in which these details are executed and linked.

We now comment on our intended audience. During the writing of this report, we were attempting to define the needs of the modeling community vis-a-vis experimental validation activities. Thus, the primary audience for this report is the joint community of code developers and code users who, at least, might be interested in our view of the needed outcomes of experimental validation. The secondary audience comprises the experimenters themselves, who may wonder about the increased intricacies of interacting with the computational modeling community in experimental validation activities. While this report contains the key definitions of the concepts underlying the proposed experimental validation process, the report also relies upon a body of published work, including other work of the present authors, which is referenced in detail.

Importantly, this report is neither intended nor written to be a requirements document for experimental validation. Attention to all of the concepts described here for a given experimental validation activity could be interpreted as a maximally acceptable approach to validation. From this perspective, requirements should then serve to define a minimally acceptable approach, though we have devoted no attention to this important topic here. There are also topics that influence experimental validation, especially the use of the results of experimental validation, which have only been touched upon in this report and require a more complete discussion. One such topic is *code qualification*, which is closely associated with the problem of when validation (and verification) results are sufficient in number and quality to support a decision to use the code for a specified stockpile problem. Qualification is clearly an important topic that is coupled to the development of requirements for experimental validation tasks.

(Page Left Blank)

Acronyms and Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
ASCI	Accelerated Strategic Computing Initiative
BE+U	Best Estimate Plus Uncertainty
CFD	computational fluid dynamics
DOE	Department of Energy
DP	Defense Programs
DSW	Directed Stockpile Work
M&S	modeling and simulation
PIRT	Phenomena Identification and Ranking Table
Sandia	Sandia National Laboratories
SLEP	Stockpile Life Extension Program
SQE	Software Quality Engineering
STS	Stockpile-to-Target Sequence
TGA	thermal gravimetric analysis
V&V	verification and validation
VALTS	Validation Test Suite
VERTS	Verification Test Suite

(Page Left Blank)

Section 1
Introduction

As defined in the Department of Energy's Accelerated Strategic Computing Initiative (ASCI) program plan, ***validation*** is "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 model applications" (DOE 2000). The important terms to emphasize in this definition are "process" and "applications." Validation, as understood by DOE and implemented at Sandia National Laboratories (Sandia) under the ASCI program, is an evaluation process that is focused by the specific application (or use) of numerical models (also called ***codes*** in this report for simplicity and with an implied restriction to only the software implementing the mathematical model of a physical process). The guidelines for developing verification and validation (V&V) plans for ASCI code projects at Sandia emphasize the critical need to focus on applications in planning V&V activities (Pilch et al. 2000a). As discussed below, these two features, process and applications, are the most important determining factors in defining appropriate validation experiments.

From a technical point of view, ***validation*** relies upon the process of comparing the results of ASCI code calculations with the results of ***physical experiments***, and has the primary ***goal*** of developing and quantifying sufficient ***confidence*** in the codes so that they can be used to predict a specified problem result. The components of a validation process designed to achieve this goal are generically termed ***validation activities***. For DOE stockpile stewardship applications, the intended usage of ASCI codes will typically have high consequence because they are connected to decisions about the nuclear weapons stockpile. At the same time, ASCI code applications supporting stockpile programs will require predictive use of codes in direct proportion to the degree that the intended use extrapolates beyond the existing validation characterization.

Perhaps the most important aspect of our present work in validation is our belief that the goal of validation activities is to assess the ***predictive*** capability of codes for specified applications. This goal is illustrated in Figure 1.1. This figure, which is discussed in greater detail in Easterling (2001a), explicitly emphasizes the need for extrapolation from experimental validation to likely stockpile application. The figure also suggests a hypothetical region of the parameter domain for the model that has been studied through experimental validation. ***Model*** in this report, unless otherwise specifically stated, always means the computational code and the physics models implemented by the code, as well as the input necessary for performing calculations, such as the mesh definition, choice of material parameters, and computational parameter settings. An understanding of the model errors determined from validation experiments is represented by $e_x = y(x) - y^*(x)$, where $y(x)$ is the result of validation experiments at the parameter domain location x, $y^*(x)$ is the model prediction at that location, and e_x is some measure of the difference between the two. The "minus" sign in the formula defining e_x is purely schematic and representative of some choice of "validation metric" (Trucano et al. 2001). We anticipate that we may not be able to perform experiments that closely approximate the specified use in many situations. The intended application thus requires utilization of the code in domains where direct experimental confirmation of prediction accuracy is not possible. Inference must thus be used to estimate e_x in the region labeled "stockpile application" in Figure 1.1. The need to perform this extrapolation emphasizes our need for models to be judged to achieve the right answer for the right reasons in the validation regime. Code ***calibration***, which is performed for the purpose of achieving some degree of agreement with complex integrated tests in the required application domain of the code by explicit tuning, will not assess *predictive* use of the code.

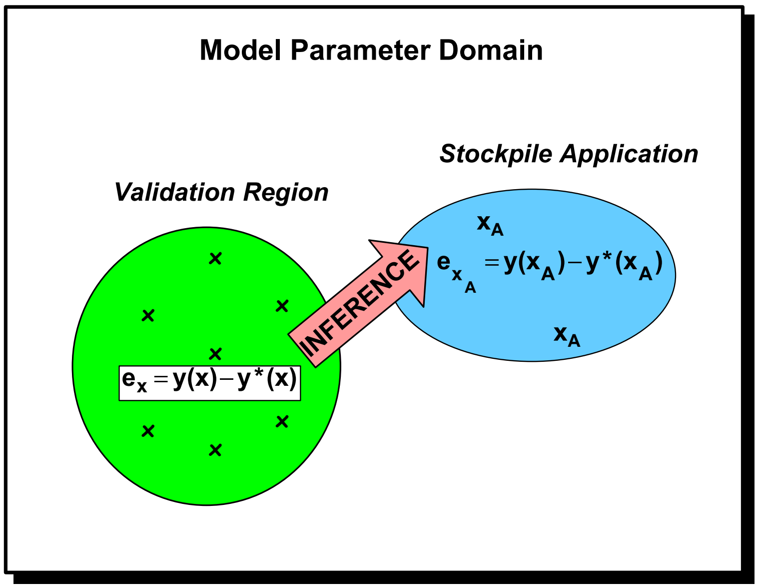

Figure 1.1. Extrapolation beyond the experimental validation regime is typically required for stockpile applications of computational science and engineering (Easterling 2001a).

The goal of the ASCI program is to develop computational science and engineering codes for which we have confidence in this predictive extrapolation. While this may not be the only role of ASCI codes, we contend that predictive ASCI code calculations in this extrapolative sense establish objectives that require the greatest rigor in validation activities. The concepts discussed in this report are intended to form a basis for achieving the necessary rigor. Explicit in the goals for these validation activities is the hope of quantitatively characterizing the margin of error between the computational extrapolation outcome and the actual physical outcome if it could be observed.

We therefore assume that the most important objective of the Sandia ASCI V&V program is to achieve ***confidence*** in the ***predictive application*** of the relevant codes to ***high-consequence*** modeling tasks. The requirement for high-consequence code use implies there are nontrivial consequences of poor code performance. The requirement for predictive code use necessitates extrapolation beyond the understanding gained strictly from experimental validation data. Traditional problems of computational science and engineering in many fields have not had to deal simultaneously with these two requirements: predictive application and high consequence. Therefore, the traditional use of complex scientific numerical models has not demanded as great attention to formalizing the methods and outputs of the validation process as will be needed for high-consequence predictive code use, with some exceptions. One discipline where this statement is not true is in numerical modeling performed as an integrated component of regulatory assessment, such as in nuclear reactor safety assessment (Boyack 1990) and waste repository performance (Cragnolino et al. 2000; Mohanty et al. 2000).

It is difficult to understand technical requirements for performing validation activities that are isolated from the intended application of the code. Validation experiments—the most important activities associated with validation—must be properly centered in the overall process that enables predictive code use. Figure 1.2 depicts our high-level view of the role of experimental validation in evaluating the predictive capability of codes.

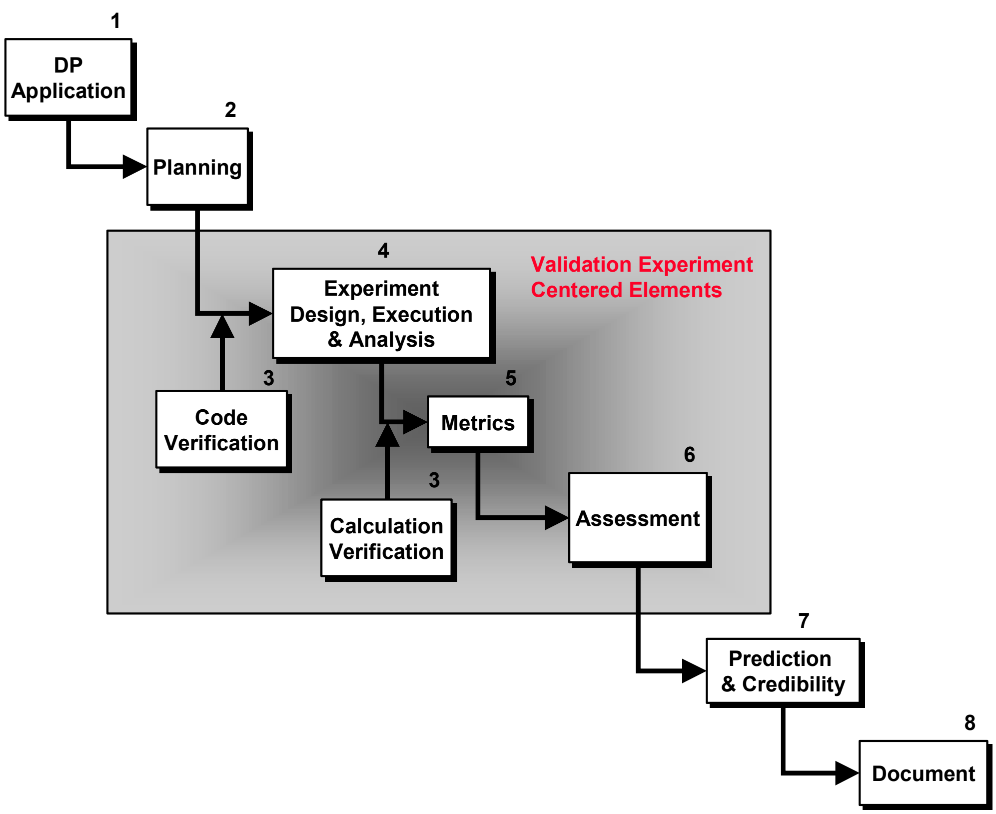

Figure 1.2. The validation process and its predictive capability.

In Figure 1.2 we suggest that the validation-prediction process relies upon the following elements:

1. Identification of the Defense Programs (DP) application driver that focuses the use of the code under discussion.

2. Careful planning of validation activities, especially the use of the Phenomena Identification and Ranking Table (PIRT) to define prioritized V&V activities for the application of the code to the given stockpile application driver.

3. Development and documentation of appropriate verification confidence in the code, as well as the specific solution verification of validation calculations.

4. Design and execution of appropriate experimental activities to mirror the validation elements in the PIRT.

5. Development and application of appropriate metrics for comparing computational results with experimental results to measure confidence in the intended application of the code.

6. Development and application of criteria for assessing the success or failure of the code when compared with the validation experiments using the comparative metrics.

7. Some determination of the conclusions of the validation activities with respect to our confidence in the required predictive application of the code, given that the required level of confidence in the capability of the code is strongly dependent upon the intended predictive application. (The constraints and requirements of how subsequent predictive application of the code is technically accomplished are only briefly touched on in the present document. We will not discuss this challenge.)

8. Accurate and full documentation of the planning, results, and consequences of the validation activity, especially its implications for confidence in the intended predictive application of the code.

The elements shown in Figure 1.2 are discussed in greater detail in the remaining sections of this report. Our thrust is that these elements govern overarching guidance for the implementation of validation experiments. Validation and associated constraints on validation experiments cannot be separated from the requirements that are used to generate objectives for application of the code, nor can validation be separated from the ultimate predictions that are demanded of the code. This single observation tightly constrains validation experiment activities, as we will argue subsequently.

Although we tend to focus on the role of experimental validation in the development of necessary confidence in the predictive high-consequence application of a specific code, there is another goal of validation. It is clearly important to define code application domains where confidence in the code is *not* sufficient and why, therefore determining with some precision the ***boundary*** of applicability of the code. Such a goal may implicitly or explicitly result from validation experiments that *fail* to develop the required confidence in the intended application of the code. As discussed by Younger (1997), it is desirable to have experimental validation tasks that have the explicit goal of defining those application domains where use of the code is questionably adequate to better quantify the boundary of applicability of the code. To perform these validation tasks in a conscious and scientifically directed manner is not necessarily any easier than to achieve desirable levels of confidence in other application domains. McMillan (1996) also touches on this important philosophical point when he emphasizes the need to design validation experiments that explicitly challenge our ability to apply a code in simulating the experiments. He calls the resulting experimental validation efforts "Challenge Problems."

To support predictive, consequential applications of ASCI codes for stockpile stewardship requires new quantitative assessment procedures for the physical-science modeling community. A tradition of very complex and groundbreaking computational modeling certainly exists in the nuclear weapons community. However, there is currently no external regulatory climate that governs how this modeling is being applied to nuclear-weapon performance and management issues for the U.S. nuclear stockpile. Such a regulatory climate helped to spur the development of enhanced formal approaches to validation of code usage in analysis of the safety of nuclear reactors. It occurs to us that the stockpile challenges of tomorrow, given the current curtailment of full-system testing for nuclear weapons, place burdens on complex computational models that are quite similar to similar burdens arising from nuclear-power regulatory practices. Thus, enhancing the formal rigor of all aspects of validation is a logical goal, even if it is not currently mandated from outside the DOE weapons community.

It is essential that we focus attention on the formal aspects of experimental validation processes and their associated activities as the uses of codes become more consequential and the predictive capability of codes is emphasized. This document reflects this concern. It is also important that a large and complex V&V program like the ASCI V&V program at Sandia achieve useful levels of consistency in approach and expected results for the experimental tasks associated with validation processes. At Sandia the ability to achieve consistency is complicated by three factors:

- Formal principles, procedures, and metrics for experimental validation activities are in an early stage of development in the entire engineering community.

- A complex spectrum of experimental activities contributes information to the process of validating use of codes, sometimes in an ad hoc fashion. These activities include scientific discovery experiments, dedicated validation experiments, and system certification experiments, as well as the use of archival experimental data. Uniform experimental-validation concepts are difficult to apply to such a diversity of experimental activities.

- ASCI does not fund the relevant validation experiments required to validate applications of the codes. Implementation of uniform concepts then requires a difficult partnership between ASCI and the associated experimental campaigns.

It is also true that the ASCI program at Sandia involves a wide variety of codes, at diverse stages of development. This fact adds additional difficulty to the task of uniformly applying concepts for experimental validation.

Despite this complexity, achieving a degree of uniformity in approach and anticipated results for validation experiment activities should aid in the execution of, and increase the consequence in, these activities. Organizational processes also contribute to the administration and coordination of validation activities, including the dissemination of information that may be useful beyond validation tasks that are narrowly focused on a particular application of a code. Finally, organizational processes such as peer review are mandatory for performing quantitative scrutiny and assessment (Pilch et al. 2000b).

For these reasons, we believe that it is worthwhile to suggest and discuss a set of general concepts for experimental validation that are useful and appropriate in a broad sense. It is the purpose of this report to specify these concepts and develop the logic that underlies this specification. These concepts are necessarily constrained by their generality. More detail beyond the current concepts is dependent on the subject matter of the particular code or codes that are involved with the associated stockpile application and cannot be developed in general terms. It is not our intent to provide such detailed guidance even in one particular subject-matter area.

In Sections 2 through 10 of this report we develop concepts of experimental validation based on the elements in the validation and prediction process outlined in Figure 1.2. In each section, the concepts fundamental to a specific element in this figure are defined, followed by a summary statement of each concept.

We believe it is fruitful to measure the relationship of specific validation experiment activities to the concepts written in this report. In Section 11 we suggest such a means of measurement. We illustrate the application of our measurement principles in three particular experimental examples in appendices A, B, and C. In Appendix A we measure a previous validation study associated with a radiation-hydrodynamics application of the ALEGRA shock-wave physics code (Trucano et al. 1999). In Appendix B and Appendix C we measure ongoing studies of the development and application of “validation metrics” to structural dynamics in a normal environment and to the thermal decomposition of foam in an abnormal environment, respectively. These examples serve to illustrate the effectiveness of the stated concepts and our ability to apply them to assess the quality and success of particular validation experiment activities. Section 12 gives a brief summary and draws some conclusions based on the main content of this report. One final appendix, Appendix D, provides some additional information about the PIRT discussed in Section 3.

Section 2
Element 1: Application Requirements

2.1 Description

As discussed in the planning guidance (Trucano and Moya 1999; Pilch et al. 2000a), the most important principle underlying the Sandia ASCI V&V program is that validation is centered on specific applications, or uses, of codes. These applications of the code are called "stockpile drivers" in the planning references. An alternative phrase is "Directed Stockpile Work (DSW) Driver." These specified code applications concentrate validation requirements that originate in Defense Programs (DP) requirements at Sandia. DP requirements emerge from diverse sources of information, including military characteristics, Stockpile-to-Target Sequence (STS) requirements, Stockpile Life Extension Program (SLEP) requirements, requirements in program guidance for the various DOE campaigns, and requirements from other weapons program projects. The DP requirements that are embedded in the stockpile drivers include schedules and priorities that must be reflected in the particular choices of validation experiment activities that will be used to develop confidence in the code application.

The requirements for the code applications also influence the ultimate predictive intent for usage of the code in the associated applications. The related DP requirements either implicitly or explicitly specify scenarios, constraints, and application decision criteria for the code applications. These requirements provide, in principle, the primary information needed to construct the specific elements of validation experiments. For example, DP requirements may offer necessary conditions for specifying conditions for code usage in validation experiments, which include code parameter or variable domains, success and failure criteria, and qualification criteria, although this information may not be at all obvious.

Figure 2.1 suggests some of the aspects involved in integrating V&V activities with the needs of DSW. In this figure we have categorized four types of factors that must be considered to understand the role that future DP applications play in influencing V&V. These categories are labeled "DP," "Coordination," "V&V," and "Review." The horizontal direction in the figure denotes a generic measure of the progression of time. While units are not given or essential, realistic time scales for at least some DP application requirements are on the order of four to five years. Information flows across the categories, specifically concentrating in the coordination activities.

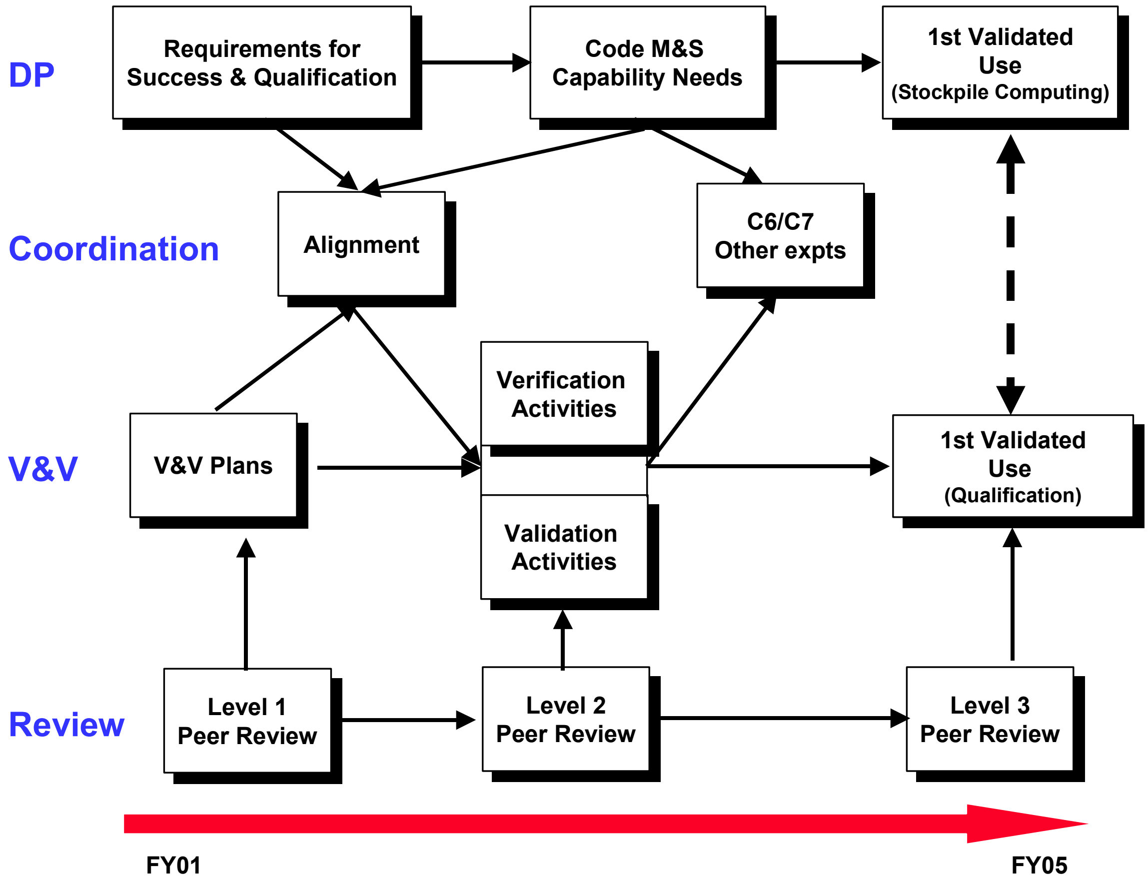

Figure 2.1. Integration of V&V program activities and DSW requirements for "1st Validated Use" of ASCI codes leads to a strong alignment.

Three major elements are emphasized in the DP category of Figure 2.1. First, there is the information collected in various forms that identifies requirements for stockpile programs and schedules. An important stimulus for these requirements and their associated qualification activities are the STS requirements. Second, specific modeling and simulation (M&S) capability is based on stockpile application requirements, as emphasized in Pilch et al. (2000a). This information may be modified over time by progress on a variety of DP activities. Third, "1st Validated Use" is a phrase that has been used in some formal planning exercises at Sandia to describe a decision point for determining when a code must first be used in an essential role for a given weapon-project activity. These decision points are associated, for example, with SLEP schedules.

V&V activities, identified in the third category of Figure 2.1, are generally directed by the planning that is governed by the DP category. This information transfer is currently most concentrated in integrated planning and alignment activities, one of the elements called out in the coordination category of the figure. Alignment emphasizes the most critical DP need: to best balance limited resources. The alignment processes are ongoing, constantly balancing the evolving planning, requirements, and schedules at the DP level with the evolving work of V&V. An important additional coordinating factor emphasized in

Figure 2.1 is the role of Campaign Six (the Weapons Engineering Certification Campaign) and Campaign Seven (the Hostile Environments Campaign). These experimental campaigns are directly engaged in the alignment processes and must be responsive to the requirements of DP, while also acting as the most important source of experimental data for validating application of Sandia ASCI codes to stockpile problems. The importance of the campaigns is a dominant coordination concern between DP and V&V, in our opinion.

Given the importance of "1st Validated Use" in modeling and simulation applications for DP projects, a logical goal of a directed set of V&V activities is to demonstrate the achievement by a given code of this requirement at a fixed point in time. Thus, we have pictured this element in Figure 2.1 as "Qualification," which is an important consequence of the progression of V&V tasks. DP clearly has a strong voice in whether or not this capability has been demonstrated.

We anticipate that the peer review process described by Pilch et al. (2000b), pictured in the fourth category of Figure 2.1, will play a significant role in achieving first validated use in both stockpile computing and qualification. We have clearly indicated a role for Level 3 Peer Review in the DP decision regarding "1st Validated Use." As described by Pilch and his colleagues, the Level 3 Peer Review is designed to aid directly in assessing the achievement of readiness for "1st Validated Use" for specific DP applications governed by the DP category in Figure 2.1. For further discussion about this topic, see Pilch et al. (2000b).

The interaction of V&V program activities, DP program schedules, executed DP project work, and the work of Campaigns Six and Seven is very complex. A more detailed discussion of this topic is beyond the scope of the present document. Our major conclusion is clear, however. Validation activities should be explicitly directed at assessing code capability for required DP applications.

From our perspective, it is crucial that planned and dedicated validation experiments be cognizant of the underlying code application and associated requirements. (**APP1**) The planning for the experimental validation project should recognize this application of the code and the STS requirements it addresses. Formal understanding of the code application requirements may be sufficiently documented in the code-application V&V plan. Additional information beyond the general V&V plan documentation may be required in the plan specific to the experimental validation project, such as (1) geometric details of the associated weapon system, (2) detailed knowledge of application scenarios, and (3) appropriate initial conditions and boundary conditions that are faithful to these scenarios. A dedicated validation experiment that is not built upon the existing defined application requirements in the code-application V&V plan is of limited value.

2.2 Application Requirements-Based Concepts

APP1: The validation experiment activity should be derived from the intended code application defined in an existing code-application V&V plan.

Section 3
Element 2: Planning

3.1 General Comments

Careful and formal planning is a critical foundation of the Sandia V&V program. Formal planning of experimental validation activities should be performed before the execution and analysis of these activities. The planning for particular experimental validation activities should also be well integrated into the larger scope of V&V planning for the particular code application that is the focus of the V&V.

A key component of V&V planning, especially of experimental validation activities, is the Phenomena Identification and Ranking Table (PIRT). This technique facilitates the collection and aggregation of information that is required to define and prioritize particular experimental validation activities. The PIRT aids in upward planning, which emphasizes the linking of one or more experimental validation activities to the driving code application. (**PLAN1**) The primary questions to be resolved in upward planning are (1) Why is the particular experimental validation activity needed for the driving code application? (2) Why is the particular experimental validation activity defined as it is? (3) What is the definition of success for the experimental validation activity? In the latter case, success is defined by such factors as success in gathering experimental data and in comparing calculations with experimental data. (**PLAN2**) As discussed below, the PIRT should be designed to capture the hierarchical approach to validation recommended by the Sandia V&V planning guidance, which also helps to further resolve the answer to the first two questions above. (**PLAN3**) The better defined and applied a PIRT is in an ongoing series of experimental validation activities, the more likely these questions can be answered before the experiments are executed.

The PIRT also enables more precise downward planning, especially that which guides the details associated with the experimental validation elements of “Experiment Design, Execution, and Analysis,” “Metrics,” and “Assessment” shown in Figure 1.2 and discussed in greater detail in subsequent sections of this report. Because of the complexity of aligning requirements of the DP, ASCI Applications Program, V&V program, and experimental campaigns mentioned above, these validation process elements directly influence schedule and cost estimates for experimental validation activities and should be accounted for in planning.

The planning details for the experimental validation activity should, of course, be formally documented. (**PLAN4**)

3.2 The Phenomena Identification and Ranking Table (PIRT)

As argued in version 2 of the Sandia V&V planning guidelines (Pilch et al. 2000a), the PIRT is the most important tool in our V&V planning process for translating requirements of the stockpile driver application into requirements on usage of the code, hence specifically on validation activities. The PIRT is particularly important for prioritizing and directing dedicated validation experiment tasks. The intended use of this methodology is thoroughly specified and elaborated in Pilch et al. (2000a) and is not repeated here. However, we do point out that the PIRT is designed to convert the DSW driver application and its associated requirements into specific technical requirements for the code, verification activities, validation activities, and consequent experimental validation requirements. It is the code technical requirements for the driving application that are the proper focus of V&V activities. As a result of a well-executed PIRT process, the validation requirements of the code application are rank ordered in importance. The prioritized PIRT elements directly create the definition and prioritization of the specific validation tasks, especially dedicated validation experiments, which are performed under the validation plan for the code application.

The PIRT is critical for planning validation experiments because it helps establish both ***sufficiency*** and ***efficiency*** of the validation activities. To demonstrate ***sufficiency*** requires a careful response to the question, What has to be done to establish a necessary level of confidence in the application of the code? To demonstrate ***efficiency*** requires evidence that limited resources (people, money, time) are balanced as a result of planning, not simply as a reaction to circumstances. We presume in this report that dedicated validation experiments supporting the validation assessment of a particular code application are directed at the most important elements in the associated PIRT. If this is not true, there is already a revealed weakness in the planned validation project. The planning for the dedicated validation experiments should make this direction explicit and clear.

The PIRT (and the underlying DSW driver requirements that it expresses) is also likely to provide information, or point to additional sources of such information, that are necessary for quantifying success or failure of the validation experiment activity. As discussed later in this report, a critical goal of any experimental validation project should be to define and apply success and failure criteria that assess the comparison of code calculations and experimental data in terms of revealed confidence in the application of the code. Any expression of acceptance criteria associated with the PIRT information must be addressed in the planning for the validation experiment.

The PIRT also provides links to application requirements governing code verification activities. Alignment with the PIRT is an additional means for assessing the level of code verification. Verification assessment of the code is an important, necessary condition before a planned validation experiment activity is conducted.

A complexity-based hierarchical approach to validation is critical for demonstrating that calculations agree with experiments for the right reasons. The design of the PIRT should reflect the recommended validation tier structure suggested by the V&V planning guidelines (Pilch et al. 2000a) that addresses hierarchical validation. Validation activities, including dedicated experiments, are logically structured in terms of physical complexity. The complexity of validation activities is given coherence by the PIRT and its detailed elements. The guidance (Pilch et al. 2000a) recommends that the Validation Test Suite (VALTS) reflect the complexity structure of the PIRT. A description of its position within the documented VALTS is one way that a planned validation experiment can be precisely located in the PIRT. Each specific problem in the VALTS, for example, could correspond to a single validation experiment project. However, we do not assume that this is generally the case for this report because it is likely that one or more of the problems identified in the VALTS will be dealt with through existing experimental data, not through the initiation of a dedicated validation experiment activity.

Recall that Pilch et al. (2000a) specify the following levels of complexity for validation activities that should be reflected in the VALTS:

- Tier 1—Single physics validation
- Tier 2—Validation for the simplest couplings in the application
- Tier 3—Validation for the full couplings in the application

An additional category, Tier 4, was defined for the VALTS in Pilch et al. (2000a). Tier 4 was defined to be "Qualification" activities. As used by Pilch and his colleagues, qualification is any experimental activity that is defined to be a formal acceptance test for application of the code to a stated stockpile problem. Because qualification is not properly the domain of our current report, it will not be discussed directly here. To understand the intent of this category, one can read the relevant material in Pilch et al. (2000a). We also provide some additional discussion in Section 9 concerning the requirements that predictive applications place on experimental validation, since in our view qualification is intrinsic to stockpile prediction.

The VALTS hierarchy is based on that suggested by Sindir, Barson, Chan, and Lin (1996) and the *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* (AIAA 1998), herein referred to as the "AIAA V&V Guide." The "Unit," "Benchmark/Subsystem," and "Complete System" categories discussed in these references map directly into the Tier 2 through Tier 3 categories above. There is no corresponding qualification category defined in the AIAA V&V Guide. There may also be some additional classification of hierarchical validation in the VALTS directly emerging from the DP application requirements, such as what might be defined in program plans of the experimental campaigns. We believe that alternative hierarchies of validation complexity can be faithfully translated into the Sandia-recommended VALTS structure above, so alternative approaches are not directly addressed here.

Validation complexity, defined through the PIRT and expressed in the VALTS, tends to be correlated with complexity of the experimental data, although this is not always the case. One form of this correlation is depicted in Figure 3.1. In this figure we have loosely suggested a possible dependence between the complexity of validation activities and the complexity of the data that might be acquired. The validation physical complexity reflects the validation tier structure specified in the Sandia V&V planning guidance (Pilch et al. 2000a) and is fully described in that document. The data complexity passes from high-level integrated system data through complex fully spatially and temporally varying data (time series, radiographs, photometric data, etc.). In Figure 3.1 we have also suggested a potential trajectory for a hierarchical set of validation activities through the validation data–physical complexity plane. The key to likely optimizing the impact of such a hierarchical experimental activity for validation of the code application under study is to rigorously account for the information, priorities, and requirements developed in the PIRT.

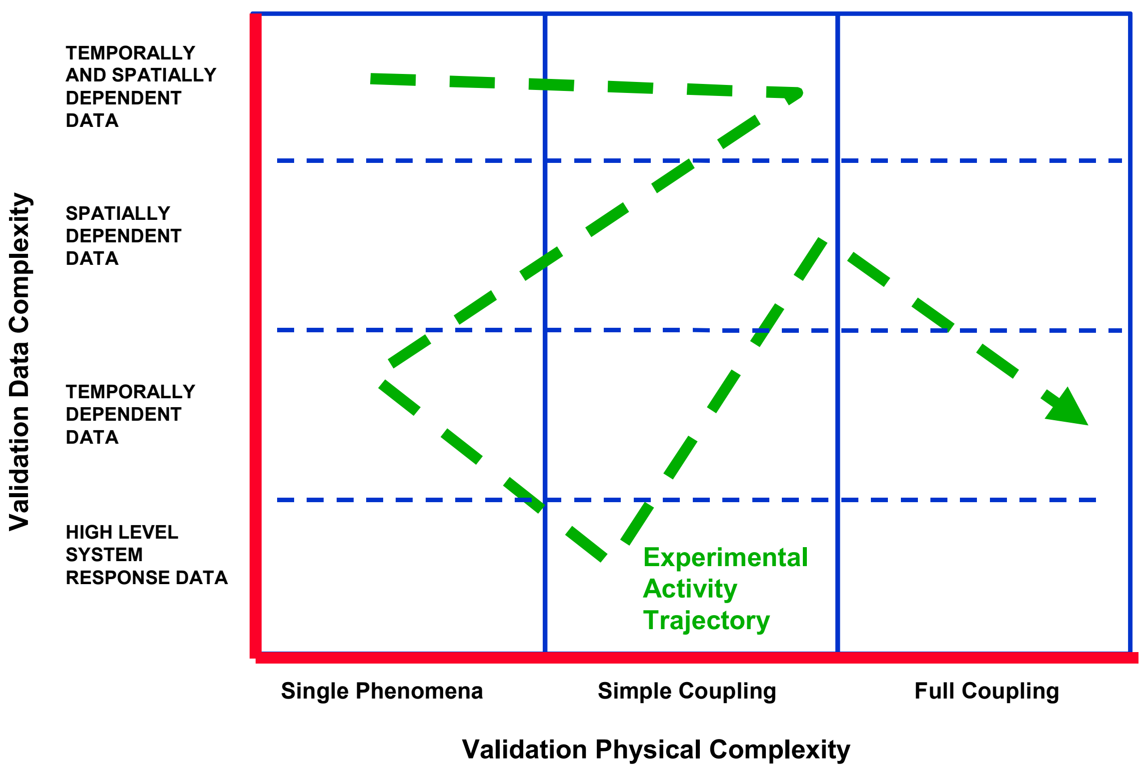

Figure 3.1. One projection of the validation process defined by a PIRT in terms of validation-data complexity and the complexity structure defined in that PIRT.

We conclude this discussion by observing that three important questions are answered by any dedicated validation experiment project that is properly linked to a well-crafted PIRT:

What additional validation activities—either involving dedicated experiments or existing experimental data, or both—should precede the current validation experiment activity?

What future validation activities—either involving dedicated experiments or existing experimental data, or both—depend upon the current validation experiment activity?

Can the current validation activity be simplified by removing components not relevant to the driver application?

Pilch et al. (2000a) observed that the PIRT is properly viewed as a process as well as a collection of information. As stressed by Boyack et al. (1990), the PIRT is most certainly not set in stone once it is formulated and documented. While a given formulation of a PIRT guides V&V activities it must also adapt to reflect the information gathered during the conduct of those activities. It is important in planning to recognize that to take the greatest advantage of its value, the PIRT can be and possibly should be adapted during the course of experimental validation activities.

Figure 3.2 shows the conceptual coupling between the PIRT, the execution of validation experiments, and their outcomes. We have isolated five categories of information that could reasonably be associated with the PIRT, either as part of the PIRT or from subsequent analysis of the PIRT. These categories are "Importance," "Conceptual Model Adequacy," "Code Adequacy," "Experimental Adequacy," and "Validation Adequacy." (One form these categories may take is described in Appendix D of this report.) The information in the five categories defines certain validation activities that are then performed, as well as the definition and application of metrics and the success and failure criteria that should be applied in assessing the outcomes of experimental-computational comparisons. The outcomes will influence the PIRT as well as the confidence we have in the code applications.

Several scenarios illustrate the potential interaction of validation experiment activities and revision and adaptation of the PIRT during the course of these activities:

- A validation experiment may be planned and conducted under the assumption that a specific PIRT element has a high importance. After the results of the experiment are analyzed, the importance of that PIRT element is found to change from high to medium or low in response to these results. (This does not argue that underlying DP requirements could or would change as a result of experiments, only that the technical importance of an element for validation may change.)

- An experiment is conducted that reveals a shift of a PIRT element from low to high importance. This may require an exploratory experiment as a succeeding experimental activity that was not planned at all in the existing PIRT.

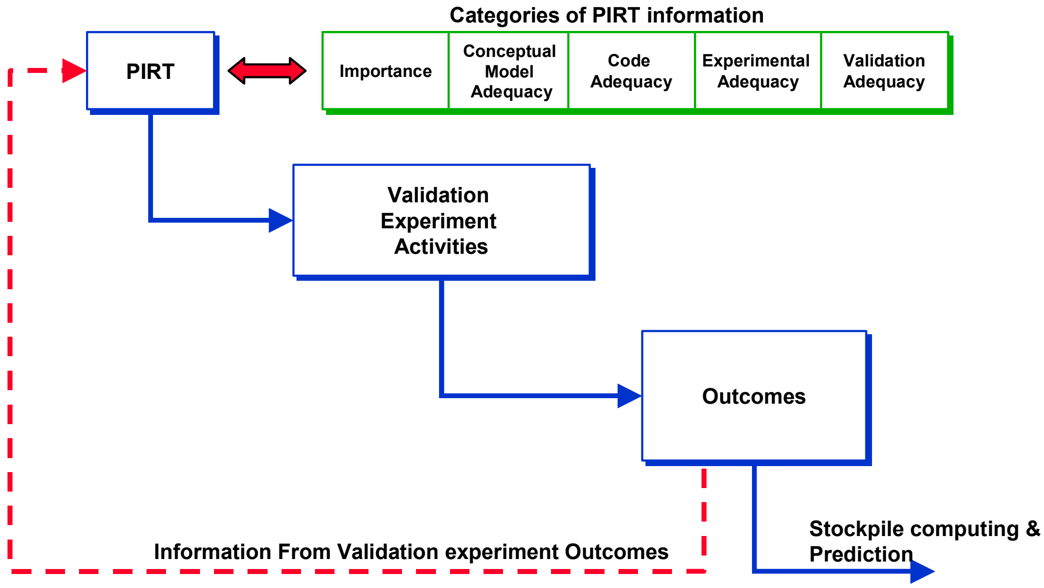

Figure 3.2. The coupling of the PIRT with the outcomes of validation experiment activities.

- An experiment is performed addressing a high-importance PIRT element. The current code implementation addressing that phenomenon is believed to be adequate. (Thus this experiment may also be some kind of qualification test for the code application.) However, it is discovered unexpectedly that the code cannot even function properly in defining the proposed experiments, thereby changing the ranking of the implementation to inadequate.

- An experiment designed to probe fully coupled phenomena reveals the presence of a completely unexpected and unknown phenomenon that is of high importance for the DSW driver. Not only must the PIRT be changed to reflect this event, but also the overall V&V effort for the code application may require significant revision. For example, a previously low-ranked phenomenon may now be ranked high, or a planned validation experiment may have to be redefined as a phenomenon exploration experiment.

- A validation experiment for a single phenomenon reveals that certain models implemented in the code must be recalibrated. This changes the code implementation from adequate to incomplete, and may require additional planning for *calibration experiments* to improve the current model capabilities.

It is clear that this kind of interaction between the PIRT and the results of validation experiment activities is almost limitless. The initial development of the PIRT is often largely subjective. This implies that there is a high possibility that subsequent experimental validation activities may change the PIRT. In addition, peer review activities suggested in Figure 2.1 may change the PIRT. For example, the Level 1 peer review could lead to PIRT modifications in the planning phases of V&V. The Level 2 peer review could lead to PIRT modifications during detailed planning and execution of experimental validation activities. Finally, continued evolution of system-level understanding could lead to PIRT modifications through the requirements and alignment elements suggested in Figure 2.1.

3.3 Planning Requirements-Based Concepts

PLAN1: The dedicated validation experiment activity should be part of a hierarchical validation activity that is defined by a PIRT. The planned validation experiments should then be well correlated with specific PIRT elements, and those elements should be clearly identified in the experimental plan.

PLAN2: Information relevant to defining success and failure for comparison of code calculations with the results of experiments should be identified in the PIRT.

PLAN3: The dedicated validation experiment activity should be defined in terms of the recommended Tier 1 through Tier 3 complexity structure if this is not explicit in the existing PIRT.

PLAN4: The validation experiments should be defined in a formal documented plan.

(Page Left Blank)

Section 4
General Comments on Experimental Validation

4.1 Overview

The key elements in the experimental validation methodology that are concentrated on the conduct of validation experiments are isolated in Figure 4.1. These elements are code and calculation verification; experimental design, execution, and analysis; metrics; and assessment. It is useful to keep in mind for this discussion that the elements presented in Figure 4.1 are necessary inputs and outputs for specific validation experiments, as will be explained next.

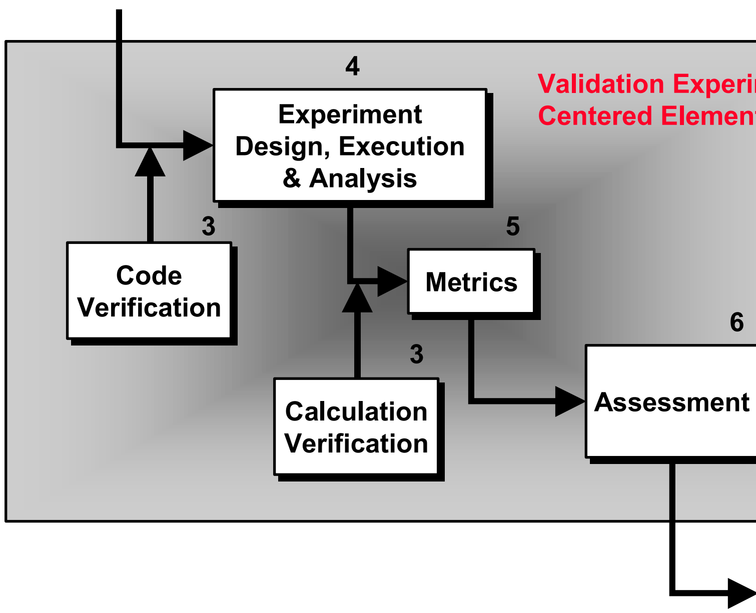

Figure 4.1. Validation experiment elements.

Inputs that are necessary for conducting validation experiments are code verification, calculations used to plan validation experiments and verification of those calculations, the definition of metrics used to compare calculations and experiments, and the specification of assessment criteria used to determine the success of the calculation-experiment comparisons. As explained in greater detail in Section 5, code verification implies the readiness of a code to be compared fruitfully with the results of validation experiments. We argue in Section 6 that the code subjected to validation should play an integral role in the predictive definition and design of validation experiments, as well as in postexperiment analysis. This task requires confidence in the functionality of the code; code verification should provide this confidence. Critical evaluation of the physical models in the code is, after all, the primary goal of experimental validation. At the same time, calculations that are used for the predictive design of validation experiments should be determined to be as accurate as possible. Confirmation of calculation accuracy is calculation verification, also described in detail in Section 5.

We find that it is logically convenient to separate the metrics used to compare calculations with experimental data, for example, the choice and application of particular norms, from criteria that are used to assess how successful these comparisons are. It is important to define both the metrics, discussed in Section 7, and the assessment criteria, discussed in Section 8, before validation experiments are executed. One's ability to define the metrics and the assessment criteria before the execution of validation experiments is a function of the planning process and the level of detail developed in the PIRT.

Outputs that logically result from executing validation experiments include experimental data, postexperiment computational analysis and verification of these calculations, results of applying defined comparison metrics to calculations and experimental data, and assessment of the results of these metrics. It is typically the case, even when predictive calculations contributed to the design of the experiments, that additional postexperiment computational analysis must be performed. For example, such analysis could aid in understanding the collected experimental data, serve as the basis for metric comparisons, or both.

Quantitative comparisons between computational and experimental results should be defined by the particular choice of metrics. Defining the metrics before the execution of validation experiments increases our ability to claim that the results of assessment of the calculation-experiment comparison are achieved for the correct reasons. Modifying or supplementing the defined metrics after the execution of validation experiments may also be required because the actual experiments may deviate from the planned experiments or because there was insufficient information in the planning stage of the experiments.

Comparing computational and experimental results without assessing the meaning of this comparison in terms of increased or decreased confidence in the code is a near-empty exercise. Assessment is the most important output of a directed validation experiment activity. Determining a quantitative level of confidence in the capability of the code in the domain of validation experiments is of increasing importance the greater the degree of overlap that exists between the parameter domain of the validation experiments and the parameter domain of the required code application. All of the elements in Figure 4.1 are defined and directed to achieve this goal in an optimal validation experiment.

4.2 Existing Guidance for Validation Experiment Activities

In this report we do not present a review of existing literature that addresses the topic of validation methodology and procedures. This literature is quite extensive and represents distinctively different perspectives and approaches, ranging from engineering and physics to operations research. Recent reviews of the literature are given in Kleijnen (1995), Balci (1997), Roache (1998), Kleindorfer et al. (1998), Oberkampf (1998), Murray-Smith (1998), Robinson (1999), Oberkampf and Trucano (2000), and Oberkampf and Trucano

(2002). As a supplement to these reviews, we will now make several general comments about the existing literature, specifically from the perspective of physics-based simulations.

Roache (1998) devotes two chapters to summarizing validation-centered activities. As such, and given their context in his most extensive discussion of V&V, we believe these chapters are of intrinsic interest. The emphasis of Roache's discussion is on practical guidance that is directed at modeling areas where he has personal experience, most notably computational aerodynamics and geophysical fluid mechanics. Roache's exposition focuses on examples from the published literature and points that can be made about these examples. Of importance to us is his cautionary tone. Specifically, Roache carefully discusses questions that arise from the need to infer confidence in computational simulations from comparison with experimental data. These questions reflect Roache's concerns about the usefulness of *existing* experimental data for validating specific applications of the code.

Roache's cautionary tone originates in his implicit focus on how to approach existing experimental data as potential information for validating specific applications of the code. He does not devote significant attention in his discussion to the specific problem we address here, which is guidance for developing and executing *dedicated validation experiments*. We believe the shift in emphasis is subtle, but important, and allows us to convert a variety of warnings into specific targets for action in considering how to perform validation experiments. Implicitly, Roache's discussion is aimed at an audience that will approach validation from the point of view that most of the necessary experimental data they will need for validation will not have been gathered in activities in which they directly participate. In our seasoned view, most existing data are of such limited value that it is doubtful that high-quality validation can proceed without dedicated experiments of the kind we discuss here. We, in effect, *require* that the weaknesses in data highlighted by Roache be eliminated, and we claim that such should be the stated goal of any high-quality validation project. Our experience supports our view that weaknesses in the data, such as incomplete specification of experimental boundary conditions, do not allow mathematical models of physical processes to be critically evaluated.

We find that other treatments in the literature are considerably less helpful than Roache's discussion, except for AIAA V&V Guide. The AIAA V&V Guide (AIAA 1998) *does* discuss the design and execution of validation experiments, but the presentation is built so closely upon the work of Oberkampf and Aeschliman (1992), Aeschliman and Oberkampf (1998), and Oberkampf and Blottner (1998) that we will effectively be discussing this work below.

It is often claimed that ASCI software is unique; for example, see DOE/DP (2001b). Concerning the demands ASCI codes place on experimental validation, this appears to be a true statement. And while one might reasonably hope that a useful formal prescription for validation experiments that addresses the unique validation requirements of the ASCI program might be found in more detailed articles available in the journal literature, the fact is that this is unlikely. The recent review of V&V by two of the authors (Oberkampf and Trucano 2002) did not uncover published guidance for experimental validation of computational science and engineering that addresses the topics of concern to us in this report. It is possible that there exists unpublished proprietary guidance in various venues that may likely address at least some of the elements in this report. Such literature is virtually useless for our purpose because it is not accessible and, therefore, is equivalent to being nonexistent. It is worth stressing that not even the current ASCI program plans (DOE 2000; DOE/DP 2001b) mention validation experiments, let alone any kind of guidance about their conduct. These documents do stress that linkage to the campaigns is essential. This, of course, expresses a certain element of programmatic reality—the ASCI program does not fund experimental work.

We believe the implicit position of the computational science and engineering community on validation experiments is "***the design of validation experiments is left to experimenters***." This turns out to be a self-fulfilling prophecy. It not only explains why Roache and Oberkampf and his colleagues believe it is critical to provide significant warnings about how "dangerous" experimental data are for validating specific applications of a code, it also creates the climate in which experimental data will continue to be dangerous for validating specific uses of a code. We simply must progress beyond this state of affairs to have any hope of achieving the stated objectives of modeling under the ASCI program, even at the risk of re-inventing a wheel.

There is one example of existing general guidance on the conduct of validation experiments that is fully appropriate to our present task. This is found in the paper of Aeschliman and Oberkampf (1998). It is worth repeating the key principles that underlie the approach these authors take to designing dedicated validation experiments:

I. **"A validation experiment should be jointly designed by experimentalists and code developers or users working closely throughout the program, from inception to documentation, with complete candor about the strengths and weaknesses of each approach."**

II. **"A validation experiment should be designed to capture the essential physics, including all relevant physical modeling data and initial and boundary conditions required by the code."**

III. **"A validation experiment should strive to emphasize the inherent synergism between computational and experimental approaches."**

IV. **"Although experimental design should be developed cooperatively, complete independence must be maintained in obtaining both the computational and experimental results."**

V. **"A hierarchy of experimental measurements of increasing computational difficulty and specificity should be made, for example, from globally integrated quantities to local measurements."**

VI. **"The experimental design should be constructed to analyze and estimate the components of random (statistical) and bias (systematic) experimental error."**

Although Aeschliman and Oberkampf (1998) stated these principles specifically in the context of computational fluid dynamics (CFD) validation activities, the principles are fully meaningful across the broad spectrum of validation activities that are required to support the Sandia ASCI program. These principles are properly recognized in the concepts articulated next.

We believe that the guidance developed in this report has greater scope and prescription than the principles and practices developed by Aeschliman and Oberkampf, although their work forms an important core of our present work. Three themes account for our increased scope and prescription. First, the technical components of the Sandia ASCI program are much broader than the earlier CFD context. Second, the present discussion of experimental validation concepts must smoothly integrate within the larger structure of organizational guidance that has been developed for the Sandia V&V program. This requirement creates programmatic and application prioritization features that our discussion must address which were not previously dealt with. For example, Aeschliman and Oberkampf did not discuss the PIRT as a planning element. Finally, our experimental validation concepts are focused upon the integral role of validation experiments within an overall *prediction* process. We are speaking to validation concepts that directly support the *predictive application of codes* with quantitative estimates of predictive uncertainty. We are striving to make this integrality as explicit as possible in the current document.

4.3 Existing Validation Data

While dedicated validation experiments are the most desirable validation experiment activities, realistically they will not be the *only* validation experiment activities. One obvious illustration of this concept is that, in the absence of further nuclear testing during the course of certain experimental validation projects, *existing* nuclear test data will be an important source of validation information. We expect that every experimental validation project will have to use existing data as well as dedicated validation experiments. For example, DOE (1998) lists existing experimental data as well as stringent dedicated experimental programs as key elements in validation activities. That document also lists classes of potentially useful existing data for validation of applications of codes to stockpile stewardship problems. These classes are

- archived data from nuclear tests
- archived data from nonnuclear tests
- fundamental physics experimental data
- systems certification test data
- stockpile surveillance data

Surveillance data, nuclear test data, and other classes of existing data are thus important, but are also more difficult to use than ongoing dedicated experimental programs because of the lessened assurance that all of the knowledge needed to perform validation activities is correctly and completely specified.

Our basic recommendation for validation activities is to use existing data if necessary. However, these data must be demonstrated to meet concepts that cannot be substantially different from those that are laid down for dedicated validation experiments in this report. (**EED1**) Many of the concepts stated in this report are just as useful for applying to existing data as they are to data acquired from a dedicated validation experiment. Yet, under most circumstances, we emphasize that existing data are unlikely to be an adequate replacement for a dedicated validation experiment activity. This view has also been recognized and emphasized by others (Younger 1997; McMillan, 1996).

4.4 Existing Experimental Data-Based Concepts

EED1: All applicable concepts in this report should be applied to guiding the use of existing experimental data in experimental validation activities.

Section 5
Element 3: Verification

5.1 Description

According to the ASCI program plan (DOE 2001), "The most essential goal of the verification portion of ASCI V&V is to ensure that the models from Advanced Applications do in fact give the analytically correct answers as they are implemented." It is essential that validation activities be performed with a code that has achieved a significant level of verification to best meet the requirement of achieving correct results in validation for the right reasons. This does not mean that we require some kind of proof that the code is bug free, or that all of the numerical algorithms function properly. That would be impossible. Rather, we believe it should be established that significant effort has been expended to verify the code to the greatest degree possible given the available resources and the schedule of validation tasks. Appropriate documentation of this level of effort in verification will provide enough confidence in the code to make its use in experimental validation activities reasonable and fruitful.

Two types of verification are generally recognized (Roache 1998; Oberkampf and Trucano 2000, 2002) as implicit in the ASCI program's view of verification and are required to perform validation activities. The first type is ***code verification***; the second type is ***calculation verification***. As will now be stressed, both types are important for applying a code in validation experiment activities. Both types were presented previously as concepts in Figure 1.1 and are necessary inputs into the experimental validation elements.

Code verification involves determining whether or not the code as a software system is appropriate for use in validation experiment activities. Two distinguishing issues are important in code verification. First, code verification should address the issue of "code as a product." Verifying the ***code as a product*** focuses on whether or not the code is sufficiently mature for use in defining, designing, and analyzing validation experiments. Applying a code in validation experiment activities involves ease of use, effectiveness of use, and stability and robustness of the code system. Deciding the degree to which these elements are true for a given code centers on the code development activities and support infrastructure of the code, which include various elements of software quality engineering (SQE) as well as other general factors.

The second issue addressed by the process of code verification is correctness of the numerical analysis factors that influence the numerical accuracy of the algorithms implemented in the code. This issue is of paramount importance in computational science and engineering codes, whereas in traditional SQE this issue receives little emphasis. This aspect of code verification might also be called ***algorithm verification***, but our preference is to subsume it under the more general umbrella of code verification in this report. The major point is to achieve demonstration and cumulative evidence that the numerical algorithms in the code are implemented and functioning properly. This topic of algorithmic verification activities is generically associated with that of code development activities. One verification component at Sandia that specifically addresses this issue is the existence and application to the code of a Verification Test Suite (VERTS), as defined by Pilch et al. (2000a). In the context of experimental validation, this issue of code verification should answer the question, Is the assessed numerical accuracy of the code on classes of problems relevant to the validation application sufficient to justify the use of the code to define, design, and analyze validation experiments?

Assessing the accuracy with which the code calculates test problems in the VERTS requires ***calculation verification***. This is the formal quantitative demonstration that an acceptable level of accuracy is attained by the code for each selected test problem. Usually, such a demonstration of accuracy requires demonstration of spatial, temporal, and/or iterative convergence of code calculations to the correct answer of the test problems defined by the VERTS. The importance of confirmation of code accuracy on selected test problems is discussed in the AIAA V&V Guide (AIAA 1998).

As applied to validation calculations, calculation verification is directed at characterizing the numerical accuracy of specific validation calculations. It is primarily aimed at resolving the problems of numerical convergence (space, time, and iterative) and *a posteriori* error estimation for complex validation simulations. Oberkampf and Trucano (2000) discuss particular underpinnings of validation upon this issue. However, unlike the case of VERTS test problems, it is more likely that convergence may not be established in validation calculations, given their expected greater complexity. Hence, empirical *a posteriori* error estimation will likely be very important for this endeavor.

The goals of code verification and calculation verification are both difficult for validation activities involving ASCI codes. Code verification is an especially difficult problem in any critical validation activity because of the complexity of ASCI codes and their varying levels of maturity. As mentioned above, verification of the accuracy of validation calculations through convergence studies is likely to be difficult because of the probable complexity of the calculations. For example, it is unlikely that calculations used to design a complex three-dimensional, multiphysics validation experiment can be converged. To properly deal with these issues in a given validation experiment project requires a combination of existing code-performance measures, user experience, and information from existing historical work using the code, as well as new accuracy assessment tasks specifically centered on the current validation experiment activity. We do not believe the validation analyst should be responsible for building the definitive case for code verification. Rather, it is the analyst's responsibility to understand and convey, based on documented evidence, what has been done at the time of the experimental validation activity.

Ideally, it is desirable to perform verification and validation (V&V) as serial processes by *first* successfully completing required code verification and *then* beginning validation and its attendant needs for calculation verification. Unfortunately, practicality requires that code verification is ongoing while calculations associated with experimental validation activities are performed. This mandatory interaction of V&V complicates the process of validation, but does not on its face offer a sufficient reason for not attempting validation.

The intellectual need to separate verification questions from validation questions is the basis of our conception that a minimal level of code verification evidence is needed before validation is started. Why waste one's time validating the application of a code that does not function properly or that has few, if any, verification test cases similar to the validation case? Also, since danger lurks in comparing experimental results with code calculations that are not strongly believed to be the product of properly designed and implemented numerical algorithms, any amount of evidence to the contrary increases confidence that the validation activity will be worthwhile. If verification confidence is weak because of lack of evidence, how can one strongly conclude that the basis of a disagreement between experimental results and code calculations is in the conceptual models embedded in the code rather than in the implementation of the numerical solution?

We cannot create a definitive list of requirements for establishing minimal verification of the code and its associated calculations. Many important verification factors will be specific to the particular code, the particular calculations, and the anticipated results of the validation activity. However, certain requirements do seem to be generally applicable, and we have listed them below. These requirements form the core of the concepts for verification that we consider part of the validation experiment activity.

- Reliance should be placed upon the code development team and any existing supplemental verification efforts to provide part or all of the needed ***code verification*** information. This information should be documented, either as part of the code development process or as part of the documentation that is uniquely generated for a given validation experiment activity. The information should include SQE information on relevant processes, code version, code support, code robustness, and code maintenance. Documented information about the software-testing performance of the code (unit, regression, benchmarks) should be available and referenced. This information is expected to be part of the documentation tree for V&V activities for Sandia ASCI codes (Pilch et al. 2000a) and draws upon guidance developed in fiscal year 2001 for SQE-related code development practice at Sandia (DOE/DP 2001a; Aragon et al. 2002). (**VER1**)

- Specific attention should be devoted to ***calculation verification*** of the VERTS associated with the particular code, with additional emphasis on those VERTS tests, if any, that are relevant to the code usage required for the validation experiment activity. For any VERTS tests that are aligned with the required code application, convergence and *a posteriori* accuracy assessment of these verification tests should be established if not already previously documented. Of course, it may be the case that no VERTS tests can be claimed to align directly with the required code application. (**VER2**)

- If no existing VERTS tests are specifically aligned with the required code application, then new VERTS elements should be defined that do align with the planned code usage for the validation activity. For these tests ***calculation verification*** should be performed. (**VER3**)

- The basis for performing ***calculation verification***, for assessing convergence and numerical accuracy in the needed validation experiment calculations, should be documented. Again, the overall goals of code calculations supporting validation experiment activities are to define, design, and analyze validation experiments. Each of these uses of the code for validation experiments requires some attention to convergence and accuracy of calculations. The most complex validation calculations will typically present the greatest difficulty for performing the necessary calculation verification. Yet a compelling basis for believing in the consistency and accuracy of the related calculations should still be formulated. VERTS-related tests, as well as preceding validation activities, all contribute experience and measures of calculation accuracy that are relevant to understanding the accuracy of more complex calculations. Calculation verification is traditionally an intuitive and ad hoc component of validation calculations. We believe that this relationship should be formalized and documented. (**VER4**)

- All associated information on code verification and calculation verification should be documented in a way that allows historical tracking and repeatability of the relevant computational work. (**VER5**)

Consider the following example. Suppose it is observed that the code at issue cannot successfully run calculations for validation experiments, for instance, because of code bugs or obvious numerical errors. This inability would first become evident in exploratory calculations intended to help define a validation experiment. (The purpose of validation experiments is not to demonstrate such limitations, of course.) The conclusion should then be that the code is *not* minimally verified for the pursuit of validation experiment activities. In our view it makes little or no sense to pursue the said validation experiments until the code is minimally verified.

An alternative example is the case where the code in question *can* run calculations that can be compared to validation experiments, and where the code has “passed” extensive VERTS testing that is well correlated with the needs for the planned validation activity. The conclusion in this case is that the code *is* minimally verified for the pursuit of validation experiment activities, and it makes better sense to expend time and money in the execution of the planned experiments.

In neither of the above examples is the issue whether or not code calculations are in good agreement with experimental data. Instead, the point illustrated is the seemingly obvious requirement of whether or not code calculations can even be usefully compared with experimental data.

5.2 Verification-Based Concepts

VER1: The code verification status should be understood by the validation analyst and documented and determined to be adequate for the pursuit of an associated validation experiment activity.

VER2: The existing VERTS for the code should contain elements that are believed to be in alignment with the associated validation experiment activity. Calculation verification should be performed and documented for these specific VERTS elements.

VER3: New VERTS elements should be defined if there is inadequate coverage in the code VERTS to contribute to assessing code verification status for the planned validation experiment activity. The calculation verification of these new elements should be performed and documented.

VER4: A calculation verification strategy (typically centered on convergence studies and *a posteriori* error estimation) should be defined for the calculations performed in the validation activity.

VER5: All necessary information required for the verification assessment for the validation experiment activity should be documented.

(Page Left Blank)

Section 6
Element 4: Experiment Design, Execution, and Analysis

6.1 General Description

We have emphasized that validation experiments must be designed purposefully with specific goals linked to application objectives and to specific elements of the PIRT. The primary goal of directed validation experiments is to ensure that data are sufficiently related to the application driver for the code to provide stringent confidence assessment when code calculations are compared with these data. These data must be precisely located within the structure of the application-specific PIRT, and must be unambiguous in defining confidence in the application of the code to the phenomena exemplified in the experiment. Because this task must be accomplished through the act of quantitatively comparing code calculations with experimental data, important requirements are placed on validation experiments to create the greatest opportunities for performing these comparisons. It is of the essence to design and execute validation experiments that allow precise and conclusive comparisons of calculations with experimental data for the purpose of assessing code fidelity and credibility. We devote Section 7 of this report to further the discussion of code-experiment comparisons. It will hopefully be made clear that the design and execution of experiments must allow us to achieve the quantification of meaningful and useful metrics. (**DES1**)

Particular validation experiments may achieve these goals to a greater or lesser degree, but any such attempt rests on a foundation of rational design concepts based on the needs of the application of the code. The more purposefully we design a validation experiment activity, the greater the probability that we can optimize the results of that activity in the face of these complex constraints. It is imperative that validation experiments balance resource constraints, including time, level of effort, available expertise, and desired fidelity. The approach for achieving this balance should be defined in the experimental plan. The needed resources are strongly coupled to the code capability under assessment and the required accuracy of the validation exercise, as defined by the specified stockpile application. (**DES2**)

The experimental, computational, and comparison activities should expedite the acceptance of computational models for system applications that extend beyond the limits of the validation activity within the application domain. This requires that the experimental, computational, and comparison activities associated with validation experiments should be able to adjudicate between competitive computational models, for example. This is one example of a problem that requires accurate determination of the intersection of the domain of acceptability of a model (as determined by validation experiments) with the domain required by the application. In particular, it is important to quantify the boundary separating the region of acceptability of the model from the region where the model is not acceptable for the application. In the case where extrapolation of the application of a model beyond the domain of validation is required, it is even more crucial to identify the region of acceptability and its boundary. This task requires the design of experiments that are expected to lie in the domain of acceptability of the model as well as experiments that sharpen our understanding of the boundary of that domain of applicability. (**DES3**)

Figure 6.1 illustrates the problem at the core of judging the boundary between believed acceptable and unacceptable code application. We show a conceptual view of the application domain and a subset of this domain referred to as the acceptable domain. These domains are defined as a multidimensional parameter space where only two of these dimensions, X_1 and X_2, are shown. Two kinds of points are also shown. An "A" denotes a location in the application domain where a specific application of a code has been previously performed; a "V" denotes a location in the domain where validation has been performed. The boundary of the acceptable domain represents the apparent limit, based on expert opinion, where the physical models in the code can be applied. We draw the reader's attention to two pairs of intended applications, denoted "A_I" and "A_O", respectively. The applications "A_I" each lie on the believed acceptable side of the boundary, while the applications "A_O" each lie on the believed unacceptable side of the boundary. Our point is that it is essentially as important to know that the code should *not* be applied to the applications "A_O" as it is to know that the code is acceptable for the applications "A_I". A complete validation effort could quantify this situation, but this is commonly not possible because of programmatic schedules, budgets, safety and environmental concerns, and international treaties.

Designing experiments that test a code in regions where the code is believed to be insufficiently accurate for the intended application helps locate this boundary, as well as provides a means for quantitatively assessing the degree of the expected inaccuracy. Because such experiments are performed purposefully rather than accidentally, these experiments also further test our grasp of the conceptual models underlying the code that are probed by the validation experiments. Obviously, this goal only makes sense when experiments that probe code inaccuracy lie close enough to the boundary of applicability to be relevant. For example, demonstrating that a Navier-Stokes fluid dynamics code will perform inadequately in modeling a reentry vehicle flow at an altitude of three-hundred thousand feet is hardly a relevant activity. Younger (1997) discusses this goal and observes that it is typically missing in critical system tests and unlikely ever to be attempted in such tests. (**DES4**)

There are still other questions that must be addressed in purposeful experimental design. For example, given the choice between two potential experiments, there should be a logical basis for choosing one or the other based on available resources and potential benefits. The code-application V&V plan is of some assistance in doing this, especially through utilization of PIRT prioritizations. In partnership with the PIRT, statistical experimental design (for example, see Cox [1958] and Dean and Voss [1999]) is an attractive basis for attacking the resource optimization problem for experiments with a focus on uncertainty quantification. Gunter (1993) has argued in favor of the utility of statistical experimental design in planning physical science experiments. The specifics of how statistical design of experiments may be applied in particular validation activities are dependent upon the subject-matter focus of these experiments. (**DES5**)

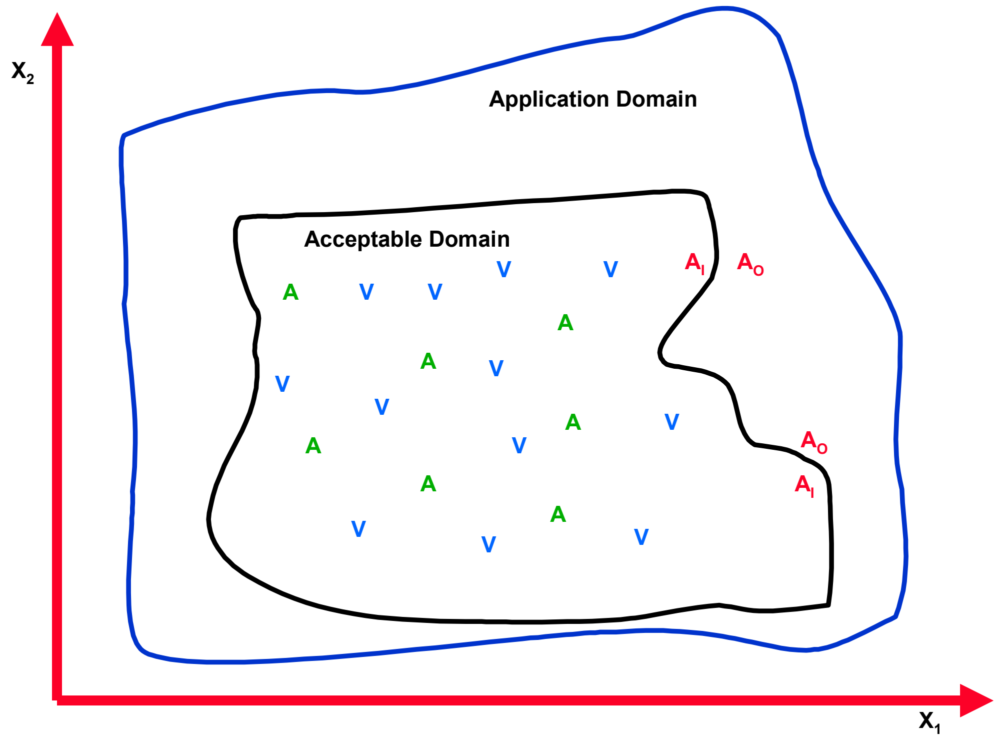

Figure 6.1. The application domain, with the boundary between acceptable and unacceptable model performance emphasized.

The experimental, computational, and comparison activities should quantify variability of the predicted results for the intended application scenario that may be due to poorly quantified environmental and material factors (in the case of component applications, for example), as well as to uncertainty in the scenario itself. This recommendation allows computational predictions of design margins that are relevant, with success dependent upon the ability to quantify uncertainty in the execution of validation experiments and in the resulting data. Uncertainty quantification in validation experiments is a critical problem—one that is revisited in our discussion below on validation metrics. Quantification of uncertainty for the experiment is generally dependent upon diagnostic resolution, experimental variability (both controlled and uncontrolled), and characterization of the experimental facility. Methodologies should be applied in validation experiments that provide information about uncertainty in these experiments. Statistical analysis of experimental results will likely have to be performed to define experimental uncertainty, and is required for the definition and application of high-quality validation metrics. For example, Aeschliman and Oberkampf (1998) and Oberkampf and Trucano (2000) stressed the importance of quantitatively estimating experimental random (precision) error for validation experiments. This can be accomplished using traditional methods that propagate individual components of random error through the entire data-flow process (Coleman and Steele 1990). This technique estimates these components and their interactions at each stage of the data flow process, from the sensor level to the experimental-result level. A different approach to estimating random error has been recently developed by Oberkampf and Aeschliman (1992) and Oberkampf et al. (1995). Their method is referred to as ***a posteriori experimental uncertainty*** estimation because it statistically analyzes final experimental results. (**DES6**)

A second recommendation is to attempt to estimate experimental bias errors. The most common method of attempting to estimate experimental bias errors is to conduct independent experiments. Independence between experiments can be attained to some degree by having independent teams conduct the same experiment, preferably from separate organizations using separate facilities. Using different experimental measurement techniques to determine the same quantity can also produce an increased degree of independence in experimental data. A different approach to estimating experimental bias error is based on statistical estimation and physical symmetry arguments (Oberkampf et al. 1995; Aeschliman and Oberkampf 1998; Oberkampf and Trucano 2000). The approach is based on the *a posteriori* error estimation previously mentioned, plus special construction of the experimental design and execution.

Ultimately, the data gathered in validation experiments should have ***robustness*** and ***specificity*** appropriate for their intended application. ***Robustness*** of the data is particularly important. By this term we simply mean that the interpretation of the experimental data gathered in the experiments is not ultimately questioned or debated or unclear. The interpretation of the data must serve as a benchmark with which to compare code calculations for purposes of validation; uncertainty in the interpretation of the data defeats this goal. Our concept of data robustness signifies our desire that comparisons of calculations and experiments for validation purposes should be a well-posed problem. Without well-posed comparisons of calculations and experiments, validation activities certainly have diminished, if not negligible, value. Achieving robustness of experimental data in this sense is often not possible in data gathered in scientific discovery experiments; such data tend to raise as many questions as they answer. See, for example, the general discussion in Collins (1992) and Franklin (1998) of problems associated with interpretation of specific examples of discovery experiments. A broader issue underlying our concern about data robustness is, of course, the challenging problem of ***data validation***. This subject is fully in the domain of the experimental community and is therefore beyond the scope of the present document. Data validation certainly relies upon full characterization of experimental equipment which may further increase the time and expense of dedicated validation experiments beyond those associated with other types of experiments. (**DES7**)

Coupled to the overall problem of experimental uncertainty is the problem of the specificity of the experimental data gathered as discussed above. What we mean by ***specificity*** is the notion that we gather *specifically* the kind of data that we need in order to perform validation. Typically, specificity thus means that the type of data needed (temperatures versus pressures versus stresses, for example) is what is actually collected. Specificity also demands that the data be collected at the spatial locations where they are required and at the times that they are required. Specificity may often be characteristic of discovery experiments, but it is less often characteristic of large-scale field tests.

6.2 Defining, Designing, and Analyzing Validation Experiments

Any appropriate methodology for the purposeful design of experiments for validation of code applications must consider the stockpile application driver and the code-application PIRT for that driver. In our view, the key features enabling rational design for validation of a code application are to ***define*** the expected results of the activity using the code itself; to ***design*** specific validation experiments by using the code in a predictive sense; and to have a structured plan for ***analyzing*** the results of the performed experiments with the code, with emphasis on the difficult comparison activities that form the core of the metrics element discussed in Section 7. (**DES8**) These activities emphasize tight coupling of the subject code to the experimental activity and ensure we are in the best position to productively learn from the comparison of calculations with experimental data, independent of whether these comparisons are good or bad, as explained below.

Consider the following illustration of the differences between defining, designing, and analyzing a validation experiment. Suppose through a series of exploratory calculations that a specific kind of sensitivity to certain physical variables appears in an application of a code and that it is desired to assess the correctness of this determination in a validation experiment activity. Deciding that such an activity should be pursued illustrates the act of ***defining*** a validation experiment. Specific PIRT elements for the code application may implicitly or explicitly assume the sensitivity to be validated, which is how such an experimental effort might be aligned with the PIRT. The code that is the object of the validation exercise should participate in ***defining*** the principles and goals of a planned validation experiment. An alternative phrase to keep in mind when the word “define” appears is “Define the purpose of the experiment.” For example, definition in this sense is a result of doing calculations that suggest the need to perform validation experiments assessing thermal transport in organic foam rather than for composite materials. The purpose of these example experiments suggests that validating the code application for thermal transport in foam is more fundamental, or has a higher priority, than a similar type of experiment for composite materials. The role of the code in the definition of validation experiments may be purely exploratory until some progress has been made on validation activities. We expect that such definition activities logically correlate with the process that generated the associated PIRT and its results. In fact, application of the code to defining validation experiments immediately helps to make more quantitative what may only be qualitative in parts of the PIRT.

An example of the ***design*** of a particular validation experiment is to use code calculations to provide specific guidance about where to locate instrumentation and what kind of data to acquire to assess the anticipated sensitivity. In the proposed validation experiments, “design” means specifying to the greatest degree possible the initial and boundary conditions, material properties, diagnostic locations and characteristics (Strain gauge? Stress gauge?), and data fidelity. The probability that these conditions “defined” through code calculations will be met identically in any complex validation experiment is probably zero, but the probability that they will be met in simpler experiments is relatively high. In most cases, the success of a validation experiment will often be determined by the degree to which the experiment matches these specifications. Deviations can be acceptable—but if the intent of the experiment was to measure **x** at location **y** for material **z** and geometry **g** and in all cases the experiment is completely different from most or all of these factors, it is unlikely that the experiment was successful. Unsuccessful dedicated validation experiments are of little value for code-application validation.

We don't require the expectation of accurate prediction of experimental results to justify the application of the code to the design of validation experiments. Why is it appropriate to have the code play a role in the design of validation experiments if we do not anticipate or require that the code can perform this task accurately? There are several reasons.

1. Design of a validation experiment ensures to some degree that optimally useful comparisons can be made between the calculations and the experiments. Experimental design using the code helps us avoid the syndrome of comparing apples and oranges after the conduct of the experiment as much as possible. A good example of this problem that often arises in integrated tests is ineffective diagnostic placement, where a critical measurement was required at location **x** but gathered at location **x1** instead. We elevate this point by calling it ***Experimental Design for Computational Analysis***, and emphasize that this is a key objective of validation experiments.

2. Designing validation experiments using the subject code forces computational prediction of the results of the planned validation experiments more than any other factor. Predictive use of code calculations in experimental design activities is of paramount importance. This, more than any other single factor, distinguishes dedicated validation experiments from experiments in which codes are used to simply analyze the experiments after they have been performed. The increase in risk associated with prediction is also a better match for the anticipated application of the code in high-consequence prediction. We may not be able to quantify it, but success of true prediction of a code has a far greater impact on our confidence in that application of the code than simple retrospective agreement, which could have happened for both good and insidious reasons. On the other hand, no factor makes the failure of a code more dramatic as the anticipation of measuring data of a certain kind and finding they are significantly different from predicted. Better to discover such an anomaly in a dedicated validation experiment than in important stockpile computing. Designing validation experiments using the subject code is *true prediction*, and comes as close as we are ever likely to come in controlled experiments to mirroring the intended predictive application of the code to the driving stockpile application.

3. Designing experiments maximizes all of the good consequences that result from closely working with a dedicated validation experiment activity, and minimizes all of the bad consequences that result from using previously gathered data.

Engaging a code in the definition and design of validation experiments can easily be distinguished from more common experimental practice. In the conventional case of nonvalidation experiments, the *experimenters* state the following to the computational analyst in the typical approach: "We are performing the following experiment. Tell us what the expected response of gauge **a** will be, given this experimental design and location for the gauge." For a *validation* experiment, on the other hand, the *computational analyst* makes the following statement to the experimenter: "Please perform the following experiment. We recommend locating the gauge **a** at location **x** based on our computational predictions. It will be very useful to see if our predictions agree with your experiment. If your experiment has deviated from our requested design, we still anticipate achieving useful validation consequences through further postexperiment analysis, as long as the design deviation is not great. We don't know what to expect if the deviation is great."

For a variety of reasons it may be desirable to go beyond the design predictions of the calculations to understand the experiment, or to increase the leverage of the acquired data. This is the act of postdictive ***analysis*** of the experiment. For example, if the experiment involved a high-velocity impact at oblique incidence, both the speed and angle of impact might be different in the conducted experiment from what were assumed in the design. Given careful work in the definition and design of the experiment, it is not likely that such experimental variations will destroy the efficacy of the experiment for validation purposes, but new calculations are then required to sharpen the comparisons between calculation and experiment for validation purposes. And, of course, significant further analysis might be required based on the results of comparing the initial computational predictions and experimental results, even if the constraints of the experiments were exactly as assumed. In our experience poor agreement between calculated and experimental data always leads to additional computational study. But the same could be true even if the agreement was excellent; for example, more detailed studies of spatial grid convergence might be desirable in such a case than were performed in the definition and design phases for the experimental effort.

Postexperiment ***analysis*** of validation experiments is the simulation of validation experiments that were in fact conducted, but after the fact. This is the code activity that is most usually associated with conventional experiments. We have good reason to believe that postexperiment analysis will be fruitful for validation if significant efforts have been made to involve the code in the design of the experiment from the beginning, as mentioned above. Postexperiment analysis adds substance to the goal of performing validation experiments that produce data that can be definitively compared with the code calculations. For example, careful analysis should be conducted on complex processing of the experimental data that are used in validation metrics. Calculation-experiment comparisons may also be sufficient to reveal problems in the experimental data or mischaracterization of the experimental facility (Aeschliman and Oberkampf 1998). We always hope that validation experiments provide a benchmark by which code calculations can be judged, but unfortunately we cannot always assume that gathered data are completely correct. Validation experiments defined to satisfy the concepts discussed here represent a balance between dependence and independence of the calculations and the experiment proper, which is at the core of principles III and IV of Aeschliman and Oberkampf quoted above.

Validation experiments should not produce data that fundamentally depend on code calculations for critical data reduction tasks. Such data do not correctly address our need for independent comparability of experimental data with calculations, and violate our need for robustness of experimental data. Experimental data that require code calculations for evaluation can never be a desirable outcome for a validation experiment, although exactly this problem may arise in other kinds of experiments. An example of what we mean is the problem of inference of shock temperature data from experimental shock hydrodynamic data (density, pressure, and velocity fields) using code calculations of shock wave physics rather than some type of diagnostic to directly measure the temperature. The only possible validation data that will emerge from shock hydrodynamics experiments without temperature diagnostics are the shock hydrodynamic data. This problem is relevant, since it arises in investigations of temperature dependence in high-pressure shock-driven material response. Such experiments often need code calculations to estimate the associated thermal conditions under shock loading. For purposes of scientific discovery, this is permissible though speculative. Such experiments, however, cannot be claimed to provide validation data for high-pressure thermomechanical material response because of the lack of independence of calculation and experiment.

The purposeful design of experiments for code-application validation is enabled and accomplished through the team of people that participate in the validation experiment activity. Most obviously, one or more experimenters are participants in this team. However, code users (analysts, designers) must also participate, given our proposed primary role of the code in definition, design, and analysis of validation experiments. Finally, one or more code developers should be on the validation experiment team. Their presence at least provides valuable expert knowledge about the perceived *a priori* capability of the code in all three of its validation experiment roles. In fact, code developers, including experts in the physical models in the code, are the most knowledgeable about the boundary of the acceptable application domain discussed in Figure 6.1. (**DES9**)

We observe that predictive design of validation experiments is an element in the Challenge Problem validation activity advocated by McMillan (1996). Attempting to sharply define the boundary of applicability of the code for a given application domain through a deliberate choice of experiments close to the boundary, but on the invalid side, has greater leverage when code predictions are used to design such experiments. We stress once again that achieving a satisfactory level of performance for the code in comparison with validation experimental data in a case of true prediction has far greater power to raise our level of confidence in the application of the code. We may not be able to quantitatively measure this fundamental observation at this time (although the Validation Metrics Project has a goal of addressing confidence so that this effect *can* be quantified), but it is obviously true. Predictive accuracy is gold, while posttest consistency is merely brass at best, and possibly fool's gold at worst.

All of the principles and methods by which validation experiments were designed, executed, and applied must, of course, be documented. Additional discussion on the nature of this documentation is presented in Section 10.

6.3 What Validation Experiments Are Not

We do not regard validation experiments as ***phenomena discovery experiments***, ***mathematical model development experiments***, ***phenomena exploration experiments***, or ***calibration experiments***. Our reasons follow (Oberkampf 1994, 2000).

Phenomena discovery experiments are experiments conducted to develop a fundamental understanding of some physical process. Examples of phenomena discovery experiments are detailed spatial and temporal measurements of fluid dynamic turbulence, experiments in high-pressure reacting flows, experiments probing the onset of phase changes in materials, experiments probing the microstructural processes underlying crack growth, and experiments designed to explore the phenomenology of fast Z-pinch implosions. As another example, performing an experiment to assess the relative roles of two important physical phenomena in a given application is *not* validation; it is phenomena discovery. It is unlikely that the results of a phenomena discovery experiment will even be comparable to an existing code calculation because it is improbable that a model of the phenomenon will be implemented in the code. And even if a comparison might be made, it is also unlikely that the experimental data will pass the test of robustness discussed above. The code cannot be properly compared with such an experiment until after a model of the phenomenon is carefully implemented, which also destroys any element of code prediction in such an experimental comparison. (**DES10**)

Mathematical model development experiments are experiments expressly conducted to construct a mathematical model of a physical process. For example, either no model exists for the process of interest, or an improvement of an existing model of the process is needed. Commonly, in mathematical model development experiments there have been no code calculations; the code calculations are typically applied after experimental data have been collected to deepen understanding of both experiment and theory, as implemented in code models. At best, the role of code calculations in mathematical model development experiments is to aid in interpreting the experimental data and in synthesizing the results of such experiments with other knowledge, either theoretical or experimental. (**DES10**)

Phenomena exploration experiments are typical of many experiments through the years that are now being used for comparison of calculations and experiments. Some characteristics of phenomena exploration experiments are that they play a role after their execution, there is little or no integration of code calculations in their definition and design, and they have no quantitative estimates of experimental uncertainty. Typically, the description and documentation of data from phenomena exploration experiments reside in published scientific literature. Because these experiments are not dedicated validation experiments, using them for validation purposes has all of the attendant problematic characteristics discussed in Section 4. This said, we also recognize that one is forced to use such experiments when dedicated validation experiment activities are not performed. It is important, however, to avoid the problem of allowing dedicated experimental validation activities to stray in the direction of phenomena exploration experiments or mathematical model development experiments.

Calibration experiments are also not validation experiments. Confusing calibration experiments with validation experiments is a common and insidious problem. The primary goal of a calibration experiment is to produce one or more pieces of data that allow code calibration (or "code tuning") to optimize its agreement with that data, and possibly other data. Calibration was formally defined in the AIAA V&V Guide as ***"The process of adjusting numerical or physical modeling parameters in the computational model for the purpose of improving agreement with experimental data."*** The AIAA guidance is quite explicit in rejecting calibration experiments as validation experiments, and we completely agree. Calibration may still be required in some cases, but calibration is not validation and does not support predictive application of ASCI codes. (**DES11**)

A particular difficulty arises in certain applications of Sandia ASCI codes associated with complex material models. Often, defining such a complex model to be appropriate or even just useful for the intended code application requires experiments that effectively calibrate that material model for the specified experimental range of material response. We do not argue that this process should not be done. There seems to be no way to avoid it in complex situations. Some faith in the accuracy of material response is also a necessary condition for beginning validation activities. None of the experimental design concepts stressed above can be adequately addressed if the most fundamental material models in the subject code are not believed to be appropriate and/or useful. But by themselves such calibration experiments are not validation experiments, not even of the material model itself. The main reason for this is that calibrating material response to a given model does not assess confidence in predictive capability. Assessing confidence in predictive capability requires *at least one additional experiment*, where the material model is used to purposefully design the new experiment so that it differs from the previous experiments in some fundamental way. That subsequent experiment is, indeed, a validation experiment.

6.4 Final Thoughts

A key issue in the design of any validation experiment for the purpose of code application is to balance needs, resources, and required results. There are many constraints that must be balanced when designing and implementing validation experiments. These include the following:

- ***Schedule***—Schedules may not be under our control. There may not be enough time to accomplish the purpose of the required experiments.

- ***Money***—There may not be enough money to accomplish the purpose of the required experiments.

- ***Experimental technology***—It may be difficult or impossible to perform the required experiments or obtain the necessary measurements.

- ***Code maturity***—The code may not be capable of performing the tasks that are required to effectively utilize the experiments.

- ***Failure***—A validation experiment that fails to produce the required data will likely create more work.

- ***Success***—Success is a partnership, defined across a complex community (DP stakeholders, experimenters, code developers, analysts, and the ASCI community).

We are *not* claiming that the only *useful* experiments are "validation experiments." Rather, it is our view that most carefully conducted experiments that are relevant to the intended application of the code are of use at one point or another in the process of assessing the correctness and applicability of a code. Dedicated validation experiments simply place the greatest possible emphasis on "carefully characterized" and "relevant," because they place greater emphasis in the key elements of robust and precise data, computational prediction, and calculation-experiment comparisons. There can be little debate about this view.

To complement code applications to directed stockpile work, as well as to assess the applicability of codes for particular problems, a spectrum of experiments and tests will likely be performed in the future. The results of these experiments should be integrated to produce information required by the validation team. These experiments and tests are

- phenomena exploration experiments

- mathematical model development experiments

- calibration experiments

- dedicated validation experiments

- system and certification tests (including surveillance assessments)

This spectrum of experiments influences the development and growth of confidence in computational models and their applications in an evolutionary sense, as depicted in Figure 6.2. In this figure a serial view of the role of the above experiments is emphasized, as well as the interaction of experiments and the code. In the left column we cast our view of experimental activities in an ordered sequence consisting of phenomena exploration, mathematical model development, calibration, and validation experiments. In reality, of course, there are possibly local loops connecting each of these experimental elements that have not been emphasized. For example, it is apparent that the conduct of experiments designed to be validation experiments could reveal the need to perform experiments from any other element in the column.

Each element in the experimental domain directly interacts with an element in the computational domain. The motivating concepts generating these computational elements are suggested by the arrows in the middle of the figure, which depict the flow of experimental information directly to the computational domain and vice versa. For example, in the case of validation this figure captures the "define—design—analyze" cycle we have suggested for validation experiments. We have simplistically characterized the computational domain through the types of activities expected to be most associated with the directly correlated experimental element. As noted above, there could also be local loops in the computational domain. And, of course, the interaction of the experimental and computational domains could be far more complex than is suggested in Figure 6.2. For example, performing validation calculations could reveal the need for phenomena exploration experiments.

We believe that our depiction in Figure 6.2 highlights the right connections between the experimental and computational domains if the experimental domain is not restricted only to validation experiments as they are defined in this report. In particular, the ultimate goal of these activities is to provide a basis of confidence for the intended application or applications of the code, a goal labeled "Foundation of Credibility" in the figure. The core of this goal, quite simply, is to establish that the code is achieving the correct calculations for the correct reasons when contrasted with the experimental data. We present this as the critical foundation on which ultimate stockpile computing must rest. The separate elements depicted below this foundation refer to subsequent system certification tests that will likely have a significant influence on the credibility assessment of code applications to stockpile problems. We have highlighted the great weight that system tests evoke by emphasizing a more unidirectional flow of information between the experimental and computational domains at this level.

Note that Appendix E presents a brief summary of various types of experiments, some of their key characteristics, and their relationship to validation.

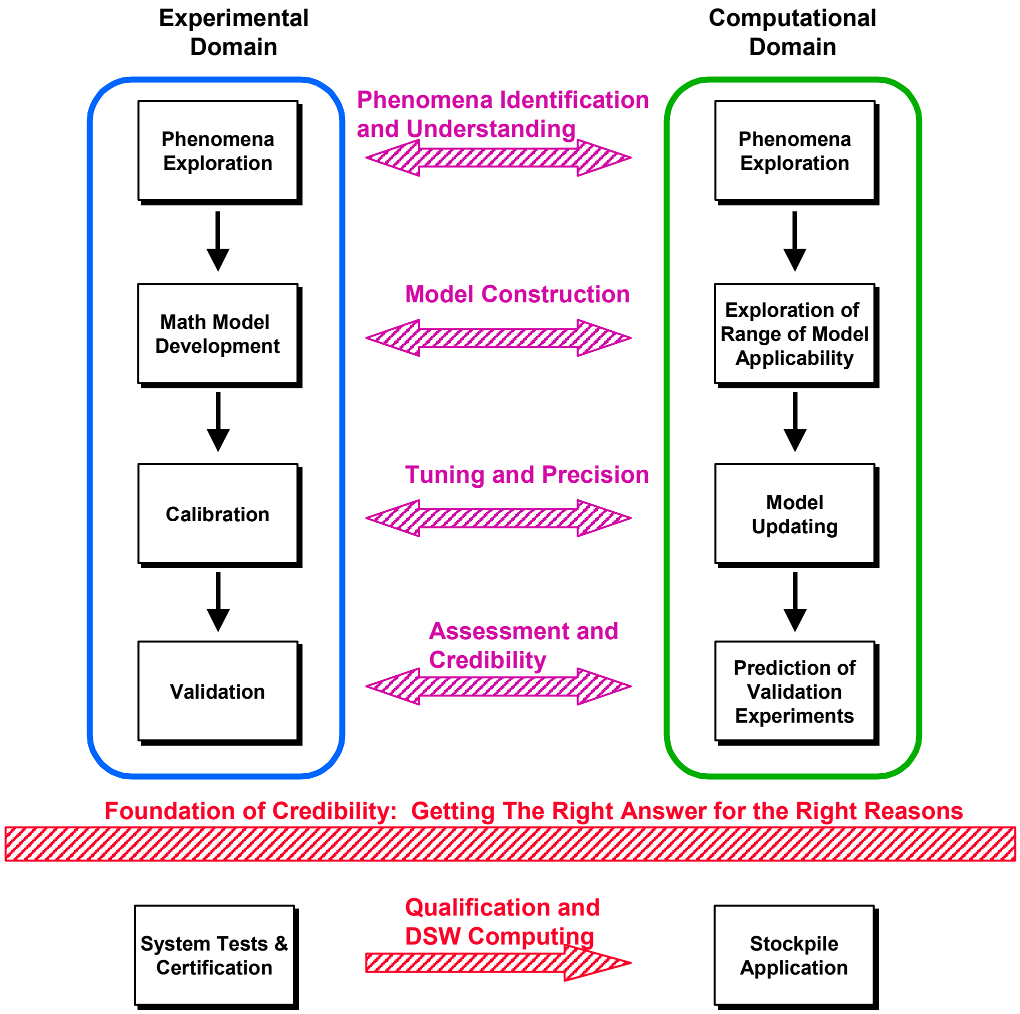

Figure 6.2. The aggregation of experimental activities that contribute to improved credibility of ASCI code applications.

6.5 Experiment Design-Based Concepts

DES1: Validation experiments should be explicitly designed to support assessment of code fidelity and confidence for the intended application through precise and conclusive comparisons of calculations with experimental data.

DES2: The planned validation experiments should specifically address the balance of resources for experiments, code capability, and required predictive confidence for the intended application.

<u>DES3</u>: The region of intended application domain parameters that is covered by the validation experiment activity should be defined in the plan. It should be understood whether the intended application extrapolates the validation domain, interpolates the validation domain, or both.

<u>DES4</u>: One or more experiments should be designed and performed with the goal of resolving the boundary of credibility of the code for the intended application.

<u>DES5</u>: Statistical design of experiments should be applied in the design of the experimental activity.

<u>DES6</u>: Experimental quantification of uncertainty, both variability and bias, should be performed. This should include planned experimental repeats to quantify variability as well as diagnostic fidelity.

<u>DES7</u>: Data resulting from the validation experiment activity and their interpretation should be robust in the sense described in this report. If not, nonrobustness of data should be specifically emphasized in documented outcomes.

<u>DES8</u>: Application of the code to the definition, design, and postexperiment analysis should be performed as part of the experimental activity.

<u>DES9</u>: The validation experiment activity should consist of a team that includes experimenters, code users, and code developers.

<u>DES10</u>: The planned validation experiments should not be phenomena exploration experiments or mathematical model development experiments. If phenomena exploration is required and performed as part of the experimental activity, it should be distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon the phenomena exploration activity should be explicitly defined in the plan and in the experimental outcomes.

<u>DES11</u>: The validation experiments should not be calibration experiments. If calibration is required and performed as part of the experimental activity, it should be clearly distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon an included calibration activity should be explicitly defined in the plan and in the experimental outcomes.

Section 7
Element 5: Metrics

7.1 Description

Element 5, ***metrics***, short for ***validation metrics***, is a term for the most important practice in validation—comparison of the results of code calculations with the results of validation experiments. The straightforward interpretation of this word is that a "metric" is simply a "measure." Thus, choice of one or more metrics defines the means used to ***measure*** the differences between calculations and experiments. Because we emphasize that the overarching goal of validation experiments is to develop sufficient confidence so that the code can be used for its defined application, we do not view the use of metrics as simply a passive means of measuring differences between calculations and experiments. Metrics must be devised that actively resolve assessment of confidence for the intended application of the code. The goal of determining confidence, hopefully quantitatively, introduces great complexity into the task of designing and applying metrics. It also complicates potential constraints that metrics enforce on the definition of validation experiment activities.

The general subject of validation metrics has recently been discussed in Trucano et al. (2001). References and specific examples are found in this paper, as well as in Coleman and Stern (1997), Hills and Trucano (1999, 2001a, 2001b), Oberkampf and Trucano (2000, 2002), Dowding (2001), Easterling (2001a, 2001b), Paez and Urbina (2001) and Urbina and Paez (2001). These references can help the interested reader gain a better understanding of the Sandia V&V program perspective on definition of metrics and their application in some specific instances. It is made clear in Trucano et al. (2001) that the choice of metrics is highly dependent on the subject matter, code, and experiment. In addition, our rigorous work to develop useful methodologies for the definition and application of metrics only began in fiscal year 2001 (Trucano et al. 2001). Only at some future time will we be able to provide more detailed guidance from a perspective as general as that presented in this report. Thus, one should consider the guidance stated here as minimal. We do believe, however, that this guidance is relevant for most validation experiment activities. (**MET1**)

Oberkampf and Trucano (2000, 2002) point out that reasonable validation metrics should generally include the following useful properties, although validation metrics need not be restricted to satisfying *only* these properties:

1. Metrics should incorporate an estimate of the numerical error in the computational simulation.

2. Metrics should not exclude any modeling assumptions or approximations used in the computation of the simulation result. Metrics should reflect *all* uncertainties and errors incurred in the modeling process.

3. Metrics should incorporate estimates of the random errors in the experimental data that are the basis of comparison.

4. Metrics should include estimates of the correlated bias error in the experimental data.

5. Metrics should reflect the level of confidence in the experimental mean that has been estimated. Oberkampf and Trucano (2000) give an example of a metric that provides this level of confidence by including a dependence on the number of experimental replications of the subject experimental data.

6. Metrics should be able to incorporate computational uncertainty that is due to both random uncertainty in experimental parameters required for defining the calculations and any uncertainty that is due to lack of experimental measurement of needed computational quantities. Thus, the metric should utilize nondeterministic methods to propagate uncertainty through the subject computational model.

Clear guidance for dedicated experimental activities naturally follows from the goal of delivering quantitative comparisons between calculations and experiments for the purpose of increasing confidence in application of the code. For example, there is greater confidence in the calculations when they are successfully compared with more precisely defined and accurately measured experimental data than with more poorly defined and measured data. This fact creates experimental guidance: Gather more precisely defined and accurately measured experimental data. This guidance probably seems obvious, except that many system-scale tests and experiments that are not designed specifically for validation purposes *do not* achieve this goal. The goal of gathering precisely defined and accurate data must be built into the design of validation experiments. The success of the definition and application of validation metrics is proportional to success in gathering accurate experimental data.

Similarly, it seems obvious that there is greater confidence in the calculations when they are successfully compared with a larger amount of experimental data than with a smaller amount. The guidance resulting from this observation thus emphasizes the importance of maximizing the amount of (good) data that are gathered on validation experiments. Of course, achieving a greater quantity of high-quality data requires balancing constrained resources, since increasing the quality and number of diagnostics in experiments typically raises the cost. We do not claim that this will be easy to achieve in any given experimental validation effort, but it is certainly required to some degree.

Finally, it is also broadly accepted that there is more confidence when a *predictive* calculation is successfully compared with experimental data than when a *postdictive* calculation is successfully compared. This idea suggests that predicting the results of a planned validation experiment before execution of the experiment is very desirable. This is one reason we stressed in Section 6 the great importance of designing validation experiments with the code under study. Whether estimating the magnitude of the acceptable domain of a code or estimating the robustness of engineering hardware, individuals tend to be overly optimistic.

The most important requirement for validation metrics is that it be possible to compare calculations and experimental data in a meaningful and quantitative way. Trucano et al. (2001) analyzed this issue in some detail, calling this capability ***alignment***. Successful alignment between code simulations and experiments is another major reason that the code to be validated for a specified application be used to define and design the experiments. As discussed in Section 6, the ability to rationally design a validation experiment by using code calculations probably guarantees to the greatest degree possible that the code calculations and resulting experimental data will at least be comparable (nothing in this discussion implies requirements about how accurate this comparison might be). Achieving this level of alignment also rationalizes the typical process of postexperiment analysis, which is given in concepts **DES3** and **DES10** in Section 6.

A major foundation for confidence assessment resulting from validation experiments is to associate success and failure criteria with validation metrics. Ideally, these criteria should be defined before the validation metrics are applied to the collected experimental data. This issue is discussed in greater detail in the next section, but this goal clearly influences the selection of validation metrics.

The importance of quantifying experimental uncertainty has been emphasized in the discussion of experimental design above (**DES8**). Quantifying experimental uncertainty is crucial for validation metrics. (**MET2**) The inference of predictive capability from validation experiments has no hope of proceeding in the absence of experimental uncertainty quantification. It is also important to quantify computational uncertainty. (**MET3**) However, we believe that validation metrics can still proceed even if information about computational uncertainty is lacking or poor. Limited computational uncertainty quantification affects the quality of the information that can be deduced from validation metrics, while lack of experimental uncertainty quantification can completely defeat the purpose of validation. Quantifying experimental uncertainty is a formidable challenge and may well be unattainable in important applications. At the very least, experimental error estimates specifying the diagnostic fidelity of the experiment are needed in any aspect of validation metrics that can be fathomed, including graphical comparisons and analysis of differences between calculations and experiments.

Given quantification of experimental uncertainty, even with limited quantification of computational uncertainty, comparing calculations with experimental data is a technical problem of quantifying uncertainty. Statistical analysis of the differences between computational and experimental results must play a key role in the analysis and presentation of metric data. (**MET4**) A fundamental goal of a validation metric is then to present at least the impact of the uncertainty of the experimental data upon the inferences that are drawn from the validation exercise. It is important to recognize that uncertainty in the experimental data (and of the computations) affects the credibility of the results of a validation metric as well as the formal details of the uncertainty in the comparison, and so strikes at assessment, which is Element 6 of our proposed validation process. This fact, in turn, will affect the impact of the validation activity on confidence in the proposed application of the code.

No matter what form the experimental data take in particular instances, one must ultimately decide whether or not those data lie within the accuracy requirements of the intended application of the code. Stated in another way, the significance of metrics rests squarely upon our determination that the measurements lie within the fidelity requirements of the validation exercise. If this is not the case, it is unlikely that the validation experiment will provide any definitive information pro or con for validation of the code application. These requirements should be addressed to a lesser or greater degree in the specification of the DP application (our Element 1 in Section 2) and in planning of the validation activity (our Element 2 in Section 3). This assessment must account for experimental uncertainty and robustness (**DES8**, **DES9**).

Uncertainty quantification of the calculations that are compared with the experimental data is important. Ideally, this should also be in the form of bias error and variability, analogous with the similar experimental information. Computational bias error is systematic error in the calculation that may be due to physical model errors (validation), numerical errors (verification), or both. Computational variability arises from variability in the specification of the simulated experiment, for example, in variability of experimentally specified initial and boundary data necessary for performing aligned calculations. These issues are discussed in greater detail in Easterling (2001a), Trucano et al. (2001), and Oberkampf and Trucano (2000, 2002).

We expect it may be very difficult to determine computational bias error in given circumstances. For one thing, the results of the validation experiment activity themselves are intended to help assess this error, given an underlying assumption that the numerical calculation is indeed verified.

Specification of computational variability is easier conceptually than specification of computational bias. For example, observed experimental variations can be propagated through ensembles of calculations using uncertainty propagation methodologies, and the computational results can then be analyzed to provide statistical data about the resulting computational variability. This approach is discussed at length in Easterling (2001a) and Trucano et al. (2001). In practice, however, difficulties arise for several reasons. First, the raw effort involved in doing ensembles of calculations to propagate uncertainty becomes very great for complex computations. Second, accurately assembling experimental variability data into precise statistical information for routine computational uncertainty propagation is very difficult in practice. Finally, there are deep questions about whether all of the variability observed in experiments can be properly captured and expressed by uncertainty propagation. The so-called “unknown unknowns” problem lies at the heart of this issue—observed variability in a collection of experimental data may be due to physics issues that are not captured in the computational model because they are unknown.

The complexity involved in the statistical analysis of differences between computational and experimental data for real validation activities and the potential dependence of this complexity upon experimental design details provide further reasons for having expertise in statistical science as an important component of the validation team, as previously recommended (**DES9**). It appears essential to have statistical expertise in performing the exploratory data analysis that is required to fully understand validation metrics when uncertainty is properly accounted for. (**MET5**)

Figure 7.1 illustrates the conceptual increase of quality in performing validation metrics as increased attention is paid to both experimental and computational uncertainty. The figure depicts a desirable progression from (a) virtually qualitative comparisons in which experimental data and calculations are presented side by side on a viewgraph without any information about uncertainties in either one, to (e) analysis of quantitative differences between experiments and calculations accounting for uncertainty in both. Let us consider this figure in some detail.

Figure 7.1(a) is a cartoon of a pure ***viewgraph norm*** comparison, a presentation of compared experimental and computational data that is often seen in practice. This comparison is marked by the qualitative nature of the chosen metric (side-by-side comparison). It is also distinguished by no information on experimental and computational uncertainty. No quantitative statement about validation can seriously be made based on such a comparison, although the statement may provide some level of confidence in the application of the code at an intuitive level. Intuition and degree of agreement between experiment and calculation, of course, is in the eyes of the beholder.

The plot in Figure 7.1(b) suggests a more quantitative comparison between experiment and calculation. While discrete experimental and computational points are suggested in this plot, the concept also encompasses curve overlays, or multidimensional plot overlays of some kind. The key problem with metrics implemented at the level of Figure 7.1(b) is that there is no recognition of uncertainty in the comparison. This plot has no values on the axes because of its conceptual orientation. However, we point out that it is hard to understand what value specific quantities on the axes might provide in the absence of experimental uncertainties. Of course, such plots are usually made because there is an intrinsic comprehension of the efficacy of the experimental data (for example, one actually knows the order of magnitude of the experimental data), or because an explicit representation of some kind of uncertainty is not available.

Figure 7.1(c) suggests that the next logical step for improving the quality of validation metrics is to place conceptual "error bars" around the experimental data. It is at this stage that a validation metric finally makes minimal sense. It is our experience that these error bars typically represent only diagnostic fidelity, not all of the components of uncertainty arising in the given experimental data. Our conclusion is that validation metrics rely *at least* upon experimental data that have diagnostic fidelity characterized and represented in the metrics. As suggested in Section 6, it is desirable to go beyond this minimalist interpretation of experimental uncertainty, however.

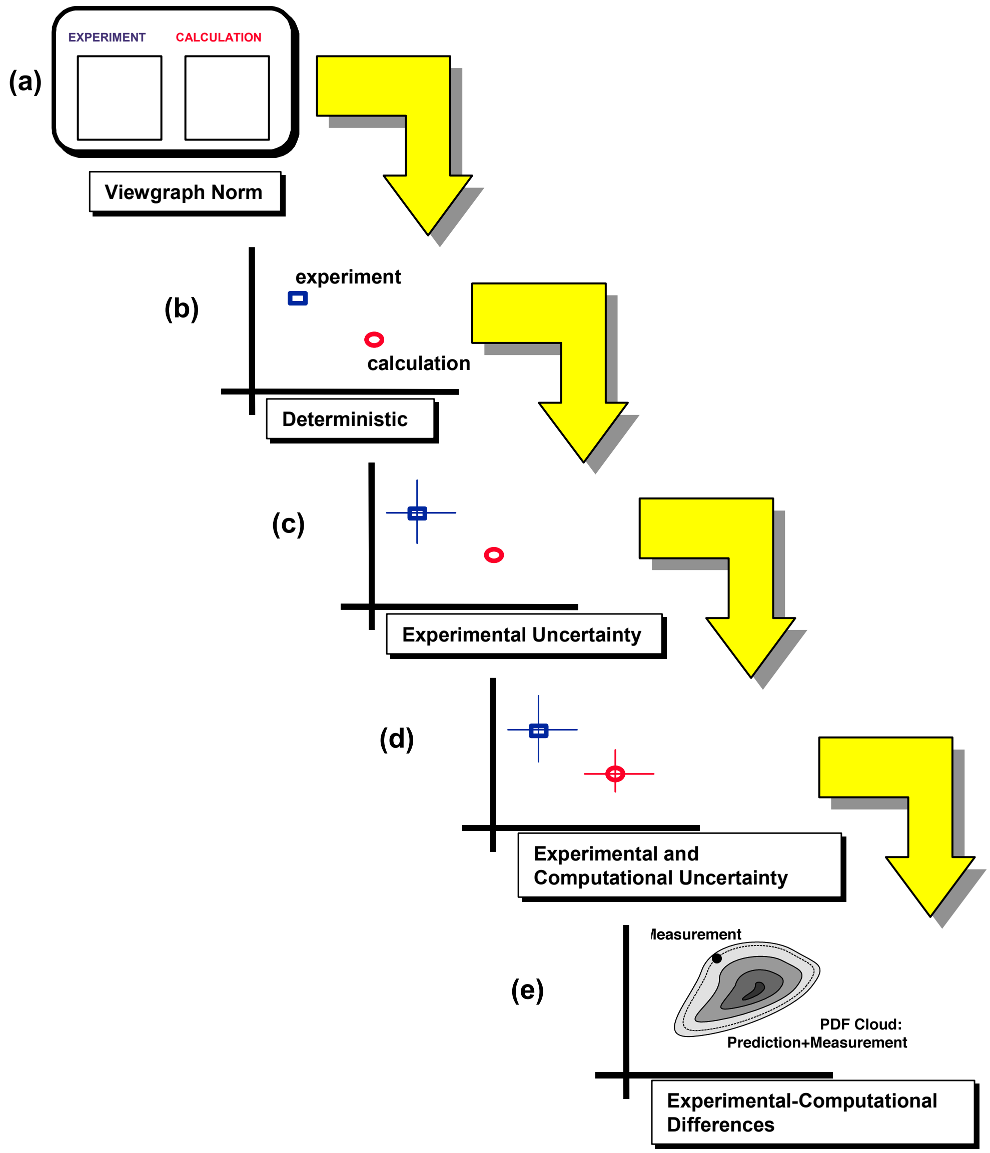

Figure 7.1. A conceptual diagram of increased quality in validation metrics as experimental and computational uncertainties are better characterized.

Figure 7.1(d) represents the case where, in addition to an estimate of experimental uncertainty, there is an estimate of computational uncertainty. For example, an *a posteriori* numerical error estimate from the computation might be given for the specific experimental variable that was measured. The computational error estimate may have come from, for example, a single-grid solution or an estimate based on solutions from multiple grids. An additional feature of Figure 7.1(d) is that the estimated computational uncertainty only deals with numerical bias error, not with computational variability due to the experiment.

Figure 7.1(e) suggests that the next quantitative improvement in validation metrics requires assessment of computational uncertainty, i.e., nondeterministic simulations. Figure 7.1(e) is meant to represent a more complete statistical analysis of the *differences* between the nondeterministic experimental data and the computational data. These differences are now typically a random field, or statistical summaries of underlying random fields, and thus quite complex. This level of statistical comparison implies much larger quantities of data from both the experimental and computational activities. The project described in Trucano et al. (2001) is designed to perform research in defining and applying validation metrics at this level of complexity. The reader is invited to read that report, as well as the other references mentioned above, to develop a better understanding of the technical problems as well as potential benefits of developing and applying validation metrics of this type.

7.2 Metrics-Based Concepts

MET1: All validation information resulting from the validation experiment activity should be based on quantitative comparisons of computational and experimental results.

MET2: Experimental data should account for uncertainty when applied in validation metrics.

MET3: Computations should account for uncertainty when applied in validation metrics.

MET4: Statistical analysis of the quantitative comparison between calculations and experiments should be performed.

MET5: A statistician should be a member of the validation team.

(Page Left Blank)

Section 8
Element 6: Assessment

8.1 Description

The validation team must be able to assess the results of the validation metrics that are applied to quantifying the comparison of calculations and experimental data. The primary goal of such an assessment is to credibly establish whether the agreement of calculations with data from validation experiments satisfies DP requirements. This is the key purpose of dedicated validation experiments, and it is what distinguishes validation from a straightforward computational simulation of an experiment. Failure to assess the validation metric results of validation experiment activities makes it virtually impossible to understand what information about model confidence has been generated as a result of the validation activity.

The performance of assessment strictly depends on definition of the metrics discussed in the previous section and the utility and relevance of the metrics to DP needs. In conventional practice, assessment criteria are typically formulated, if at all, *after* the experimental data have been collected. We believe it is very useful to define some or all such assessment criteria, assuming that this can be done at all, during the DP application specification activity, Element 1, and the planning activity, Element 2. It is recognized that specification of the assessment requirements will be difficult. Importantly, our view of assessment here is strictly local to the validation activity under scrutiny.

Specification of the assessment requirements will be difficult for three reasons. First, DP requirements for the accuracy of computational analyses are commonly very vague. When accuracy requirements are given, they are typically given for relatively high-level system response measures. For example, on an earth penetrator weapon the requirement might be that the penetration depth be predicted to $\pm\,\delta$ meters for a specified target material. Second, if DP gives little or no guidance for detailed system response measures, how should these assessment requirements be determined? For example, on an earth penetrator weapon, how should the accuracy requirement be determined for predicting the maximum lateral forces on all weapon components? Third, how should assessment requirements be determined at the single-physics level (Tier 1) and the low levels of physics coupling (Tier 2) discussed in Section 3? For example, for an earth penetrator application, how should the accuracy requirement be determined for predicting the maximum lateral forces on a metal penetrator impacting a block of man-made ice?

Although the questions and concerns listed above for specifying the assessment requirements are significant, it is our view that the quantification of requirements at multiple levels is the correct approach. When this approach is compared with the traditional test-based engineering approach, it is seen that the approaches are actually analogous. The test-based engineering approach, however, is more ad hoc at the different levels, and there is much more emphasis on testing of the full-scale system.

Two approaches are recommended for guidance in determining the assessment requirements. First, the assessment element is not intended to completely answer the question of whether the code is suitable for the intended application. Clearly, some view of the needed application of the code should influence the criteria for assessing validation metric results. The fact remains that the quality of comparisons of calculations and experiments can also be judged somewhat independently of ultimate application, purely as a scientific subject-matter problem. People who historically engage in computational modeling of experiments make these judgments as a matter of course, but rarely are these judgments quantified or formalized to the degree being advocated for validation purposes in this report.

Second, guidance for assessment requirements for Tier 1 and Tier 2 validation experiments should also be derived from sensitivity and uncertainty analyses done at the full-system level (Tier 3). Stated differently, the impact on Tier 3 of given levels of accuracy at Tiers 1 and 2 should be determined with sensitivity and uncertainty analyses. These analyses will probably need to be done in the DP application and planning elements, Elements 1 and 2, respectively. Some of the issues involved in uncertainty estimation are discussed in Section 9.

Because of the imprecision involved in specifying the assessment requirements, success or failure in meeting such requirements should not be viewed as a "go–no go" assessment. There is, of course, a wide spectrum of possibilities in computational-experimental comparisons between success and failure. In this discussion, however, the two extremes of "success" and "failure" are emphasized because it is important to assess the results of validation metrics as rigidly as possible. We recognize that a statement of success or failure in a metric comparison may seem to presume that the experimental data used in the validation metric are accurate and that all of the failure in the comparison resides, therefore, with the computational model. This need not be the case, as illustrated in Oberkampf and Aeschliman (1992). Those authors found a problem in experimental data, rather than the computation, as a result of the application of careful and forceful validation metrics. Ultimately, the experimental design element discussed in Section 6 must lead to experimental results that have the highest probability of functioning as appropriate benchmarks for the assessment of computational results. Assessment is tightly dependent upon the quality of experimental results.

Using this language, let us focus on quantitative ***success criteria*** for the moment. Success criteria are statements of how close metric comparisons of calculations and experiments must be to claim that a given calculation or set of calculations agrees with the experiment to a specified level of confidence. Success criteria should be defined in correspondence to defined validation metrics. Successful comparison of calculations and experiments may also be determined by demonstrating that a chosen metric is smaller than a defined quantitative threshold, which may be larger than experimental error bars or a more general functional of those error bars. Such a threshold might be chosen from the underlying application, from requirements based on the subject matter, or from additional expert judgment by the validation team. A success criterion could also be more complex than a simple threshold and require such elements as multiple weighted validation metrics, weighted experimental data, subjective judgment (accounting for epistemic uncertainty), probabilistic or statistical criteria (accounting for aleatory uncertainty), and other complexities. (**SF1**)

Making validation-metric success criteria as quantitative as possible is important. It is possible that success criteria may be qualitatively or indirectly defined, perhaps through a process of assimilated experience during the course of the validation activity. In our opinion, however, qualitative criteria, or criteria that are developed after the activity of calculation and experiment comparison has begun, make the validation job more difficult and undermine the credibility of the assessment. In addition, qualitative criteria developed after the execution of validation experiment activities lead to the danger of concentrating the definition of success on discovering any feature in the comparison that is "good," at the expense of answering the harder question: Is there an important reason that this feature in the computational-experimental comparison should be good?

Attaching quantitative success criteria to the definitions of validation metrics eliminates ambiguity from the meaning of computational-experimental comparisons. The net effect of this statement is that a validation metric could also be viewed as a *pair* of concepts—one being the definition of the metric, the other being a definition of what it means for the resulting comparison to be "good" or "successful." Defining the metric and its associated success criteria may be quite difficult in various circumstances, but this goal is very desirable, nonetheless, and should be attempted in all validation activities.

Success criteria may be highly *nonunique*, meaning that two different people may define two distinctly different success criterions for a given validation metric. For example, one person may believe that computations should pass through all experimental error bars, while another may believe that only a subset of the data have this requirement. Contradictions resulting from the nonuniqueness of success criteria, of course, have to be dealt with at the level of subject-matter expertise or, ultimately, by applying knowledge about the underlying application requirements. Nonuniqueness does not diminish our desire or capability to formulate rational and communicable criteria.

We believe it also is important to define quantitative ***failure criteria*** for validation metrics, ideally at the time that the metrics are defined. (**SF2**) This concept is aimed primarily at the situation in which success criteria are not defined. We have found that it may be very difficult to define success, while a clearer notion of failure may be available. If such is the case, we encourage that at least failure criteria be stated and coupled with the corresponding validation metric. An unsuccessful comparison of calculations and experiments may be measured by demonstrating that a chosen metric is larger than a defined quantitative threshold. For example, while one may not be able to categorically define a threshold for an applied validation metric that defines success, one might be able to categorically state that that the model has failed if the calculation is greater than, say, 100% from the data in a point-wise comparison. A failure criterion could also be more complex than a simple threshold, just as in the case of success criteria, and require such elements as multiple weighted metrics, weighted data, subjective judgment, probabilistic or statistical criteria, or other complexities.

If success criteria have been defined, it may seem logical to define failure as not achieving the success criteria. This conclusion is especially attractive if the success criteria have been defined using simple thresholds for metric comparisons. But as the success criteria become more complex, and especially as they may involve probabilistic inference or epistemic uncertainty, this conclusion may be too strong to draw in many cases. The logic of how we properly group success and failure of validation metrics in a complex validation activity is not simple and may not lend itself to either/or conclusions. For example, it is obvious that the opposite of a failure criterion does not necessarily define success, either. This is one reason we emphasized above that there may be a range, or interval value, between "success" and "failure." Experience with real computations, real experiments, real validation metrics, and real assessment criteria will expand our understanding of this issue in the future. In our opinion, failure criteria are likely to be an important complement to success criteria for validation metrics.

8.2 Assessment-Based Concepts

SF1: Success criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

SF2: Failure criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

Section 9
Element 7: Prediction and Credibility

9.1 Description

The influence of the intended application of the code upon the validation activity is expressed in our methodology through Element 1, the definition of the required DP application, and through Element 2, the detailed plan of the validation activity. Element 7, prediction and credibility, completes the dependence of the experimental validation activity on the intended application of the code. The primary purpose of this element is to ensure that the question of ***credibility of the code for the intended application*** is asked at the completion of the specific validation activity or activities and that some attempt is made to answer it. (**PRE1**) The desirable outcome of this element is to quantitatively determine the contribution of the validation activity to our understanding of code credibility, especially through the results of the metrics and assessment elements.

We emphasize prediction in the definition of this element because, as stated before, the primary type of code application that we are focusing our methodology on is high-consequence prediction. Our expectation is that the underlying application of the code specified in Element 1 represents a region of the application domain that is different from the region where validation has been performed, and thus requires prediction from the validation database. As was stressed in Element 4 (Section 6), making predictions for validation experiments that *will be* conducted is the most powerful contributor to building predictive credibility.

We believe it is useful to have a sharp concept of a desirable approach to predictive application of a code. Pilch et al. (2000a) discussed such a concept in the context of developing verification and validation (V&V) plans. Those authors argued that V&V should aim to support a canonical concept of prediction expressed as

Best Estimate Plus Uncertainty (BE+U)

Stated simply, this concept argues that important modeling results should be presented as a best estimate of the "real" answer, such as provided by an experiment or by the most accurate possible code calculation, coupled with a quantification of the uncertainty in that estimate. The goal of using carefully structured V&V activities to develop credible **BE+U** for required applications is well understood in the context of assessing nuclear power safety, for example, as discussed in Boyack (1990). The question posed by the prediction and credibility element is then, How do the results and consequences of the subject validation activity influence the credibility of the intended **BE+U**? Our view is that validation should provide information for both the best estimate and the corresponding estimate of uncertainty. Of course, the greater the consequence of the intended application, the greater the importance of having sufficient levels of credibility associated with **BE+U**, which is why we have chosen to couple these concepts in this element.

Figure 9.1 presents four idealized tasks that contribute to the delivery of **BE+U** in a generic sense. The term "idealized" is used for at least two reasons. First, our specification and brief description of the four key tasks is very simplified and hides a multitude of technical questions surrounding even partial completion of such tasks. Many of the detailed mathematical procedures needed to quantify each of these tasks are presently research topics, especially for Task 3 (application credibility) and Task 4 (forecast uncertainty). Second, the diagram does not represent all the ways in which these tasks may be coupled, rather than conducted sequentially. Nonetheless, Figure 9.1 is a valuable means of illustrating how the results of the validation methodology presented in this report logically couple to the development and presentation of credible predictions for the required applications.

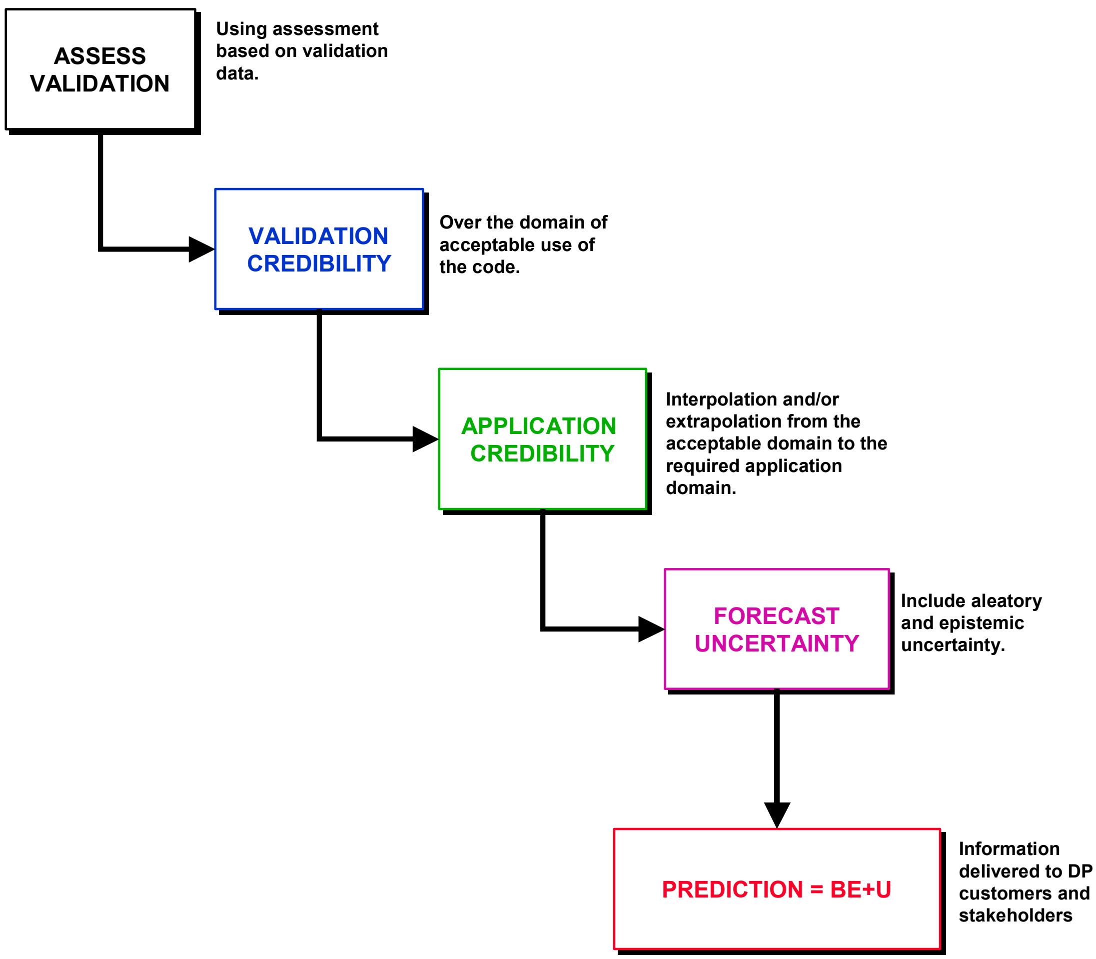

Figure 9.1. Tasks that lead to the development of a credible "Best Estimate + Uncertainty" for the required application.

The tasks in Figure 9.1 are briefly discussed below, as a detailed discussion of these topics is beyond the scope of this document.

1. **Assess Validation**

   This task is the aggregation of all of the information created in the validation methodology through the execution of the assessment element discussed in Section 8. We stress that the assessment information resulting from the validation activity should include cases of both “success” and “failure” in comparison with validation experiments.

2. **Validation Credibility**

   This task attempts to draw conclusions regarding the credibility of the code over the entire domain in which the code has been judged “acceptable.” Put another way, this task involves using the assessment information and conclusions of Task 1, plus knowledge gained from applications of the code for which there were no validation data available. These data were referred to as the “A” in Figure. 6.1. Thus, determining credibility requires balancing the positive and negative results included in Task 1, as well as the positive and negative results obtained from applications of the code.

3. **Credibility for the Application**

   This task projects the determination of credibility that results from Task 2 to the intended application domain. A separate determination of “application credibility” is not needed if the application domain is essentially the same as the validation domain. Unfortunately, it is highly unlikely that this is the case. Thus, developing application credibility will require interpolation or extrapolation of the validation credibility conclusions, or both. Ideally, progress on the application credibility task most directly supports development of the “Best Estimate” component of **BE+U**. The interpolation or extrapolation from the validation database that is required to characterize credibility for the best estimate in an application should be strictly physics based for maximal credibility. An example of the best estimate that is not strictly physics based is the use of an experimentally determined correction factor, or bias correction, in the extrapolation procedure. Thus, our desire to rely upon our validation methodology to support this extension beyond the validation database is a consistent goal from this perspective.

4. **Forecast Uncertainty**

   This task directly addresses the “Uncertainty” component of **BE+U**. The information developed in Task 3 of necessity has associated uncertainty, and it is the purpose of Task 4 to quantify this uncertainty. (In Figure 9.1 we suggest that this uncertainty may be aleatory, amenable to probabilistic description, or epistemic, requiring more than a straightforward probabilistic quantification.) Some of this total uncertainty directly originates in the previous tasks, where, for example, uncertainty is a fundamental component in Task 1, validation assessment. A second source of uncertainty is due to the interpolation and/or extrapolation from the acceptable use domain to the application domain. A third source of uncertainty could be due to the application domain itself, thus adding further complexity to this task. We note that if the "Best Estimate" component is simply a deterministic prediction, then the uncertainty we are discussing here could be a probabilistic description of the variability of the best estimate in response to characterized uncertainties in the application domain or validation information. However, even if the "Best Estimate" is a probabilistic statement, this task expresses the need to quantify uncertainty associated with this probabilistic statement. The need to quantify uncertainty in probabilistic forecasts is well recognized in many fields. A sample of modern texts on this topic consists of Ayyub (2001), Bedford and Cooke (2001), Haimes (1998), Haldar and Mahadevan (2000), Kumanoto and Henley (1996), and Melchers (2001).

9.2 Prediction-Based Concepts

PRE1: The contribution of the validation experiment activity to quantitatively understanding credibility of the code for the intended application should be characterized.

Section 10
Element 8: Documentation

10.1 Description

It is necessary to ***document*** the content and results of any validation experiment project. Validation is the development of evidence that the code can be successfully applied to the driver application with quantifiable confidence. As such, validation documentation is a necessary repository of this evidence. To support the DOE Stockpile Stewardship Program, it is also important to preserve this information. Such preservation requires suitable ***archiving***, which places requirements that influence the form of the documentation. Finally, it is important to ***disseminate*** this documented evidence, both within Sandia and to the larger DOE DP community, both in the near term and in the long term. Dissemination requirements also influence the form of validation documentation.

At Sandia, a records management system for the ASCI V&V program is being developed. In the long term, this system will provide an appropriate vehicle for archival and dissemination requirements. Thus, we expect that the requirements for information to be placed within this system will be requirements about the generation of validation documentation. It is our opinion that validation documentation should be compatible with any requirements that are necessary for using the records management system. The comments that follow are limited to general guidance on the content of appropriate validation experiment documentation. (**DOC1**)

Validation documentation associated with particular validation experiments is part of a larger framework of documented evidence on the verification and validation (V&V) of the code for its intended applications. Validation documentation may rely upon and point to other documented information associated with the V&V activities that intersect the validation experiment activity that is the focus of the documentation we emphasize here, such as V&V plans, code-specific software quality documents, and verification documents. Any documentation developed for a specific validation activity is also likely to be needed in subsequent validation efforts. Validation documentation should be written in such a way that it is integrated into the overall V&V documentation tree of the code. Existing documentation should be of high enough quality that it can be relied upon by the validation team during the planning and execution of a given validation activity. (**DOC2**)

The validation documentation should contain information about all of the relevant work of the validation experiment activity. This includes, but is not necessarily restricted to the following: (**DOC3**)

1. Information about the code application that is targeted by the validation experiment activity, the origin of this application in DP requirements, and the particular modeling requirements that this application creates. Of special interest is information about stockpile decision processes that will utilize specific code

calculation results, as well as characterization of the uncertainties that are associated with the stated applications of the code. (See Section 2.)

2. A detailed discussion of the relationship of the validation experiment activity to the PIRT for the code application. This includes an assessment of the importance and priority of the activity and a detailed discussion of the physical phenomena in the PIRT that are being validated by the activity. (See Section 3.)

3. A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity. This focuses on two specific areas: (1) the assessed capability of the code to successfully contribute to the experimental design process and (2) the details of how specific validation calculations were verified during the course of the validation experiment activity. (See Section 4.)

4. Documentation of how the code was used in the definition, design, and analysis of each validation experiment. Enough information should be included to allow repetition of the described calculations by others. This latter point is essential for the calculations used in the specified metrics, as well as for success and failure information. Information that is typically necessary for enabling new computation of old calculations at some point in the future includes mesh construction information, calculation geometry, computational initial and boundary conditions, computational model inputs such as material-model input specifications, and selection of computational algorithm parameters such as iterative tolerances and numerical smoothing parameters. Typically the easiest way to provide this information is to insert input-file echoes into the documentation, but one does not necessarily need to document every input file used. For example, this need could be addressed by presenting one input echo for the preprocessing, code calculation, and postprocessing (if any) associated with a specific calculation, and a table of varied parameters if more than one calculation is presented. Input files from multiple code calculations do not need to be echoed in the documentation unless they represent differences that cannot be easily summarized in tables. (See Section 6.)

5. A complete description of each experiment, sufficient to allow experimental replication in the future. It is important to discuss why the experiments are neither calibration nor phenomena exploration experiments. If this is not true in one or more cases, discussion of the proper role of calibration and/or phenomena exploration in the validation experiment activity is essential. (See Section 6.)

6. A description of the analysis of experimental data. It is important to document information about the uncertainty in the acquired experimental data, including estimation of both random and bias errors. This should include information on the method for estimating random error, how replication experiments were executed, test unit to unit variability, the method for estimating bias errors, diagnostic precision, the quantitative meaning of error bars, and any other information that

helps to clarify the experimental uncertainty required for validation metrics. It is also important to describe in detail any data analysis procedures that were applied to produce the data that were compared with calculations in the validation metrics element. Robustness of the experimental data with respect to the application of selection or analysis criteria should also be addressed. (See Section 7.)

7. The methods and results of the validation metrics applied in the validation experiment activity. This includes graphical and other comparisons between calculations and experiments, characterization of computational uncertainty that contributes to these comparisons, and statistical analysis of the comparisons between calculations. Defined success and failure criteria for these metrics should be documented. The raw data that are used in the metrics must be in a form that is compatible with the archival and dissemination requirements of the V&V program's records management system. For example, the data that are used to make a particular plot in the documentation need to be archived. Because it is unlikely that all of the data can be included within the document, the data should be archived in electronic form, and the information necessary to describe the archiving and retrieval of those data should be spelled out in the validation documentation. (See Section 7 and Section 8.)

8. A characterization of the credibility of the code for the specified application, based on the results from the application of the defined validation metrics and the assessment. This discussion should define credibility and explain how it is revealed by the validation metrics and the results of the assessment element. The implications of this characterization should be related back to the underlying application driver, with particular attention devoted to the regions of the application domain that are associated with success and failure for the specified metrics. If the level of credibility engendered in the code application for this particular validation experiment activity is sufficient for the intended application, this point should be specifically discussed. If the level of credibility is judged to be inadequate, there should be a discussion that addresses necessary model improvements, as well as possible subsequent validation activities that could raise the level of credibility. In the worst possible case, the experimental activity might reveal that the code as currently assessed is inappropriate for use on the intended application. The reasons for such a conclusion must be made clear in the documentation. (See Sections 4, 7, 8, and 9.)

9. Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application. A discussion should be given for the sources of uncertainty that were considered, as well as those sources that were neglected. Assumptions should be discussed concerning any probabilistic analyses, particularly those involving epistemic uncertainty. (See Section 9.)

Given the scope of this information, it should be evident that one or more documents, such as SAND Reports, will likely be required to achieve all of our proposed content. It could well be that a series of reports is required, such as

- Code Verification
- Validation Experiment Design
- Validation Experiments and Data Analysis
- Validation Assessment
- Predictive Credibility Assessment

While it may be fully appropriate to break the needed content into multiple documents, individual documents should each be compatible with the overall guidance for integration in the V&V documentation tree, for archival, and for dissemination.

10.2 Documentation-Based Concepts

DOC1: Validation experiment activity documentation should be compatible with the Sandia V&V program's records management system.

DOC2: Validation experiment activity documentation should be integrated into the associated V&V documentation tree for the code and its applications.

DOC3: Validation experiment activity documentation should contain information on the following topics:

- Information on the application and requirements that are driving the validation experiment activity, including references to information DP documentation and the overall V&V plan that the activity is part of.
- Specific discussion on the associated PIRT and how the validation experiment activity is located in it.
- A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity.
- A comprehensive discussion of the definition, design, and analysis of each validation experiment via use of the code.
- A complete description of each experiment, sufficient to allow experimental repeats in the future.
- A description of the analysis of experimental data, including quantification of uncertainty in the acquired experimental data.
- The methods and results of the validation metrics applied in the validation experiment activity, including the defined success and failure criteria for these metrics.

- An assessment of confidence in the code application resulting from the application of the defined metrics, the results, and their performance versus the defined success and failure criteria.

- Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application.

(Page Left Blank)

Section 11
Appraising Validation Methodology

11.1 Introduction

The concepts developed in the above sections of this report should be useful for several reasons. Primarily, they describe what we believe to be fundamental content in any experimental activity that is designed to be a validation experiment. We believe this is especially true for experimental validation of codes that are intended for high-consequence applications such as DP applications at Sandia. In that context, we believe that these concepts and their associated description in this report provide a useful basis of appraisal for a validation activity that is struggling with the problem of how to define dedicated validation experiments. Such appraisal could be applied both to specific validation experiment activities and to a general collection of such activities associated with the overall validation plan for the targeted code application.

An accumulation of such appraisals could also suggest ways for the Sandia V&V program to facilitate improved validation activities in the future. For example, the Sandia V&V program identifies formal peer review as an important quality process for the execution and management of the V&V program (Pilch et al. 2000b). The concepts and methodology discussed in this report could be very useful when applied in peer review assessment of implemented validation activities, especially the Level 2 peer review defined in Pilch et al. (2000b). However, the concepts we have proposed in this document that could be used for appraisal of validation activities are at an early stage of development. We expect that during fiscal year 2002 these principles will be discussed and debated by the larger ASCI/DP community here at Sandia.

This section briefly summarizes the concepts we have developed for experimental validation. The summary is followed by a simple scoring system we devised that can be applied to these concepts for appraising validation activities. We do not claim that our suggested scoring system represents the most desirable or effective appraisal method. However, this system should suffice for estimating the quality of proposed and realized validation activities. Evidence of how useful this appraisal method can be is demonstrated in appendices A, B, and C, where three separate experimental activities are scored according to our recommendations. Conclusions drawn from the exercise of scoring these experimental activities are discussed in these appendices.

11.2 Measurement

11.2.1 Summary of Concepts

Below, we have summarized the specific concepts that are developed and discussed in Sections 2 through 10. These concepts form a checklist that can be used to appraise proposed or realized validation activities. Note that the specified concepts are also located in the body of text where they are first discussed and summarized at the end of each section of the text in which they first appear.

APP1: The validation experiment activity should be derived from the intended code application defined in an existing code-application V&V plan.

PLAN1: The dedicated validation experiment activity should be part of a hierarchical validation activity that is defined by a PIRT. The planned validation experiments should then be well correlated with specific PIRT elements, and those elements should be clearly identified in the experimental plan.

PLAN2: Information relevant to defining success and failure for comparison of code calculations with the results of experiments is identified in the PIRT.

PLAN3: The dedicated validation experiment activity should be defined in terms of the recommended Tier 1 through Tier 3 complexity structure if this is not explicit in the existing PIRT.

PLAN4: The validation experiments themselves should be defined in a formal documented plan.

EED1: All applicable concepts in this report should be applied to guiding the use of existing experimental data in experimental validation activities.

VER1: The code verification status should be understood by the validation analyst and documented and determined to be adequate for the pursuit of an associated validation experiment activity.

VER2: The existing VERTS for the code should contain elements that are believed to be in alignment with the associated validation experiment activity. Calculation verification should be performed and documented for these specific VERTS elements.

VER3: New VERTS elements should be defined if there is inadequate coverage in the code VERTS to contribute to assessing code verification status for the planned validation experiment activity. The calculation verification of these new elements should be performed and documented.

VER4: A calculation verification strategy (typically centered on convergence studies and *a posteriori* error estimation) should be defined for the calculations performed in the validation activity.

VER5: All necessary information required for the verification assessment for the validation experiment activity should be documented.

DES1: Validation experiments should be explicitly designed to support assessment of code fidelity and confidence for the intended application through precise and conclusive comparisons of calculations with experimental data.

DES2: The planned validation experiments should specifically address the balance of resources for experiments, code capability, and required predictive confidence for the intended application.

DES3: The region of intended application domain parameters that is covered by the validation experiment activity should be defined in the plan. It should be understood whether the intended application extrapolates the validation domain, interpolates the validation domain, or both.

DES4: One or more experiments should be designed and performed with the goal of resolving the boundary of credibility of the code for the intended application.

DES5: Statistical design of experiments should be applied in the design of the experimental activity.

DES6: Experimental quantification of uncertainty, both variability and bias, should be performed. This should include planned experimental repeats to quantify variability as well as diagnostic fidelity.

DES7: Data resulting from the validation experiment activity and their interpretation should be robust in the sense described in this report. If not, nonrobustness of data should be specifically emphasized in documented outcomes.

DES8: Application of the code to the definition, design, and postexperiment analysis should be performed as part of the experimental activity.

DES9: The validation experiment activity should consist of a team that includes experimenters, code users, and code developers.

DES10: The planned validation experiments should not be phenomena exploration experiments or mathematical model development experiments. If phenomena exploration is required and performed as part of the experimental activity, it should be distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon the phenomena exploration activity should be explicitly defined in the plan and in the experimental outcomes.

DES11: The validation experiments should not be calibration experiments. If calibration is required and performed as part of the experimental activity, it should be clearly distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon an included calibration activity should be explicitly defined in the plan and in the experimental outcomes.

MET1: Statistical analysis of the quantitative comparison between calculations and experiments should be performed.

MET2: Experimental data should account for uncertainty when applied in validation metrics.

MET3: Computations should account for uncertainty when applied in validation metrics.

MET4: All validation information resulting from the validation experiment activity should be based on quantitative comparisons of computational and experimental results.

MET5: A statistician should be a member of the validation team.

SF1: Success criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

SF2: Failure criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

PRE1: The contribution of the validation experiment activity to understanding credibility of the code for the intended application should be characterized.

DOC1: Validation experiment activity documentation should be compatible with the Sandia V&V program's records management system.

DOC2: Validation experiment activity documentation should be integrated into the associated V&V documentation tree for the code and its applications.

DOC3: Validation experiment activity documentation should contain information on the following topics:

- Information on the application and requirements that are driving the validation experiment activity, including references to information DP documentation and the overall V&V plan that the activity is part of.
- Specific discussion on the associated PIRT and how the validation experiment activity is located in it.
- A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity.
- A comprehensive discussion of the definition, design, and analysis of each validation experiment via use of the code.
- A complete description of each experiment, sufficient to allow experimental repeats in the future.
- Description of the analysis of experimental data, including quantification of uncertainty in the acquired experimental data.
- The methods and results of the validation metrics applied in the validation experiment activity, including the defined success and failure criteria for these metrics.
- An assessment of confidence in the code application resulting from the application of the defined metrics, the results, and their performance versus the defined success and failure criteria.
- Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application.

11.2.2 Scoring System

We recommend that the checklist of concepts in Section 11.2.1 represent the focus of comparison for a planned or executed experimental activity for purposes of appraisal. Then, as a means of self-assessment each concept listed should be scored with a 0 to 3 evaluation criterion. The recommended values are defined as follows:

0 Not addressed in the validation activity

1 Poorly addressed, but present in the validation activity

2 Well addressed, but some improvement is identified for the validation activity

3 Well addressed and no improvement is identified for the validation activity

We recommend that a cumulative score not be calculated. A cumulative score for the entire set of concepts makes little sense given the level of information provided in this document. We do recommend identifying strengths and weaknesses of the validation activity by examining the overall pattern of scores for the aggregated concepts. This is the approach taken for each example in appendices A, B, and C. Aggregate scoring for the concepts is represented by a histogram in each appendix.

(Page Left Blank)

Section 12
Summary

In this report we have defined and described concepts of experimental validation and their role in a general methodology for application of ASCI computational engineering and science codes. The concepts are at an early stage of development, and we anticipate that our thinking will evolve as we apply the proposed methodology. The material in this report has been organized to reflect critical elements in a process that starts with the definition of application requirements for stockpile computational modeling and ends with the delivery of the necessary computational results. This process, outlined in Figure 1.1, has been the major underlying theme in this report. Each of the important elements in that figure—specified application; planning; code and solution verification; experimental design, execution, and analysis; validation metrics; assessment; predication and credibility; and documentation—creates useful concepts for the planning and execution of experimental validation projects.

We have emphasized that our concepts can be fruitfully applied for appraising validation activities, either during the planning phases of validation experiments or as a means of determining the impact of the implemented experiments. Three appendices demonstrate the application of the stated concepts and a suggested method of appraisal for three different experimental activities.

Some important features of our discussion are summarized in the following list, which presents three characteristics of concepts for validation activities. First, we state an ideal goal for a validation activity. Next, we emphasize the potential for possible limitations upon that stated ideal. Finally, we give a practical compromise that we believe is representative of the way progress will be made in more realistic validation activities. This list is intended to be cautionary, not discouraging, and points out that the realities of budget and schedule are always the most important constraints in planned validation activities.

1. **The Ideal:** Calculation and experiment are defined to allow the greatest degree of alignment for purposes of computational-experimental comparisons.

   The Possible: The quality of the computational-experimental alignment can mainly be assessed only during the execution of the experimental activity.

   The Practical: Focus on where the alignment between the given validation experiment and the code is believed to be good, and seek to improve this alignment as the activity progresses.

2. **The Ideal:** The calculations run cleanly and without problems, allowing efficient, clear, and conclusive comparisons with experimental data.

The Possible: The calculations are difficult to run and pose significant difficulties for quantitative comparison with experimental data.

The Practical: Restrict—***don't eliminate!***—validation activities until the associated code is demonstrably robust and verified and capable of generating results that are worthy of high-quality comparisons with experimental data.

3. **The Ideal:** The experimental data are well characterized, and a large number of experimental realizations are available for estimating random and bias errors.

   The Possible: The experimental data are moderately characterized, and a small number of experimental realizations are obtained.

   The Practical: Focus on critical experimental data and target fewer experimental data of higher quality.

4. **The Ideal:** The validation activity is part of a well-defined and documented validation plan.

   The Possible: The validation activity is part of a validation plan that has ill-defined components and some ad hoc characteristics.

   The Practical: Emphasize improving the planning and coordination framework for the ongoing and future validation activity.

5. **The Ideal:** The code is used to define, design, and analyze the validation experiments.

   The Possible: The code is only applied in some of the definition, design, and analysis of the experiment, but most of the code results are used in postexperiment analysis.

   The Practical: If the problem is due to inadequate code capability, slow down the pace of experimental activities until the code can play a more active role in the definition and design of validation experiments. If the problem is due to imbalanced priorities between computational and experimental workloads, place a higher priority on performing some definition and design of experiments with the code.

6. **The Ideal:** There is sufficient time and money for detailed documentation of the validation activity.

   The Possible: Few resources and little time are available for documentation.

   The Practical: Realize that undocumented validation work is essentially useless for the goals of ASCI and the Stockpile Stewardship Program. Reduce the overall calculation and experimental work to increase the level of effort and money devoted to documentation. Seek to reduce the number of new validation activities until old ones have been sufficiently documented.

7. **The Ideal:** There are sufficient personnel and the right personnel for a fully coupled computational-experimental validation activity.

   The Possible: There are insufficient personnel, and they may be unbalanced to the experimental side.

   The Practical: Reduce the scope of the experimental activity until better coupling of the computational and experimental work can be achieved.

8. **The Ideal:** The validation activity is strongly decoupled from hardware qualification activities.

   The Possible: The validation assessment is part of a hardware qualification activity.

   The Practical: Emphasize those features of the hardware qualification that are most compatible with validation.

9. **The Ideal:** The physics embedded in the validation activity is understood.

   The Possible: The validation activity is used to explore physics issues.

   The Practical: Emphasize where the physics is best understood for validation. Precisely define where the physics is insufficiently understood for validation so that experimental scientific-discovery activities can begin to be defined and separated from more focused validation experiments in the future.

10. **The Ideal:** Lessons learned from the validation activity are clearly formulated, documented, and broadly communicated.

    The Possible: Embarrassments are typically not publicized, let alone documented.

    The Practical: Lessons learned are of vital importance to the overall validation effort for a code application. Stress that lessons learned are information, not blame apportionment. Publicize a subset of the lessons learned that is not all positive, even if it is impossible for political reasons to publicize all of the problems that arose in the conduct of the validation activity.

We have attempted to argue the need for implementing as many of the concepts discussed in this document as possible in a robust validation activity. Nonetheless, it is clear that there is room for graded approaches within our proposed methodology. We recognize that it is unlikely that full implementation of all of the concepts will be an option in some or most validation activities. As demonstrated in the above list, compromises can be made and still build confidence in computational modeling and simulation.

We believe that appraising a validation activity, whether planned or already executed, with respect to the concepts described in this report increases the understanding and value of that activity. Examples of such assessments have been included in the appendices to illustrate exactly this point. It is our hope that those who engage in future validation activities will find these concepts to be useful in planning, as well as in scrutinizing the results of the experimental activity for future improvement.

References

1. Aeschliman, D. P., and W. L. Oberkampf (1998). “Experimental Methodology for Computational Fluid Dynamics Code Validation.” *AIAA Journal* 36, no. 5: 733–741.

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

3. Aragon, K., J. Zepper, K. Byle, D. Eaton, and M. Ellis (2002). *ASCI Applications Software Quality Engineering Practices*. SAND2002-0121. Albuquerque, NM: Sandia National Laboratories.

4. Ayyub, B. M. (2001). *Elicitation of Expert Opinions for Uncertainty and Risks*. New York: CRC Press.

5. Balci, O. (1997). “Principles of Simulation Model Validation, Verification, and Testing.” *Trans. Soc. for Computer Simulation International* 14: 3–12.

6. Bedford, T., and R. Cooke (2001). *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge, UK: Cambridge University Press.

7. Boyack, B. E., et al. (1990). “Quantifying Reactor Safety Margins Part 1: An Overview of the Code Scaling, Applicability, and Uncertainty Evaluation Methodology.” *Nuclear Engineering and Design* 119: 1–15.

8. Budge, K. G. (1999). *Verification of the Radiation Package in ALEGRA*. SAND99-0786. Albuquerque, NM: Sandia National Laboratories.

9. Coleman, H. W., and W. G. Steele (1999). *Experimentation and Uncertainty Analysis for Engineers*. Second Edition. New York: John Wiley & Sons.

10. Coleman, H. W., and F. Stern (1997). “Uncertainties and CFD Code Validation.” *J. Fluids Engineering* 119: 795–803.

11. Collins, H. M. (1992). *Changing Order: Replication and Induction in Scientific Practice*. Chicago: University of Chicago Press.

12. Cox, D. R. (1958). *Planning of Experiments*. New York: John Wiley & Sons (republished in 1992).

13. Cragnolino, G. A., S. Mohanty, D. S. Dunn, N. Sridhar, and T. M. Ahn (2000). “An Approach to the Assessment of High-Level Radioactive Waste Containment – I: Waste Package Degradation.” *Nuclear Engineering and Design* 201: 289–306.

14. Dean, A., and D. Voss (1999). *Design and Analysis of Experiments*. New York: Springer Verlag.

15. DOE (U. S. Department of Energy) (1998). “Strategic Computing and Simulation Validation & Verification Plan.” Unpublished draft manuscript.

16. DOE (U. S. Department of Energy) (2000). *Advanced Simulation and Computing (ASCI) Program Plan*. 01-ASCI-Prog-001.

17. DOE/DP (U. S. Department of Energy Defense Programs) (2001a). *ASCI Software Quality Engineering: Goals, Principles, and Guidelines*. DOE/DP/ASC-SQE-2000-FDRFT-VERS2.

18. ——— (2001b). *Advanced Simulation and Computing (ASCI) Program Plan*. 01-ASCI-Prog-01.

19. Dowding, K. (2001). "Quantitative Validation of Mathematical Models." In *Proceedings of 2001 ASME International Mechanical Engineering Congress Exposition*, New York, NY, November 11–16.

20. Easterling, R. G. (2001a). *Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations*. SAND2001-0243. Albuquerque, NM: Sandia National Laboratories.

21. ——— (2001b). *Quantifying the Uncertainty of Computational Predictions*. SAND2001-0919C. Albuquerque, NM: Sandia National Laboratories.

22. Franklin, A. (1998). "Selectivity and the Production of Experimental Results." *Arch. His. Exact Sci*. 53: 399–485.

23. Gunter, B. H. (1993). "How Statistical Design Concepts Can Improve Experimentation in the Physical Sciences." *Computers in Physics* 7, no. 3: 262–272.

24. Haimes, Y. Y. (1998). *Risk Modeling, Assessment, and Management*. New York: John Wiley & Sons, Inc.

25. Haldar, A., and S. Mahadevan (2000). *Probability, Reliability, and Statistical Methods in Engineering Design*. New York : John Wiley & Sons, Inc.

26. Hills, R. G., and T. G. Trucano (1999). *Statistical Validation of Engineering and Scientific Models: Background*. SAND99-1256. Albuquerque, NM: Sandia National Laboratories.

27. Hills, R. G., and T. G. Trucano (2001a). *Statistical Validation of Engineering and Scientific Models with Application to CTH*. SAND2001-0312. Albuquerque, NM: Sandia National Laboratories.

28. ——— (2001b). *Statistical Validation of Engineering and Scientific Models: A Maximum Likelihood Based Metric*. SAND2001-1783. Albuquerque, NM: Sandia National Laboratories.

29. Hobbs, M., K. Erickson, and T. Y. Chu (1999). *Modeling Decomposition Unconfined Rigid Polyurethane Foam*. SAND99-2758. Albuquerque, NM: Sandia National Laboratories.

30. Kleindorfer, G. B., L. O'Neill, and R. Ganeshan (1998). "Validation in Simulation: Various Positions in the Philosophy of Science." *Management Science* 44: 1087–1099.

31. Kleijnen, J. P. C. (1995). "Verification and Validation of Simulation Models." *European Journal of Operational Research* 82: 145–162.

32. Kumamoto, H., and E. J. Henley (1996). *Probabilistic Risk Assessment and Management for Engineers and Scientists*. New York: The Institute of Electrical and Electronics Engineers, Inc.

33. McMillan, C. (1996). "Challenge Problems." Unpublished manuscript, Lawrence Livermore National Laboratory.

34. Melchers, R. E. (1999). *Structural Reliability and Prediction*. New York: John Wiley & Sons.

35. Mohanty, S., R. B. Codell, T. M. Ahn, and G. A. Cragnolino (2000). "An Approach to the Assessment of High-Level Radioactive Waste Containment – II: Radionuclide Releases from an Engineered Barrier System." *Nuclear Engineering and Design* 201: 307–325.

36. Oberkampf, W. L. (1994). “A Proposed Framework for Computational Fluid Dynamics Code Calibration/Validation.” In *18th AIAA Aerospace Ground Testing Conference*, American Institute of Aeronautics and Astronautics, 1994. AIAA Paper No. 94-2540.

37. Oberkampf, W. L. (1998). *Bibliography of Verification and Validation in Computational Simulation*. SAND98-2041. Albuquerque, NM: Sandia National Laboratories.

38. Oberkampf, W. L. (2000). “Design, Execution, and Analysis of Validation Experiments.” *Verification and Validation of Computational Fluid Dynamics*, von Karman Institute for Fluid Dynamics Lecture Series, Rhode-Saint-Genese, Belgium, June 5-8, 2000. AIAA Paper No. 2000-08 (also SAND2000-1315).

39. Oberkampf, W. L., and D. P. Aeschliman (1992). “Joint Computational Experimental Aerodynamics Research On A Hypersonic Vehicle 1. Experimental Results.” *AIAA Journal* 30, no. 8: 2000–2009.

40. Oberkampf, W. L., D. P. Aeschliman, J. F. Henfling, and D. E. Larson (1995). “Surface Pressure Measurements for CFD Code Validation in Hypersonic Flow.” In *26th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics. AIAA Paper No. 95-2273.

41. Oberkampf, W. L., and F. G. Blottner (1998). “Issues in Computational Fluid Dynamics Code Verification and Validation.” *AIAA Journal* 36, no. 5: 687–695.

42. Oberkampf, W. L., and T. G. Trucano (2000). “Validation Methodology in Computational Fluid Dynamics.” In *Fluids 2000 Conference*, June 19–22, 2000, Denver, CO. AIAA Paper No. 2000-2549.

43. Oberkampf, W. L., and T. G. Trucano (2002). “Verification and Validation in Computational Fluid Dynamics.” *Progress in Aerospace Sciences* (to be published), published as SAND2002-0529.

44. Paez, T., and A. Urbina (2001). “Validation of Structural Dynamics Models Via Hypothesis Testing.” In *Proceedings of the 2001 SEM Annual Conference*, Portland, Oregon.

45. Pilch, M., T. Trucano, J. Moya, G. Froehlich, A. Hodges, and D. Peercy (2000a*). Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format: Version 2.0*. SAND2000-3101. Albuquerque, NM: Sandia National Laboratories.

46. ——— (2000b). *Peer Review Process for the Sandia ASCI V&V Program: Version 1.0*. SAND2000-3099. Albuquerque, NM: Sandia National Laboratories.

47. Roache, P. J. (1998). *Verification and Validation in Computational Science and Engineering*. Albuquerque, NM: Hermosa Publishers.

48. Robinson, S. (1999). “Simulation Verification, Validation and Confidence: A Tutorial.” *Trans. Soc. for Computer Simulation International* 16: 63–69.

49. Sindir, M. M., S. L. Barson, D. C. Chan, and W. H. Lin (1996). “On the Development and Demonstration of a Code Validation Process for Industrial Applications.” In *27th AIAA Fluid Dynamics Conference*, New Orleans, LA. AIAA Paper No. 96-2032.

50. Murray-Smith, D. J. (1998) “Methods for the External Validation of Continuous Systems Simulation Models: A Review.” *Mathematical and Computer Modelling of Dynamical Systems* 4: 5–31.

51. Trucano, T. G., K. G. Budge, J. Lawrence, J. Asay, C. Hall, K. Holland, C. Konrad, W. Trott, G. Chandler, and K. Fleming (1999). *Analysis of Z Pinch Shock Wave Experiments*. SAND99-1255. Albuquerque, NM: Sandia National Laboratories.

52. Trucano, T. G., and J. L. Moya (1999). *Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format: Version 1.0*. SAND99-3098. Albuquerque, NM: Sandia National Laboratories.

53. Trucano, T. J., R. G. Easterling, K. J. Dowding, T. L. Paez, A. Urbina, V. J. Romero, B. M. Rutherford, and R. G. Hills (2001). *Description of the Sandia Validation Metrics Project*. SAND2001-1339. Albuquerque, NM: Sandia National Laboratories.

54. Urbina A. and T. L. Paez (2001). "Statistical Validation Of Structural Dynamics Models." In *Proceedings of the 47th Annual Technical Meeting & Exposition of the Institute of Environmental Sciences and Technology*, Phoenix, Arizona, April 22–25, 2001.

55. Younger, S. M. (1997). "Confidence in the Absence of Full System Testing." Unpublished manuscript, Los Alamos National Laboratory.

Appendix A: Assessment of a Radiation-Hydrodynamics Experiment Activity

T. Trucano

A.1 Introduction

A computational study of a set of shock wave experiments performed on the Sandia Z machine in 1998 is presented in the work of Trucano et al. (1999). The experiments discussed, Shots Z189 and Z190, were the first shock wave experiments performed on the Z machine in which the time-resolved VISAR interferometry technique was successfully applied to diagnose the capability of the radiation-hydrodynamics in the Sandia ALEGRA radiation-hydrodynamics code. The details will not be presented here, but are fully accounted for in Trucano et al. (1999). Because of the use of a technique involving secondary hohlraums on the Z machine, as well as because of the success of the VISAR system that was developed specifically for the time-resolved data acquisition, these experiments should properly be considered as major technical advances. They marked the beginning of a very successful Sandia experimental program in shock wave physics that uses the Z machine, which remains a very interesting (potential) validation capability for strong shock radiation-hydrodynamics. The question addressed in this appendix, however, is to what degree the work reported in Trucano et al. (1999) is compatible with the concepts developed in this report.

The original hope in performing computational assessment of these experiments was to extract significant validation information for radiation-hydrodynamics applications of the ALEGRA code. The ultimate conclusion of Trucano and his coauthors was that this goal was not achieved for the particular experiments Z189 and Z190. The following analysis provides a more recent perspective on why that was the case. In this analysis we will assess this activity, mainly through the information provided in the report by Trucano et al. (1999), the major documentation associated with the computational work. Any information used in this analysis that is not found in that report originates from the recollections of the primary author of these concepts and will be identified as such when used.

A.2 Summary of Conformance to the Concepts

To make consideration of each concept self-contained, we will repeat its definition along with our observations of the conformance of the work of Trucano et al. (1999) to the particular concept.

APP1: The validation experiment activity should be derived from the intended code application defined in an existing code-application V&V plan.

A V&V plan associated with a specific stockpile-driver application for ALEGRA did not exist at the time of this work.

Score = 0

PLAN1: The dedicated validation experiment activity should be part of a hierarchical validation activity that is defined by a PIRT. The planned validation experiments should then be well correlated with specific PIRT elements, and those elements should be clearly identified in the experimental plan.

A PIRT did not exist that properly located this experiment in the validation activities for ALEGRA. This implies that all scores for PIRT-related concepts are zero.

Score = 0

PLAN2: Information relevant to defining success and failure for comparison of code calculations with the results of experiments is identified in the PIRT.

Score = 0

PLAN3: The dedicated validation experiment activity should be defined in terms of the recommended Tier 1 through Tier 3 complexity structure if this is not explicit in the existing PIRT.

Score = 0

PLAN4: The validation experiments themselves should be defined in a formal documented plan.

The work reported in Trucano et al. (1999) attempted to leverage an ongoing high-quality laboratory radiation-hydrodynamics experimental activity. This activity was not aimed at validation, but rather at (1) experimental capability development and (2) phenomena discovery. Certainly no "validation plan" was developed. In addition, no plan existed that specified the use of ALEGRA in this experimental activity.

Score = 0

EED1: All applicable concepts in this report should be applied to guiding the use of existing experimental data in experimental validation activities.

This item does not strictly apply to this work. The experimental work was performed and the data were analyzed while contact with the ALEGRA analysis was ongoing. This concept is strictly aimed at previously acquired data that has been analyzed and reported prior to its use in code application validation.

Score = 0

VER1: The code verification status should be understood by the validation analyst and documented and determined to be adequate for the pursuit of an associated validation experiment activity.

A variety of information existed regarding the code verification status, but none of it was fully documented. The situation for these particular calculations was further complicated because the radiation transport package (SPARTAN) used in these particular calculations was developed and maintained at Los Alamos National Laboratory. Associated information on code verification was therefore not under the control of the ALEGRA team.

Score = 1

VER2: The existing VERTS for the code should contain elements that are believed to be in alignment with the associated validation experiment activity. Calculation verification should be performed and documented for these specific VERTS elements.

A radiation-hydrodynamics verification benchmark suite was in existence at the time of this work and was documented (Budge 1999). It is interesting that this report is not referenced at all in Trucano et al. (1999). Also, this set of benchmarks is not strictly relevant to the SPARTAN (radiation transport package) calculations that were performed. Instead, this verification suite was (and is) focused on testing Sandia-developed multigroup diffusion packages and related radiation transport schemes. Significant details of calculation verification were addressed in the Budge report, but a completely systematic effort was not documented.

Score = 1

VER3: New VERTS elements should be defined if there is inadequate coverage in the code VERTS to contribute to assessing code verification status for the planned validation experiment activity. The calculation verification of these new elements should be performed and documented.

New VERTS problems were not defined and studied as part of this work.

Score = 0

VER4: A calculation verification strategy (typically centered on convergence studies and *a posteriori* error estimation) should be defined for the calculations performed in the validation activity.

Calculation verification was not addressed in Trucano et al. (1999). Some issues of grid sensitivity and radiation group resolution were studied, but as part of computational uncertainty and not as formal convergence and accuracy studies.

Score = 0

VER5: All necessary information required for the verification assessment for the validation experiment activity should be documented.

Score = 0

DES1: Validation experiments should be explicitly designed to support assessment of code fidelity and confidence for the intended application through precise and conclusive comparisons of calculations with experimental data.

This factor did not enter into the design of these experiments.

Score = 0

DES2: The planned validation experiments should specifically address the balance of resources for experiments, code capability, and required predictive confidence for the intended application.

No information, plan or otherwise, was devoted to discussing this balance.

Score = 0

DES3: The region of intended application domain parameters that is covered by the validation experiment activity should be defined in the plan. It should be understood whether the intended application extrapolates the validation domain, interpolates the validation domain, or both.

The application domain and its associated parameters were not discussed in this work.

Score = 0

DES4: One or more experiments should be designed and performed with the goal of resolving the boundary of credibility of the code for the intended application.

Failure to compare adequately with ALEGRA was not one of the objectives of this work.

Score = 0

DES5: Statistical design of experiments should be applied in the design of the experimental activity.

Statistical design of experiments was not applied.

Score = 0

DES6: Experimental quantification of uncertainty, both variability and bias, should be performed. This should include planned experimental repeats to quantify variability as well as diagnostic fidelity.

Experimental error bars were developed through fairly painstaking work. The difficulty of this suggests the importance of, for example, doing repeat experiments to characterize experimental variability. No repeat experiments were performed. There was considerable concern in the experimental work, partly as a result of the analysis, about the adequacy of the characterization and repeatability of the shock waves that were generated in these experiments. All in all, the conclusion is that experimental uncertainty was very important and not characterized in any way other than by diagnostic error bars.

Score = 1

DES7: Data resulting from the validation experiment activity and their interpretation should be robust in the sense described in this report. If not, nonrobustness of data should be specifically emphasized in documented outcomes.

Difficulty with robustness was addressed, though inadequately. The documentation in Trucano et al. (1999) is also not the ideal vehicle for discussion of these issues. Knowledge about the concern for robustness in these and similar experiments resulted from informal communication among the team, and some of the issues that were discussed were never reported in the above document.

Score = 1

DES8: Application of the code to the definition, design, and postexperiment analysis should be performed as part of the experimental activity.

ALEGRA was used only informally during the experiment design phase, and Trucano et al. (1999) discuss the extensive postexperiment analysis for which it was used. ALEGRA did not enter into the definition of the experiments.

Score = 1

DES9: The validation experiment activity should consist of a team that includes experimenters, code users, and code developers.

As represented by the authors on the report (Trucano et al. 1999), experimenters, code users, and one code developer (Budge) were participants in the activity.

Score = 3

DES10: The planned validation experiments should not be phenomena exploration experiments or mathematical model development experiments. If phenomena exploration is required and performed as part of the experimental activity, it should be distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon the phenomena exploration activity should be explicitly defined in the plan and in the experimental outcomes.

Shots Z189 and Z190 had elements of phenomena exploration. But worse, from a validation perspective, is that these experiments were mainly focused on experimental capability development. This was their great triumph, but one of their biggest weaknesses for code application validation.

Score = 0

DES11: The validation experiment(s) should not be a calibration experiment. If calibration is required and performed as part of the experimental activity, it should be clearly distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon an included calibration activity should be explicitly defined in the plan and in the experimental outcomes.

Shots Z189 and Z190 were not calibration experiments.

Score = 3

MET1: All validation information resulting from the validation experiment activity should be based on quantitative comparisons of computational and experimental results.

Curve overlays were the major methods used to compare calculations and experimental data. Some specific comments about differences in features on plots were given. Error bars were included on overlays of calculations and experimental data.

Score = 1.

MET2: Experimental data should account for uncertainty when applied in validation metrics.

No accuracy requirements are specified. However, it is clear that some of the experimental error bars specified in Trucano et al. (1999) are too imprecise for certain application validation needs. A major issue was also revealed about uncertainty in an important boundary condition, but this was not quantified.

Score = 0.

MET3: Computations should account for uncertainty when applied in validation metrics.

Formal computational uncertainty analyses were not performed. Some effort was devoted to assessing the influence of various specifications in the modeling, such as grid resolution, group resolution for the radiation transport algorithms, and treatment of the radiative source boundary condition used in the calculations. The boundary condition is an important candidate for more formal methods of quantifying the impact of uncertainty in this modeling, and that was not performed.

Score = 1

MET4: Statistical analysis of the quantitative comparison between calculations and experiments should be performed.

Statistical analysis of the comparisons between calculations and experiments was not performed.

Score = 0

MET5: A statistician should be a member of the validation team.

A statistician did not participate in this modeling activity.

Score = 0

SF1: Success criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

Success criteria were not defined.

Score = 0

SF2: Failure criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

Failure criteria were not stated. However, significant discussion was presented that revolved around this issue in an indirect fashion.

Score = 1

PRE1: The contribution of the validation experiment activity to understanding credibility of the code for the intended application should be characterized.

The prediction paradigm for stockpile computing dependent upon this work is not discussed at all.

Score = 0

DOC1: Validation experiment activity documentation should be compatible with the Sandia V&V program's records management system.

These requirements do not yet exist.

Score = 0

DOC2: Validation experiment activity documentation should be integrated into the associated V&V documentation tree for the code and its applications.

The conclusion of this work was that no significant validation was accomplished. This work is referenced as "evidence" when discussing the V&V documentation tree for ALEGRA, mainly because it is one of the few cases where ALEGRA is compared with important radiation-hydrodynamics data.

Score = 1

DOC3: Validation experiment activity documentation should contain information on the following topics:

- Information on the application and requirements that are driving the validation experiment activity, including references to information DP documentation and the overall V&V plan that the activity is part of. (NO)

- Specific discussion on the associated PIRT and how the validation experiment activity is located in it. (NO)

- A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity. (NO)

- A comprehensive discussion of the definition, design, and analysis of each validation experiment via use of the code. (YES, within the limits discussed above)

- A complete description of each experiment, sufficient to allow experimental repeats in the future. (YES, to a degree.)

- A description of the analysis of experimental data, including quantification of uncertainty in the acquired experimental data. (YES, within the limits discussed above.)

- The methods and results of the validation metrics applied in the validation experiment activity, including the defined success and failure criteria for these metrics. (YES, within the limits described above.)

- An assessment of confidence in the code application resulting from the application of the defined metrics, the results, and their performance versus the defined success and failure criteria. (NO)

- Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application. (NO)

Score = 1

A.3 Scoring Summary

The aggregate scores for all of the experimental validation concepts for this work are shown in Figure A.1.

The main purpose behind doing this assessment is to demonstrate that the presented concepts can be appraised and that this appraisal can provide useful information, even in as unlikely a situation as this example. Figure A.1 suggests that most of the concepts in the present report were not operational at all in this particular study. The scoring of this example further supports the conclusion that was specifically drawn in Trucano et al. (1999): the computational analysis of the experiments reported in that document was insufficient to be called validation.

One further comment is appropriate at this point. An interesting approach when appraising particular validation experiments is to select specific concepts or sets of concepts and understand the pattern of their scores across one or more validation activities. This information more appropriately conveys a sense of strength or weakness about validation experiment concepts that are deemed particularly important, as well as correlations among these concepts. There are other alternatives for extracting useful information from an appraisal of the conformance of validation experiment activities with respect to the guidance developed in this report that we hope to develop in future applications of this validation methodology.

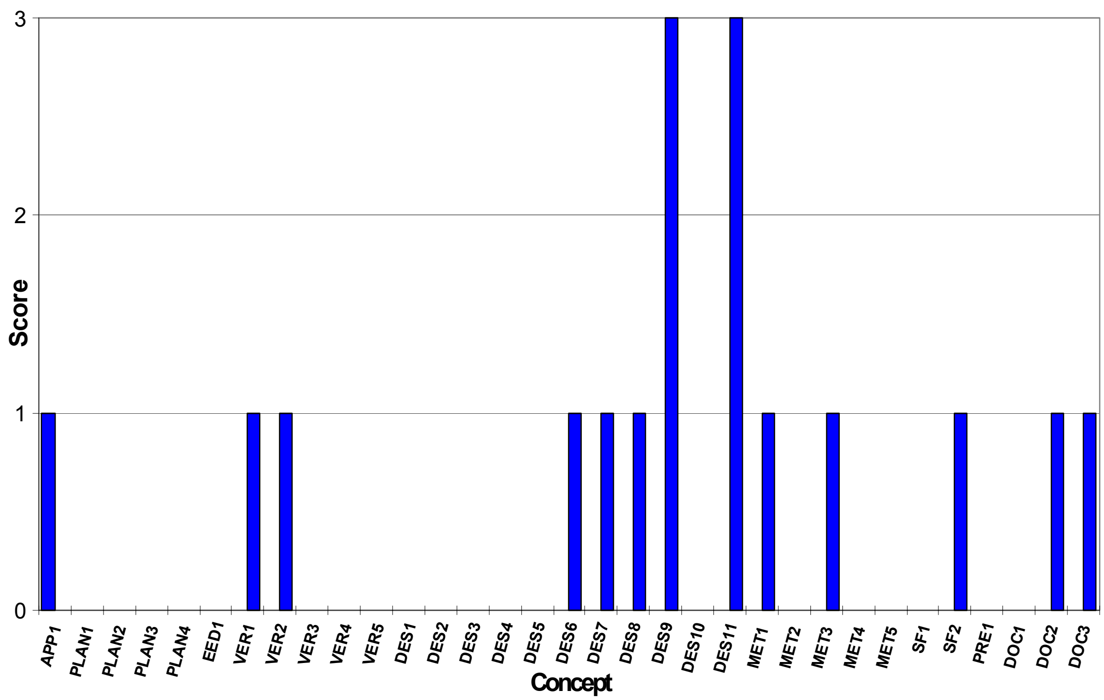

Figure A.1. Scoring summary.

Appendix B: Assessment of a Structural Mechanics Validation Experiment Activity

A. Urbina and T. Paez

B.1 Introduction

This investigation developed probabilistic models of measures of behavior of mathematical models of structural dynamic systems, as well as the corresponding experimental measures of behavior of the physical systems being modeled. We used this information in a probabilistic/statistical framework to determine the nonacceptability or acceptability of the mathematical model. This can be done for arbitrary measures of system behavior; however, we chose to focus on a metric of the spectral density of system response. Spectral density is the fundamental and most widely employed measure of second-order statistical behavior in systems excited with stationary random inputs. A detailed description of this validation activity can be found in Urbina and Paez (2001). The targeted code for the assessment is SALINAS.

The intent of this case study was to develop a validation metric and a technique to be applied to a structural dynamic system. Although the physical system that was used for this case study was very simple (when compared to a weapon system), it is expected that the validation techniques developed and metrics could also be applied to an actual weapon system. As such, our study did not fully align with the PIRT or the V&V plans for SALINAS (the code used in this case study). Both documents are aimed at validation activities for a weapon system. To find out how well our case study conformed to the criteria of this experimental validation document, we rated aspects of our study that were similar to those elements found in the PIRT and in the V&V plans even though they are clearly written for a more complex system.

B.2 Summary of Conformance to the Guidelines

The following is a summary of two independent assessments by the authors of the conformance of our case study to the guidelines presented in this document. Comments are made when appropriate to justify the scoring in each individual category. A graph summarizing the scoring by category from each assessment is presented in Section B.3.

APP1: The validation experiment activity should be derived from the intended code application defined in an existing code-application V&V plan.

The V&V plan addresses weapon components. Although our case study is a simpler system, it focuses on validation based on spectral density. This is considered in the V&V plan.

Score = 1 (Urbina)   Score = 0 (Paez)

PLAN1: The dedicated validation experiment activity should be part of a hierarchical validation activity that is defined by a PIRT. The planned validation experiments should then be well correlated with specific PIRT elements, and those elements should be clearly identified in the experimental plan.

The target code application has a PIRT and our case study, although not related to weapon components, is well correlated with a specific PIRT element (Generic bolted joint response: Linear isotropic material modal response).

Score = 1 (Urbina) Score = 0 (Paez)

PLAN2: Information relevant to defining success and failure for comparison of code calculations with the results of experiments is identified in the PIRT.

No explicit information is given in the PIRT.

Score = 0 (Urbina) Score = 0 (Paez)

PLAN3: The dedicated validation experiment activity should be defined in terms of the recommended Tier 1 through Tier 3 complexity structure if this is not explicit in the existing PIRT.

Score = 0 (Urbina) Score = 1 (Paez)

PLAN4: The validation experiments themselves should be defined in a formal documented plan.

The case study's experiment was planned but not fully documented. While the code was used in definition, design, and analysis, these intended uses were not planned and documented in advance of the experiments.

Score = 2 (Urbina) Score = 2 (Paez)

EED1: All applicable guidelines in this report should be applied to guiding the use of existing experimental data in experimental validation activities.

No existing experimental data were used. Experimental data were collected at the same time the analysis was being performed.

Score = 0 (Urbina) Score = 0 (Paez)

VER1: The code verification status should be understood by the validation analyst and documented and determined to be adequate for the pursuit of an associated validation experiment activity.

The PIRT indicates that the model adequacy is unknown.

Score = 0 (Urbina) Score = 0 (Paez)

VER2: The existing VERTS for the code should contain elements that are believed to be in alignment with the associated validation experiment activity. Calculation verification should be performed and documented for these specific VERTS elements.

The existing VERTS are aligned with the case study. Calculation verification documentation has not been found.

Score = 2 (Urbina) Score = 1 (Paez)

VER3: New VERTS elements should be defined if there is inadequate coverage in the code VERTS to contribute to assessing code verification status for the planned validation experiment activity. The calculation verification of these new elements should be performed and documented.

New VERTS have not been defined.

Score = 0 (Urbina)   Score = 0 (Paez)

<u>VER4</u>: A calculation verification strategy (typically centered on convergence studies and *a posteriori* error estimation) should be defined for the calculations performed in the validation activity.

None has been found in the documentation.

Score = 0 (Urbina)   Score = 3 (Paez)

<u>VER5</u>: All necessary information required for the verification assessment for the validation experiment activity should be documented.

Score = 0 (Urbina)   Score = 0 (Paez)

<u>DES1</u>: Validation experiments should be explicitly designed to support assessment of code fidelity and confidence for the intended application through precise and conclusive comparisons of calculations with experimental data.

Yes

Score = 2 (Urbina)   Score = 3 (Paez)

<u>DES2</u>: The planned validation experiments should specifically address the balance of resources for experiments, code capability, and required predictive confidence for the intended application.

These were not addressed.

Score = 0 (Urbina)   Score = 1 (Paez)

<u>DES3</u>: The region of intended application domain parameters that is covered by the validation experiment activity should be defined in the plan. It should be understood whether the intended application extrapolates the validation domain, interpolates the validation domain, or both.

Score = 1 (Urbina)   Score = 2 (Paez)

<u>DES4</u>: One or more experiments should be designed and performed with the goal of resolving the boundary of credibility of the code for the intended application.

These experiments were not considered in this case study.

Score = 0 (Urbina)   Score = 0 (Paez)

<u>DES5</u>: Statistical design of experiments should be applied in the design of the experimental activity.

No statistical design of experiments was applied.

Score = 0 (Urbina)   Score = 0 (Paez)

<u>DES6</u>: Experimental quantification of uncertainty, both variability and bias, should be performed. This should include planned experimental repeats to quantify variability as well as diagnostic fidelity.

Uncertainty quantification was performed computationally (using experimental data). The experiment was repeated but results were not used.

Score = 2 (Urbina)   Score = 2 (Paez)

DES7: Data resulting from the validation experiment activity and their interpretation should be robust in the sense described in this report. If not, nonrobustness of data should be specifically emphasized in documented outcomes.

Score = 2 (Urbina) Score = 3 (Paez)

DES8: Application of the code to the definition, design, and postexperiment analysis should be performed as part of the experimental activity.

The code was used for definition and postexperiment analysis.

Score = 1 (Urbina) Score = 2 (Paez)

DES9: The validation experiment activity should consist of a team that includes experimenters, code users, and code developers.

Yes

Score = 3 (Urbina) Score = 2 (Paez)

DES10: The planned validation experiments should not be phenomena exploration experiments or mathematical model development experiments. If phenomena exploration is required and performed as part of the experimental activity, it should be distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon the phenomena exploration activity should be explicitly defined in the plan and in the experimental outcomes.

The validation experiment was not a phenomena exploration experiment.

Score = 3 (Urbina) Score = 3 (Paez)

DES11: The validation experiments should not be calibration experiments. If calibration is required and performed as part of the experimental activity, it should be clearly distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon an included calibration activity should be explicitly defined in the plan and in the experimental outcomes.

The validation experiment was not used as a calibration experiment. A separate experiment was done for calibration purposes.

Score = 3 (Urbina) Score = 3 (Paez)

MET1: All validation information resulting from the validation experiment activity should be based on quantitative comparisons of computational and experimental results.

Validation and confidence statements are based on analysis of computational and experimental differences.

Score = 2 (Urbina) Score = 3 (Paez)

MET2: Experimental data should account for uncertainty when applied in validation metrics.

For the case study, the accuracy was arbitrarily set. A formal definition is required for the weapon's components. This should be included in the V&V plans.

Score = 2 (Urbina) Score = 2 (Paez)

MET3: Computations should account for uncertainty when applied in validation metrics.

Yes

Score = 2 (Urbina)   Score = 2 (Paez)

MET4: Statistical analysis of the quantitative comparison between calculations and experiments should be performed.

Yes

Score = 3 (Urbina)   Score = 3 (Paez)

MET5: A statistician should be a member of the validation team.

A statistician was consulted but was not a full-time member of the validation team.

Score = 2 (Urbina)   Score = 2 (Paez)

SF1: Success criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

A success criterion was specified for this particular case study.

Score = 2 (Urbina)   Score = 2 (Paez)

SF2: Failure criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

A failure criterion was specified for this case study.

Score = 2 (Urbina)   Score = 3 (Paez)

PRE1: The contribution of the validation experiment activity to understanding credibility of the code for the intended application should be characterized.

The case study was explicitly planned to obtain best estimate plus uncertainty.

Score = 3 (Urbina)   Score = 3 (Paez)

DOC1: Validation experiment activity documentation should be compatible with the Sandia V&V program's records management system.

No

Score = 0 (Urbina)   Score = 1 (Paez)

DOC2: Validation experiment activity documentation should be integrated into the associated V&V documentation tree for the code and its applications.

No

Score = 0 (Urbina)   Score = 0 (Paez)

DOC3: Validation experiment activity documentation should contain information on the following topics:

- Information on the application and requirements that are driving the validation experiment activity, including references to information DP documentation and the overall V&V plan that the activity is part of. (NO)
- Specific discussion on the associated PIRT and how the validation experiment activity is located in it. (NO)

- A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity. (NO)
- A comprehensive discussion of the definition, design, and analysis of each validation experiment via use of the code. (NO)
- A complete description of each experiment, sufficient to allow experimental repeats in the future. (YES)
- A description of the analysis of experimental data, including quantification of uncertainty in the acquired experimental data. (YES)
- The methods and results of the validation metrics applied in the validation experiment activity, including the defined success and failure criteria for these metrics. (YES)
- An assessment of confidence in the code application resulting from the application of the defined metrics, the results, and their performance versus the defined success and failure criteria. (YES)
- Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application. (NO)

Score = 1 (Urbina) Score = 1 (Paez)

B.3 Scoring Summary

The graph in Figure B.1 shows a summary of the scoring.

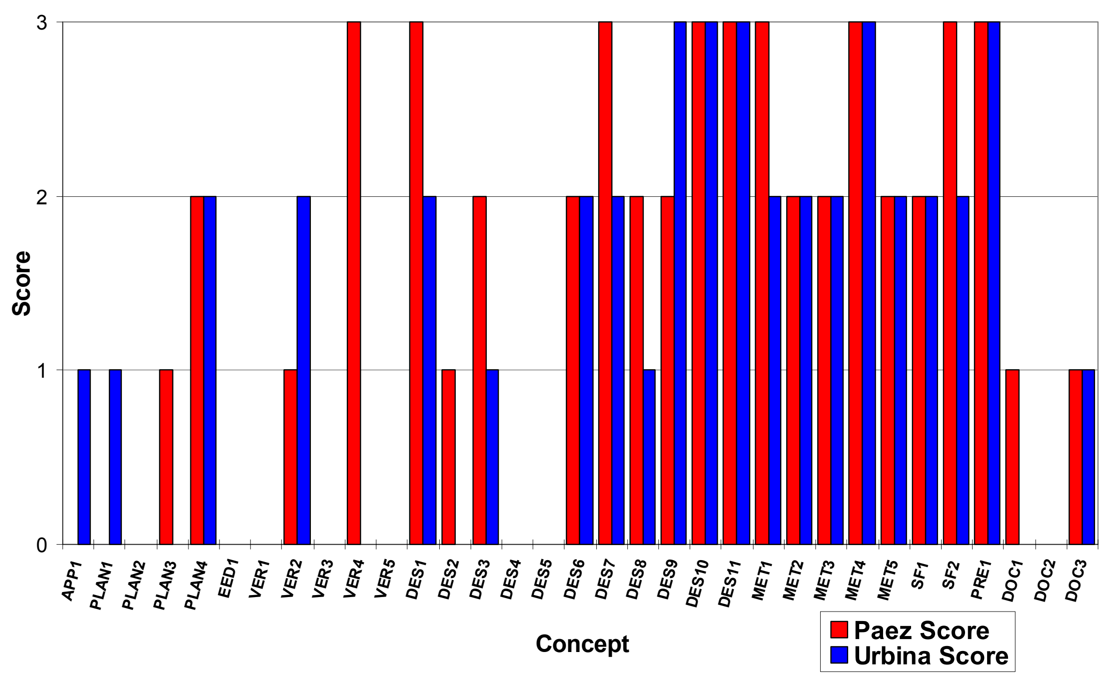

Figure B.1. Scoring summary.

From the figure, we also note the following:

1. Nineteen categories were scored the same by the two authors.
2. Urbina scored four categories higher than Paez.
3. Paez scored eleven categories higher than Urbina.
4. Categories that have a score of zero (eight categories) represent areas that were not applicable or addressed in our study.

Our personal interpretation of each question is reflected by the variation in the scoring. This could be a potential pitfall when filling in the assessment and looking at the resulting score. On the other hand, this assessment does bring out the strengths and weakness in the validation activity. In our case three areas of improvement are highlighted:

1. No experiments were designed to produce bad agreement.
2. No statistical design of experiments was used.
3. Our documentation needs improvement

In addition, this assessment will be a first step in writing a lessons-learned document on our progress on our validation metrics case study. Some conclusions extracted from this assessment follow. First, more work is needed in the planning phase to fully correlate the validation experiments with the V&V plans and the elements in the PIRT. Second, there is a need to define objective success and failure criteria. And third, these guidelines could be used in the test-planning phase as well as a posttest assessment tool.

(Page Left Blank)

Appendix C: Assessment of a Thermal Validation Experiment Activity

K. Dowding

C.1 Introduction

Predicting the thermal response of a weapon in an abnormal environment is an important application of ASCI-scale computer models. Predicting the “thermal race”—the time differential between two components reaching threshold temperatures—assesses the safety design of the weapon. The components of interest are potted in foam, such as polyurethane. In an abnormal environment the temperature/heat fluxes are high enough to initiate and sustain thermal decomposition of the foams encasing the components. Since the presence of foam greatly impacts the thermal response of the components, to accurately predict the thermal race with computer models requires addressing the decomposition and removal of foam.

A physics-based model for the thermal decomposition of polyurethane foam has been developed. The model is called CPUF (Hobbs et al. 1999). The model is based on analysis of the foam’s chemical structure, hypothesizing its chemical evolution under a thermal environment, and relating the chemistry to the macroscopic structure. The model development was supported by hundreds of thermal gravimetric analysis (TGA) type experiments. The TGA was used for discovery, model development, and finally for calibrating the finalized model, CPUF.

A second, separate series of experiments were dedicated to validating CPUF. A cylindrical can (6 cm in diameter and 9 cm in length) was filled with foam and exposed to a high-temperature environment at one end of the can. X-ray imagery tracked the surface of the foam as it decomposed. A total of 15 experiments was conducted at unconfined conditions. Validation of CPUF was studied for conditions of “driving” boundary temperature, orientation, density, and presence of a component in the foam. These dedicated validation experiments are evaluated with regard to the experimental guidelines.

At the time of writing, the analysis of the validation experiments had not been finalized. Comparisons between the model and experiment for unconfined conditions (ambient pressure) had been done, but were not complete. A set of experiments at confined conditions (greater than ambient pressure) was conducted, but had not been compared to the model. Hence, the evaluation that follows is for the validation experiments at unconfined conditions.

C.2 Summary of Conformance to the Guidelines

APP1: The validation experiment activity should be derived from the intended code application defined in an existing code-application V&V plan.

The validation experiments support the code application of assessing "assured nuclear safety in abnormal thermal environments." The code application is defined in an existing V&V plan. The validation activities were driven by the application requirements as best the requirements were understood at the time the experiments were conducted.

Score = 2

PLAN1: The dedicated validation experiment activity should be part of a hierarchical validation activity that is defined by a PIRT. The planned validation experiments should then be well correlated with specific PIRT elements, and those elements should be clearly identified in the experimental plan.

The validation experiments are correlated with the PIRT element related to thermal decomposition of polyurethane foam. The ranking or importance of this particular PIRT element, relative to other elements in the PIRT, has not been identified.

Score = 2

PLAN2: Information relevant to defining success and failure for comparison of code calculations with the results of experiments is identified in the PIRT.

No success or failure criteria are defined.

Score = 0

PLAN3: The dedicated validation experiment activity should be defined in terms of the recommended Tier 1 through Tier 3 complexity structure if this is not explicit in the existing PIRT.

The validation activity is identified as a Tier 2 activity in the tiered complexity structure. The relationship of the planned validation activity to the overall complexity structure is not well characterized. Thus it is not clear how the planned validation experiments are related to past validation experiments or carry forward into other validation activities.

Score = 1

PLAN4: The validation experiments themselves should be defined in a formal documented plan.

While no plan for the experimental activities exists, a summary of the validation activities is documented in memo format. The document summarizes the experiments conducted, conditions for each experiment, and any notable comments about the experiment. The summary was produced after conducting the validation experiments, but the test matrix covered was planned before conducting the experiments. We believe that the code was applied for designing some aspects of the validation experiment. The design calculations were not documented.

Score = 1

EED1: All applicable guidelines in this report should be applied to guiding the use of existing experimental data in experimental validation activities.

The current study utilized dedicated experiments, not previously existing data.

Score = 0

<u>VER1</u>: The code verification status should be understood by the validation analyst and documented and determined to be adequate for the pursuit of an associated validation experiment activity.

The code studied was Coyote. The code verification status is not formally documented, but Coyote has been used extensively and is generally deemed adequate to pursue validation.

Score = 1

<u>VER2</u>: The existing VERTS for the code should contain elements that are believed to be in alignment with the associated validation experiment activity. Calculation verification should be performed and documented for these specific VERTS elements.

A VERTS does not exist.

Score = 0

<u>VER3</u>: New VERTS elements should be defined if there is inadequate coverage in the code VERTS to contribute to assessing code verification status for the planned validation experiment activity. The calculation verification of these new elements should be performed and documented.

No additional verification tests (VERTS elements) have been conducted.

Score = 0

<u>VER4</u>: A calculation verification strategy (typically centered on convergence studies and *a posteriori* error estimation) should be defined for the calculations performed in the validation activity.

A calculation verification strategy has not been defined for the validation activity but studies have been performed on a simplified model (1D) of the experiment. The dependence of the 1D calculations on the discretization (time and space) has been studied, and a bias correction for the 2D calculations was developed.

Score = 2

<u>VER5</u>: All necessary information required for the verification assessment for the validation experiment activity should be documented.

Verification assessment is not documented.

Score = 0

<u>DES1</u>: Validation experiments should be explicitly designed to support assessment of code fidelity and confidence for the intended application through precise and conclusive comparisons of calculations with experimental data.

The experiments were not specifically designed to support assessment, particularly with respect to a metric.

Score = 0

<u>DES2</u>: The planned validation experiments should specifically address the balance of resources for experiments, code capability, and required predictive confidence for the intended application.

No plan exists. The noted summary document does not address the issues, either.

Score = 0

DES3: The region of intended application domain parameters that is covered by the validation experiment activity should be defined in the plan. It should be understood whether the intended application extrapolates the validation domain, interpolates the validation domain, or both.

The region of application domain parameters covered by the validation experiments was identified.

Score = 1

DES4: One or more experiments should be designed and performed with the goal of resolving the boundary of credibility of the code for the intended application.

No experiments were conducted to produce bad agreement with the code. It was, however, expected that experiments at the lower temperature threshold and for a heating orientation from the top would have worse agreement than higher temperatures and heating from the bottom.

Score = 1

DES5: Statistical design of experiments should be applied in the design of the experimental activity.

No statistical design of experiments was conducted.

Score = 0

DES6: Experimental quantification of uncertainty, both variability and bias, should be performed. This should include planned experimental repeats to quantify variability as well as diagnostic fidelity.

Quantification of experimental uncertainty was not addressed. Experiments were only repeated when the anomalous results were observed. The experimental test matrix has some experiments that could be used as repeats over a given duration of the experiment.

Score = 0

DES7: Data resulting from the validation experiment activity and their interpretation should be robust in the sense described in this report. If not, nonrobustness of data should be specifically emphasized in documented outcomes.

Robustness was not specifically addressed.

Score = 0

DES8: Application of the code to the definition, design, and postexperiment analysis should be performed as part of the experimental activity.

The code has been applied extensively for postexperiment analysis.

Score = 2

DES9: The validation experiment activity should consist of a team that includes experimenters, code users, and code developers.

The team includes experimenters and the (math) model developers. One model developer served as a code user and performed some code development, but was not a part of the core code-development team.

Score = 2

DES10: The planned validation experiments should not be phenomena exploration experiments or mathematical model development experiments. If phenomena exploration is required and performed as part of the experimental activity, it should be distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon the phenomena exploration activity should be explicitly defined in the plan and in the experimental outcomes.

The validation experiment is not a phenomena exploration experiment.

Score = 3

DES11: The validation experiments should not be calibration experiments. If calibration is required and performed as part of the experimental activity, it should be clearly distinguished from the validation activity. Dependence of inferred confidence from the validation activity upon an included calibration activity should be explicitly defined in the plan and in the experimental outcomes.

The validation experiments do not have calibration directly associated with the experiments. A separate series of calibration experiments were conducted to support model development and identify model parameters. The calibration experiments are separate and independent of the validation activities.

Score = 3

MET1: All validation information resulting from the validation experiment activity should be based on quantitative comparisons of computational and experimental results.

A study of the differences between the model and experiment is planned.

Score = 3

MET2: Experimental data should account for uncertainty when applied in validation metrics.

Neither the accuracy of the experimental data nor the requirements for the code application are known.

Score = 0

MET3: Computations should account for uncertainty when applied in validation metrics.

Inputs to the computational model with expected uncertainty in their values have been propagated through the model. Error bars have been calculated for the propagated uncertainty.

Score = 2

MET4: Statistical analysis of the quantitative comparison between calculations and experiments should be performed.

The differences between the calculations and experiments were statistically analyzed.

Score = 2

MET5: A statistician should be a member of the validation team.

A statistician is a member of the validation team.

Score = 3

SF1: Success criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

No success criteria exist for the outcome of a metric.

Score = 0

SF2: Failure criteria should be defined that will be applied to assess the results of validation metrics for comparison of calculations and experiments.

No failure criteria exist for the outcome of the metric.

Score = 0

PRE1: The contribution of the validation experiment activity to understanding credibility of the code for the intended application should be characterized.

The role of the validation activity in supporting **BE+U** for the code application is not explicitly understood or incorporated into the validation experiments. It is unclear how to relate the **BE** for the code application to an appropriate measure in the validation experiment or how the validation activity supports assessing **BE+U**.

Score = 0

DOC1: Validation experiment activity documentation should be compatible with the Sandia V&V program's records management system.

Not applicable (?)

Score = 0

DOC2: Validation experiment activity documentation should be integrated into the associated V&V documentation tree for the code and its applications.

No

Score = 0

DOC3: Validation experiment activity documentation should contain information on the following topics:

Information on the application and requirements that are driving the validation experiment activity, including references to information DP documentation and the overall V&V plan that the activity is part of. (in process)

- Specific discussion on the associated PIRT and how the validation experiment activity is located in it. (NO)
- A comprehensive discussion of verification activities, both code and calculation, centered on the validation experiment activity. (NO)
- A comprehensive discussion of the definition, design, and analysis of each validation experiment via use of the code. (NO)
- A complete description of each experiment, sufficient to allow experimental repeats in the future. (YES)
- A description of the analysis of experimental data, including quantification of uncertainty in the acquired experimental data. (NO)
- The methods and results of the validation metrics applied in the validation experiment activity, including the defined success and failure criteria for these metrics. (NO)
- An assessment of confidence in the code application resulting from the application of the defined metrics, the results, and their performance versus the defined success and failure criteria. (NO)

- Information about the contribution of the validation experiment activity to the BE+U paradigm for predictive code application. (NO)

Score = 1

C.3 Scoring Summary

The scoring summary is presented in Figure C.1.

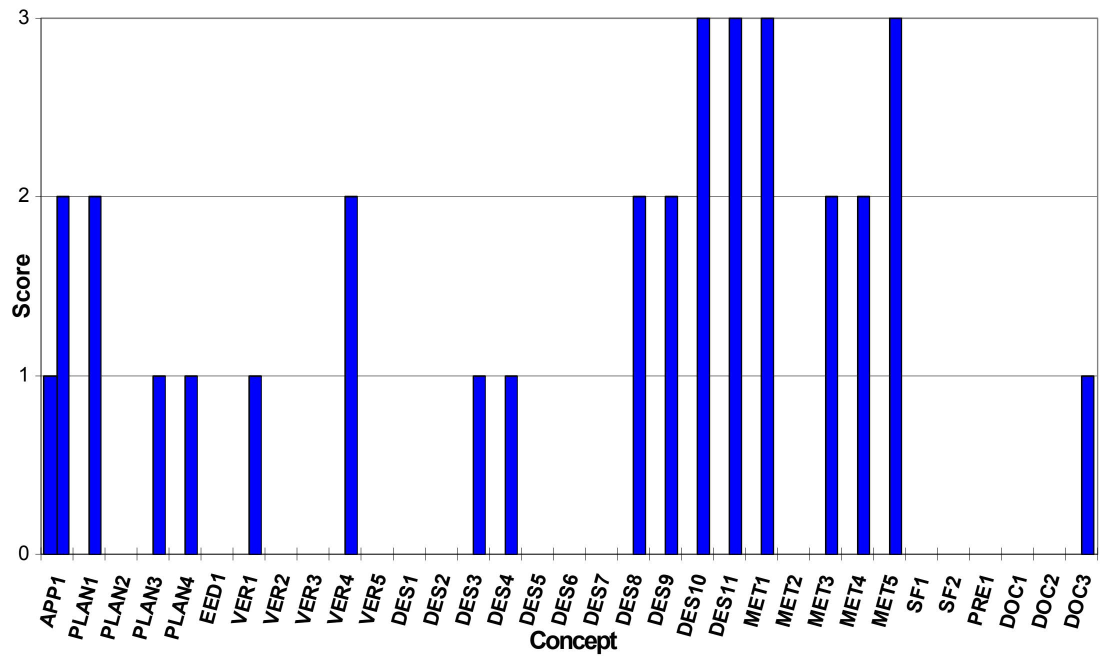

Figure C.1. Scoring summary.

A summary of scores by concept group is listed below. The summary is organized by grouping the concepts, which is a useful way to understand the overall assessment. There are nine groupings and a group may have one or more concepts. On average a particular concept is poorly addressed, but present. Several groups, including those addressing prediction (**PRE**), existing experimental data (**EED**), and success/failure (**SF**) are not addressed at all. These groups account for only 4 out of 34 concepts, however. For the groups that are addressed, the requirements (**REQ**) and metrics (**MET**) have the highest group average (2). Both groups are well addressed with some remediation identified. The experimental design (**DES**) and PIRT (**PRT**) are poorly addressed, but present. The verification (**VER**) and documentation (**DOC**) have the lowest (group) average, 0.6 and 0.33, respectively.

Group	Number of Principles	Score	Average Score for Group
REQ	1	2	2
PRT	3	3	1
PRE	1	0	0
EED	1	0	0
VER	5	3	0.6
DES	13	13	1
MET	5	10	2
SF	2	0	0
DOC	3	1	0.33
Total	34	32/102 = 0.31	6.93/9 = 0.77

The outcome of this evaluation process shows that this validation project (1) has a limited understanding of what validation experiments should encompass (at the time these experiments were conducted); and (2) the experiments involved were more definable as model development/physics exploration. Issues of prediction, success and failure, and existing experimental data were not addressed; the experimental design and the PIRT were only poorly addressed.

Appendix D: Examples of PIRT Information Categories

In Section 3, Figure 3.2 illustrates the kind of detailed information of relevance in defining and assessing validation experiments that is included in, or that can be synthesized from, PIRTs. This information was categorized in Figure 3.2 as "Importance," "Conceptual Model Adequacy," "Code Adequacy," "Experimental Adequacy," and "Validation Adequacy." This appendix provides examples of what the nature of this information could be.

Importance defines a convenient framework for quantitatively assessing the importance of an identified PIRT element to a potential experimental validation activity. Something akin to this framework is required to properly prioritize experimental validation activities and their implied requirements on the other categories of information in the context of PIRT rankings. A metric that relates the defined PIRT element to the intended model application and required confidence in this application is implied in this prioritization. An example of a ranking system for importance follows.

IMPORTANCE (Example)		
Score	**Descriptor**	**Definition**
2	High	The PIRT element has first-order importance to the metric of interest. Model adequacy, code capability, and validation adequacy should be at the "Adequate Level."
1	Medium	The PIRT element has second-order importance to the metric of importance. Model adequacy, code capability, and validation should at least be at the "Incomplete Level."
0	Low	The PIRT element has negligible importance to the metric of interest. It is not necessary to model this phenomenon for this application.
2	Uncertain	The PIRT element is potentially important. A better understanding of its importance should be explored through sensitivity studies or bounding experiments and the PIRT then revised.

Conceptual Model Adequacy ranks current understanding of the adequacy of the underlying conceptual model in the code as it relates to a chosen PIRT element. An example of a ranking system for conceptual model adequacy follows.

CONCEPTUAL MODEL ADEQUACY (Example)		
Score	**Descriptor**	**Definition**
5	Adequate	One or more candidate *physics*-based models are available that are believed to adequately represent or bound the spectrum of possible model forms over the full parameter space of the application.
4		One or more candidate *correlation*-based models are available that are believed to adequately represent or bound the spectrum of possible correlation forms over the full parameter space of the application.
3	Incomplete	One or more candidate *physics*-based or *correlation*-based models are available that are believed to nominally reflect a zeroth-order representation of the phenomena over the parameter space of the application.
2		Significant discovery activities have been completed. At least one candidate model or correlation form is emerging.
1	Inadequate	Discovery activities are planned and funded, but no significant activities have occurred and model form is still unknown or very speculative.
0		No clue about model form, and no significant discovery activity planned or in progress.

Code Adequacy ranks current understanding of the adequacy of the code for calculating the phenomenon of the chosen PIRT element. An example of a ranking system for code adequacy follows.

CODE ADEQUACY (Example)		
Score	**Descriptor**	**Definition**
5	Adequate	Phenomena implemented in code at the current level of understanding. Extensive regression suite tests code robustness (as code is under development), and there are specific problems in the regression test suite that test the implementation of the specified phenomena. The VERTS includes elements that test the numerical correctness of the implementation. There are no outstanding (reported) bugs or issues that can undermine credibility of the proposed calculations.
4		Phenomena represented in the code at the current level of understanding. Extensive regression suite tests code robustness (as code is under development), and there are specific problems in the regression test suite that test the implementation of the specified phenomena. There are no outstanding (reported) bugs or issues that can undermine credibility of the proposed calculations.
3	Incomplete	Phenomena represented in the code at the current level of understanding. There is an extensive regression suite, but the regression suite does not specifically test the implementations of the phenomena of interest. There are no outstanding (reported) bugs or issues that can undermine credibility of the proposed calculations.
2		Phenomena represented in the code at the current level of understanding. There is an inadequate regression suite, or the regression suite does not specifically test the implementations of the phenomena of interest, or the issue/bug tracking system is inadequate to determine whether or not there are bugs or issues that can undermine credibility of the proposed calculations.
1	Inadequate	Phenomena represented in the code at less than the current level of understanding.
0		Phenomena not represented in the code or certain critical enabling capabilities are not functional.

Experimental Adequacy ranks subsequent understanding of the adequacy of the executed validation experiments for achieving validation goals associated with the phenomenon of the chosen PIRT element. An example of a ranking system for experimental adequacy follows.

EXPERIMENTAL ADEQUACY (Example)		
Score	**Descriptor**	**Definition**
4	High	All necessary measurements are quantitative and uncertainty is specified. In particular, error bars are provided for all quantitative data, with a specified interpretation of their meaning.
3		All necessary measurements are quantitative and uncertainty is partially specified (there are known uncertainties that have not been quantified).
2	Medium	Some necessary measurements are qualitative and uncertainty is not addressed. In particular, no error bars are provided for any provided data.
1	Low	All necessary measurements are qualitative and uncertainty is not addressed.
0		Experiment failed to produce required data.

Validation Adequacy ranks subsequent understanding of the success of the experimental validation activity for assessing capability of the code for modeling the phenomenon involved in the chosen PIRT element. An example of a ranking system for validation adequacy is given in the table below.

VALIDATION ADEQUACY (Example)		
Score	**Descriptor**	**Definition**
5	Adequate	Predictive capability of the model or correlation is quantified over the full parameter space of the application. There is a statistically significant database that is fully relevant to the application.
4		Predictive capability of the model or correlation is quantified over the parameter space of the database. The degree of *extrapolation* is quantified. There is a statistically significant database that is fully relevant to the application.
3	Incomplete	Statistical comparison of data and calculations that does not quantify predictive capability of the model or correlation over the parameter space of the database. The degree of extrapolation (if any) may not be quantified. The database may not be statistically significant or fully relevant to the application.
2		Ad hoc (nonstatistical) comparisons of experimental data (that may or may not be statistically significant) or data traces.
1	Inadequate	Ad hoc comparison of experiment "pictures" with prediction "pictures."
0		No significant comparisons with experiment data.

(Page Left Blank)

Appendix E: Features of Different Types of Experiments

The following table briefly summarizes various classes of experiments, identifying their features and relation to validation as discussed in Section 6. Note that the various tiers of validation experiments listed are discussed in Section 3.

Experiment Type	Example Characteristics	Relation to Validation
Phenomena Discovery	▪ Phenomena ID ▪ Unit, coupled, integral experiments ▪ Inadequate data robustness and specificity ▪ Low-consequence model comparisons ▪ Primary goal is **science**	▪ No prediction ▪ No definitive comparisons with calculations ▪ Not validation
Mathematical Model Development	▪ Math formulation and implementation ▪ Unit, coupled, integral experiments ▪ Critical model comparisons ▪ Robust and precise data ▪ Primary goal is **numerical implementation**	▪ No prediction ▪ No definitive comparisons with calculations ▪ Not validation
Tier 1 (Unit) Calibration	▪ Calibration of constitutive models ▪ Unit experiments ▪ Robust and precise data ▪ Model adjustment ▪ Primary goal is **numerical implementation**	▪ No prediction ▪ Comparisons are parameter ID, not critical ▪ Not validation
Tier 1 (Unit) Validation	▪ Unit experiments ▪ Robust and precise data ▪ Critical model comparisons ▪ Primary goal is **validation**	▪ Validation
Tier 2 (Simple Couplings) Calibration	▪ Coupled phenomena experiments ▪ Robust and precise data ▪ Model adjustment ▪ Primary goal is **numerical implementation**	▪ No prediction ▪ Comparisons are parameter ID, not critical ▪ Not validation
Tier 2 (Simple Couplings) Validation	▪ Coupled phenomena experiments ▪ Robust and precise data ▪ Critical model comparisons ▪ Primary goal is **validation**	▪ Validation
Tier 3 (Integral) Calibration	▪ Integral experiments ▪ Robust and precise data ▪ Model adjustment ▪ Primary goal is **numerical implementation**	▪ No prediction ▪ Comparisons are parameter ID, not critical ▪ Not validation
Tier 3 (Integral) Validation	▪ Integral experiments ▪ Robust and precise data ▪ Critical model comparisons ▪ Primary goal is **validation**	▪ Validation
Tier 4 (Qualification)	▪ Unit, coupled, integral ▪ Robust data, not required to be precise ▪ Primary goal is **DP acceptance**	▪ Primarily qualification and acceptance ▪ Possibly validation

(Page Left Blank)

Distribution

EXTERNAL DISTRIBUTION

M. A. Adams
Jet Propulsion Laboratory
4800 Oak Grove Drive, MS 97
Pasadena, CA 91109

M. Aivazis
Center for Advanced Computing
Research
California Institute of Technology
1200 E. California Blvd./MS 158-79
Pasadena, CA 91125

Charles E. Anderson
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78284-0510

Bilal Ayyub (2)
Department of Civil Engineering
University of Maryland
College Park, MD 20742

Ivo Babuska
Texas Institute for Computational
and Applied Mathematics
Mail Code C0200
University of Texas at Austin
Austin, TX 78712-1085

Osman Balci
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

S. L. Barson
Boeing Company
Rocketdyne Propulsion & Power
MS IB-39
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Steven Batill (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

S. Beissel
Alliant Techsystems, Inc.
600 Second St., NE
Hopkins, MN 55343

David Belk
WL/MNAA
101 W. Eglin Blvd., Suite 219
Eglin AFB, FL 32542-6810

Ted Belytschko (2)
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road
Evanston, IL 60208

James Berger
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Pavel A. Bouzinov
ADINA R&D, Inc.
71 Elton Avenue
Watertown, MA 02472

John A. Cafeo
General Motors R&D Center
Mail Code 480-106-256
30500 Mound Road
Box 9055
Warren, MI 48090-9055

James C. Cavendish
General Motors R&D Center
Mail Code 480-106-359
30500 Mound Road
Box 9055
Warren, MI 48090-9055

Chun-Hung Chen (2)
Department of Systems Engineering &
Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, VA 22030

Wei Chen
Dept. of Mechanical Engr. (M/C 251)
842 W. Taylor St.
University of Illinois at Chicago
Chicago, IL 60607-7022

Kyeongjae Cho (2)
Dept. of Mechanical Engineering
MC 4040
Stanford University
Stanford, CA 94305-4040

Thomas Chwastyk
U.S. Navel Research Lab.
Code 6304
4555 Overlook Ave., SW
Washington, DC 20375-5343

Harry Clark
Rocket Test Operations
AEDC
1103 Avenue B
Arnold AFB, TN 37389-1400

Hugh Coleman
Department of Mechanical &
Aero. Engineering
University of Alabama/Huntsville
Huntsville, AL 35899

Raymond Cosner (2)
Boeing-Phantom Works
MC S106-7126
P. O. Box 516
St. Louis, MO 63166-0516

Thomas A. Cruse
398 Shadow Place
Pagosa Springs, CO 81147-7610

P. Cuniff
U.S. Army Soldier Systems Center
Kansas Street
Natick, MA 01750-5019

Department of Energy (3)
Attn: William Reed, NA-114
Jamileh Soudah, NA-114
B. Pate, NA-114
1000 Independence Ave., SW
Washington, DC 20585

U. M. Diwekar (2)
Center for Energy and
Environmental Studies
Carnegie Mellon University
Pittsburgh, PA 15213-3890

David Dolling
Department of Aerospace Engineering
& Engineering Mechanics
University of Texas at Austin
Austin, TX 78712-1085

Robert G. Easterling
7800 Northridge NE
Albuquerque, NM 87109

Isaac Elishakoff
Dept. of Mechanical Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431-0991

Ashley Emery
Dept. of Mechanical Engineering
Box 352600
University of Washingtion
Seattle, WA 98195-2600

Scott Ferson
Applied Biomathematics
100 North Country Road
Setauket, New York 11733-1345

Joseph E. Flaherty (2)
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12181

John Fortna
ANSYS, Inc.
275 Technology Drive
Canonsburg, PA 15317

Roger Ghanem
Dept. of Civil Engineering
Johns Hopkins University
Baltimore, MD 21218

Mike Giltrud
Defense Threat Reduction Agency
DTRA/CPWS
6801 Telegraph Road
Alexandria, VA 22310-3398

James Glimm (2)
Dept. of Applied Math & Statistics
P138A
State University of New York
Stony Brook, NY 11794-3600

James Gran
SRI International
Poulter Laboratory AH253
333 Ravenswood Avenue
Menlo Park, CA 94025

Bernard Grossman (2)
Dept. of Aerospace &
Ocean Engineering
Mail Stop 0203
215 Randolph Hall
Blacksburg, VA 24061

Sami Habchi
CFD Research Corp.
Cummings Research Park
215 Wynn Drive
Huntsville, AL 35805

Raphael Haftka (2)
Dept. of Aerospace and Mechanical
Engineering and Engr. Science
P. O. Box 116250
University of Florida
Gainesville, FL 32611-6250

Achintya Haldar (2)
Dept. of Civil Engineering
& Engineering Mechanics
University of Arizona
Tucson, AZ 85721

Tim Hasselman
ACTA
2790 Skypark Dr., Suite 310
Torrance, CA 90505-5345

G. L. Havskjold
Boeing Company
Rocketdyne Propulsion & Power
MS GB-09
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

George Hazelrigg
Division of Design, Manufacturing
& Innovation
Room 508N
4201 Wilson Blvd.
Arlington, VA 22230

David Higdon
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Richard Hills (2)
College of Engineering, MSC 3449
New Mexico State University
P. O. Box 30001
Las Cruces, NM 88003

F. Owen Hoffman (2)
SENES
102 Donner Drive
Oak Ridge, TN 37830

Luc Huyse
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78284-0510

G. Ivy
Logicon R&D Associates
P.O. Box 92500
Los Angeles, CA 90009

Ralph Jones (2)
Sverdrup Tech. Inc./AEDC Group
1099 Avenue C
Arnold AFB, TN 37389-9013

Leo Kadanoff (2)
Research Institutes Building
University of Chicago
5640 South Ellis Ave.
Chicago, IL 60637

George Karniadakis (2)
Division of Applied Mathematics
Brown University
192 George St., Box F
Providence, RI 02912

Alan Karr
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

J. J. Keremes
Boeing Company
Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Hyoung-Man Kim
Boeing Company
M/S: ZC-01
502 Gemini Ave.
Houston, TX 77058

K. D. Kimsey
U.S. Army Research Laboratory
Weapons & Materials Research Directorate
AMSRL-WM-TC 309 120A
Aberdeen Proving Gd, MD 21005-5066

B. A. Kovac
Boeing Company
Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

P. Krysl
Department of Computer Science
California Institute of Technology
1200 E. California Blvd./MS 256-80
Pasadena, CA 91125

Chris Layne
AEDC
Mail Stop 6200
760 Fourth Street
Arnold AFB, TN 37389-6200

W. K. Liu (2)
Northwestern University
Dept. of Mechanical Engineering
2145 Sheridan Road
Evanston, IL 60108-3111

Robert Lust
General Motors, R&D and Planning
MC 480-106-256
30500 Mound Road
Warren, MI 48090-9055

Sankaran Mahadevan (2)
Dept. of Civil & Environmental Engineering
Vanderbilt University
Box 6077, Station B
Nashville, TN 37235

Hans Mair
Institute for Defense Analysis
Operational Evaluation Division
1801 North Beauregard Street
Alexandria, VA 22311-1772

W. McDonald
Naval Surface Warfare Center
Code 420
101 Strauss Avenue
Indian Head, MD 20640-5035

Gregory McRae (2)
Dept. of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Michael Mendenhall (2)
Nielsen Engineering & Research, Inc.
510 Clyde Ave.
Mountain View, CA 94043

Sue Minkoff (2)
Dept. of Mathematics and Statistics
University of Maryland
1000 Hilltop Circle
Baltimore, MD 21250

Max Morris (2)
Department of Statistics
Iowa State University
304A Snedecor-Hall
Ames, IW 50011-1210

Paul Muessig
Naval Air Warfare Center
Joint Accreditation Support Activity
Weapons Division, Code 418000D
1 Administration Circle
China Lake, CA 93555-6100

R. Namburu
U.S. Army Research Laboratory
AMSRL-CI-H
Aberdeen Proving Gd, MD 21005-5067

NASA/Ames Research Center (2)
Attn: Unmeel Mehta, MS T27 B-1
David Thompson, MS 269-1
Moffett Field, CA 94035-1000

NASA/Glen Research Center (2)
Attn: John Slater, MS 86-7
Chris Steffen, MS 5-11
21000 Brookpark Road
Cleveland, OH 44135

NASA/Langley Research Center (7)
Attn: Dick DeLoach, MS 236
Michael Hemsch, MS 280
Tianshu Liu, MS 238
Jim Luckring, MS 280
Joe Morrison, MS 128
Ahmed Noor, MS 369
Sharon Padula, MS 159
Hampton, VA 23681-0001

C. Needham
Applied Research Associates, Inc.
4300 San Mateo Blvd., Suite A-220
Albuquerque, NM 87110

A. Needleman
Division of Engineering, Box D
Brown University
Providence, RI 02912

Robert Nelson
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

Dick Neumann
8311 SE Millihanna Rd.
Olalla, WA 98359

Efstratios Nikolaidis (2)
MIME Dept.
4035 Nitschke Hall
University of Toledo
Toledo, OH 43606-3390

D. L. O'Connor
Boeing Company
Rocketdyne Propulsion & Power
MS AC-15
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Tinsley Oden (2)
Texas Institute for Computational
and Applied Mathematics
Mail Code C0200
University of Texas at Austin
Austin, TX 78712-1085

Michael Ortiz (2)
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Dale Pace
Applied Physics Laboratory
Johns Hopkins University
111000 Johns Hopkins Road
Laurel, MD 20723-6099

Alex Pang
Computer Science Department
University of California
Santa Cruz, CA 95064

Allan Pifko
2 George Court
Melville, NY 11747

Cary Presser (2)
Process Measurements Div.
National Institute of Standards
and Technology
Bldg. 221, Room B312
Gaithersburg, MD 20899

P. Radovitzky
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

W. Rafaniello
DOW Chemical Company
1776 Building
Midland, MI 48674

Chris Rahaim (2)
Parks College of Engineering
3450 Lindell Blvd.
St. Louis University
St. Louis, MO 63103

Pradeep Raj (2)
Computational Fluid Dynamics
Lockheed Martin Aeronautical Sys.
86 South Cobb Drive
Marietta, GA 30063-0685

J. N. Reddy
Dept. of Mechanical Engineering
Texas A&M University
ENPH Building, Room 210
College Station, TX 77843-3123

John Renaud (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

E. Repetto
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Patrick J. Roache
1108 Mesa Loop NW
Los Lunas, NM 87031

A. J. Rosakis
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Tim Ross (2)
Dept. of Civil Engineering
University of New Mexico
Albuquerque, NM 87131

J. Sacks
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Sunil Saigal (2)
Carnegie Mellon University
Department of Civil and
Environmental Engineering
Pittsburgh, PA 15213

Len Schwer
Schwer Engineering & Consulting
6122 Aaron Court
Windsor, CA 95492

Paul Senseny
Factory Mutual Research Corporation
1151 Boston-Providence Turnpike
P.O. Box 9102
Norwood, MA 02062

E. Sevin
Logicon RDA, Inc.
1782 Kenton Circle
Lyndhurst, OH 44124

Mark Shephard (2)
Rensselaer Polytechnic Institute
Scientific Computation Research Center
Troy, NY 12180-3950

Tom I-P. Shih
Dept. of Mechanical Engineering
2452 Engineering Building
East Lansing, MI 48824-1226

T. P. Shivananda
Bldg. SB2/Rm. 1011
TRW/Ballistic Missiles Division
P. O. Box 1310
San Bernardino, CA 92402-1310

Y.-C. Shu
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Don Simons
Logicon
222 W. Sixth St.
P.O. Box 471
San Pedro, CA 90733-0471

M. M. Sindir
Boeing Company
Rocketdyne Propulsion & Power
MS GB-11
P. O. Box 7922
6633 Canoga Avenue
Canoga Park, CA 91309-7922

Ashok Singhal
CFD Research Corp.
Cummings Research Park
215 Wynn Drive
Huntsville, AL 35805

R. Singleton
Engineering Sciences Directorate
Army Research Office
4300 S. Miami Blvd.
P.O. Box 1221
Research Triangle Park, NC 27709-2211

W. E. Snowden
DARPA
7120 Laketree Drive
Fairfax Station, VA 22039

Bill Spencer (2)
Dept. of Civil Engineering
and Geological Sciences
University of Notre Dame
Notre Dame, IN 46556-0767

Fred Stern
Professor Mechanical Engineering
Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City Iowa 52242

D. E. Stevenson (2)
Computer Science Department
Clemson University
442 Edwards Hall, Box 341906
Clemson, SC 29631-1906

Tim Swafford
Sverdrup Tech. Inc./AEDC Group
1099 Avenue C
Arnold AFB, TN 37389-9013

Kenneth Tatum
Sverdrup Tech. Inc./AEDC Group
740 Fourth Ave.
Arnold AFB, TN 37389-6001

Ben Thacker
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78284-0510

Fulvio Tonon (2)
Geology and Geophysics Dept.
East Room 719
University of Utah
135 South 1460
Salt Lake City, UT 84112

Robert W. Walters (2)
Aerospace and Ocean Engineering
Virginia Tech
215 Randolph Hall, MS 203
Blacksburg, VA 24061-0203

Leonard Wesley
Intellex Inc.
5932 Killarney Circle
San Jose, CA 95138

Justin Y-T Wu
Applied Research Associates
Probabilistic Engineering
811 Spring Forest Rd.
Raleigh, NC 27609

Ren-Jye Yang
Ford Research Laboratory
MD2115-SRL
P.O.Box 2053
Dearborn, MI 4812

Simone Youngblood (2)
DOD/DMSO
Technical Director for VV&A
1901 N. Beauregard St., Suite 504
Alexandria, VA 22311

M. A. Zikry
North Carolina State University
Mechanical & Aerospace Engineering
2412 Broughton Hall, Box 7910
Raleigh, NC 27695

FOREIGN DISTRIBUTION:

Yakov Ben-Haim (2)
Department of Mechanical Engineering
Technion-Israel Institute of Technology
Haifa 32000
ISRAEL

Gert de Cooman (2)
Universiteit Gent
Onderzoeksgroep, SYSTeMS
Technologiepark - Zwijnaarde 9
9052 Zwijnaarde
BELGIUM

Graham de Vahl Davis
CFD Research Laboratory
University of NSW
Sydney, NSW 2052
AUSTRIALIA

Luis Eca (2)
Instituto Superior Tecnico
Department of Mechanical Engineering
Av. Rovisco Pais
1096 Lisboa CODEX
PORTUGAL

Charles Hirsch (2)
Department of Fluid Mechanics
Vrije Universiteit Brussel
Pleinlaan, 2
B-1050 Brussels
BELGIUM

Igor Kozin (2)
Systems Analysis Department
Riso National Laboratory
P. O. Box 49
DK-4000 Roskilde
DENMARK

K. Papoulia
Inst. Eng. Seismology & Earthquake Engineering
P.O. Box 53, Finikas GR-55105
Thessaloniki
GREECE

Max Ruppert
UniBw Munich - BauV 2.2
Inst. Engng.Mech. & Struct.Mech.
D - 85577 Neuibiberg
GERMANY

Lev Utkin
Institute of Statistics
Munich University
Ludwigstr. 33
80539, Munich
GERMANY

Malcolm Wallace
National Engineering Laboratory
East Kilbride
Glasgow
G75 0QU
UNITED KINGDOM

Peter Walley
6 Jewel Close
Port Douglas
Queensland 4871
AUSTRALIA

Department of Energy Laboratories

Los Alamos National Laboratory (47)
Mail Station 5000
P.O. Box 1663
Los Alamos, NM 87545
Attn: Peter Adams, MS B220
Mark C. Anderson, MS D411
Thomas Bement, MS F600
Robert Benjamin, MS K557
J. M. Booker, MS P946
Terrence Bott, MS K557
D. Cagliostro, MS F645
David Crane, MS P946
John F. Davis, MS B295
Helen S. Deaven, MS B295
Barbara DeVolder, MS F663
Scott Doebling, MS P946
S. Eisenhawer, MS K557
Dawn Flicker, MS F664
George T. Gray, MS G755
Ken Hanson, MS B250
R. Henninger, MS D413
Brad Holian, MS B268
Kathleen Holian, MS B295
Darryl Holm, MS B284
James Hyman, MS B284
Cliff Joslyn, MS B265
James Kamm, MS D413
Jeanette Lagrange, MS D445
S. Keller-McNulty, MS F600
Joseph Kindel, MS B259
Ken Koch, MS F652
Douglas Kothe, MS B250
Len Margolin, MS D413
Harry Martz, MS F600
Mike McKay, MS F600
Kelly McLenithan, MS F664
Mark P. Miller, MS P946
John D. Morrison, MS F602
Karen I. Pao, MS B256
M. Peterson-Schnell, MS B295
Doug Post, MS F661 X-DO
William Rider, MS D413
Tom Seed, MS F663
David Sharp, MS B213
Richard N. Silver, MS D429
Ronald E. Smith, MS J576
Christine Treml, MS H851
David Tubbs, MS B220
Daniel Weeks, MS B295
Morgan White, MS F663
Alyson G. Wilson, MS F600

Lawrence Livermore National Laboratory (18)
7000 East Ave.
P.O. Box 808
Livermore, CA 94550
Attn: T. F. Adams, MS L-095
Steven Ashby, MS L-561
John Bolstad, MS L-023
Peter N. Brown, MS L-561
T. Scott Carman, MS L-031
R. Christensen, MS L-160
Evi Dube, MS L-095
Richard Klein, MS L-023
Roger Logan, MS L-125
C. F. McMillan, MS L-098
C. Mailhiot, MS L-055
J. F. McEnerney, MS L-023
M. J. Murphy, MS L-282
Daniel Nikkel, MS L-342
Cynthia Nitta, MS L-096
Peter Raboin, MS L-125
Peter Terrill, MS L-125
Charles Tong, MS L-560

Argonne National Laboratory
Attn: Paul Hovland
MCS Division
Bldg. 221, Rm. C-236
9700 S. Cass Ave.
Argonne, IL 60439

SANDIA INTERNAL

1	MS 1152	1642	M. L. Kiefer
1	MS 1186	1674	R. J. Lawrence
1	MS 0525	1734	P. V. Plunkett
1	MS 0525	1734	R. B. Heath
1	MS 0525	1734	S. D. Wix
1	MS 0429	2100	J. S. Rottler
1	MS 0427	2100	R. C. Hartwig
1	MS 0453	2101	H. J. Abeyta
1	MS 0427	2104	P. A. Sena
1	MS 0427	2104	F. F. Dean
1	MS 0482	2109	R. A. Paulsen
1	MS 0447	2111	J. O. Harrison
1	MS 0447	2111	P. Davis
1	MS 0447	2111	P. D. Hoover
1	MS 0479	2113	W. J. Tedeschi
1	MS 0479	2113	M. H. Abt
1	MS 0481	2114	M. A. Rosenthal
1	MS 0481	2114	W. C. Moffatt
1	MS 0481	2131	K. D. Meeks
1	MS 0482	2131	R. S. Baty

1	MS 0482	2131	K. Ortiz
1	MS 0509	2300	M. W. Callahan
1	MS 0634	2951	K. V. Chavez
1	MS 0769	5800	D. S. Miyoshi
1	MS 0759	5845	I. V. Waddoups
1	MS 0775	5862	R. V. Matalucci
1	MS 0751	6117	L. S. Costin
1	MS 0708	6214	P. S. Veers
1	MS 0490	6252	J. A. Cooper
1	MS 0736	6400	T. E. Blejwas
1	MS 0744	6400	D. A. Powers
1	MS 0747	6410	A. L. Camp
1	MS 0747	6410	G. D. Wyss
1	MS 0748	6413	D. G. Robinson
1	MS 0748	6413	R. D. Waters
1	MS 1137	6534	S. M. DeLand
1	MS 1137	6534	G. D. Valdez
1	MS 1137	6536	L. M. Claussen
1	MS 1137	6536	G. K. Froehlich
1	MS 1137	6536	A. L. Hodges
1	MS 1138	6536	M. T. McCornack
1	MS 1137	6536	S. V. Romero
1	MS 0716	6804	P. G. Kaplan
1	MS 1395	6820	D. K. Belasich
1	MS 1395	6820	M. J. Chavez
1	MS 1395	6820	J. G. Miller
1	MS 1395	6821	M. K. Knowles
1	MS 1395	6821	J. W. Garner
1	MS 1395	6821	E. R. Giambalvo
1	MS 1395	6821	T. Hadgu
1	MS 1395	6821	S. C. James
1	MS 1395	6821	J. S. Stein
1	MS 0779	6840	M. G. Marietta
1	MS 0779	6840	P. Vaughn
1	MS 0779	6849	J. C. Helton
1	MS 0779	6849	L. C. Sanchez
1	MS 0778	6851	G. E. Barr
1	MS 0778	6851	R. J. MacKinnon
1	MS 0778	6851	P. N. Swift
1	MS 0779	6852	B. W. Arnold
1	MS 0779	6852	R. P. Rechard
1	MS 9007	8200	D. R. Henson
1	MS 9202	8205	R. M. Zurn
1	MS 9005	8240	E. T. Cull, Jr.
1	MS 9051	8351	C. A. Kennedy
1	MS 9017	8700	T. M. Dyer
1	MS 9404	8725	J. R. Garcia
1	MS 9404	8725	W. A. Kawahara
1	MS 9405	8726	R. E. Jones
1	MS 9161	8726	P. A. Klein
1	MS 9161	8726	E. P. Chen
1	MS 9405	8726	R. A. Regueiro
1	MS 9042	8727	J. J. Dike
1	MS 9042	8727	J. L. Handrock
1	MS 9042	8727	A. R. Ortega
1	MS 9042	8728	C. D. Moen
1	MS 9003	8900	K. E. Washington
1	MS 9012	8920	P. E. Nielan
1	MS 9003	8950	C. M. Hartwig
1	MS 9217	8950	M. L. Martinez-Canales
1	MS 9217	8950	P. D. Hough
1	MS 1110	8950	L. J. Lehoucq
1	MS 9217	8950	K. R. Long
1	MS 9217	8950	J. C. Meza
1	MS 0841	9100	T. C. Bickel
1	MS 0841	9100	C. W. Peterson
1	MS 0826	9100	D. K. Gartling
1	MS 0824	9110	A. C. Ratzel
1	MS 0834	9112	M. R. Prairie
1	MS 0834	9112	S. J. Beresh
1	MS 0826	9113	W. Hermina
1	MS 0834	9114	J. E. Johannes
1	MS 0834	9114	K. S. Chen
1	MS 0834	9114	R. R. Rao
1	MS 0834	9114	P. R. Schunk
1	MS 0825	9115	W. H. Rutledge
1	MS 0825	9115	F. G. Blottner
1	MS 0825	9115	B. Hassan
1	MS 0825	9115	D. W. Kuntz
1	MS 0825	9115	M. A. McWherter-Payne
1	MS 0825	9115	J. L. Payne
1	MS 0825	9115	D. L. Potter
1	MS 0825	9115	C. J. Roy
1	MS 0825	9115	W. P. Wolfe
1	MS 0836	9116	E. S. Hertel
1	MS 0836	9116	D. Dobranich
1	MS 0836	9116	R. E. Hogan
1	MS 0836	9116	C. Romero
1	MS 0836	9117	R. O. Griffith
1	MS 0836	9117	R. J. Buss
1	MS 0847	9120	H. S. Morgan
1	MS 0555	9122	M. S. Garrett
1	MS 0893	9123	R. M. Brannon
1	MS 0847	9124	J. M. Redmond
1	MS 0847	9124	K. F. Alvin
1	MS 0553	9124	T. G. Carne
1	MS 0847	9124	R. V. Field
1	MS 0553	9124	T. Simmermacher
1	MS 0553	9124	D. O. Smallwood
1	MS 0847	9124	S. F. Wojtkiewicz
1	MS 0557	9125	T. J. Baca
1	MS 0557	9125	C. C. O'Gorman
1	MS 0847	9126	R. A. May
1	MS 0847	9126	S. N. Burchett
1	MS 0847	9126	K. E. Metzinger
1	MS 0824	9130	J. L. Moya
1	MS 1135	9132	L. A. Gritzo
1	MS 0555	9132	J. T. Nakos
1	MS 0836	9132	S. R. Tieszen
20	MS 0828	9133	M. Pilch

1	MS 0828	9133	A. R. Black
1	MS 0828	9133	B. F. Blackwell
1	MS 0828	9133	K. J. Dowding
20	MS 0828	9133	W. L. Oberkampf
1	MS 0557	9133	T. L. Paez
1	MS 0828	9133	V. J. Romero
1	MS 0828	9133	M. P. Sherman
1	MS 0557	9133	A. Urbina
1	MS 0847	9133	W. R. Witkowski
1	MS 1135	9134	S. Heffelfinger
1	MS 0835	9140	J. M. McGlaun
1	MS 0835	9141	S. N. Kempka
1	MS 0835	9141	R. J. Cochran
1	MS 0835	9142	J. S. Peery
1	MS 0847	9142	S. W. Attaway
1	MS 0847	9142	M. L. Blanford
1	MS 0847	9142	M. W. Heinstein
1	MS 0847	9142	S. W. Key
1	MS 0847	9142	G. M. Reese
1	MS 0827	9143	J. D. Zepper
1	MS 0827	9143	K. M. Aragon
1	MS 0827	9143	H. C. Edwards
1	MS 0847	9143	G. D. Sjaardema
1	MS 0827	9143	J. R. Stewart
1	MS 0321	9200	W. J. Camp
1	MS 0318	9200	G. S. Davidson
1	MS 1111	9209	S. J. Plimpton
1	MS 1110	9211	D. E. Womble
1	MS 1110	9211	R. Carr
1	MS 1110	9211	S. Y. Chakerian
1	MS 0847	9211	M. S. Eldred
1	MS 0847	9211	A. A. Giunta
1	MS 1110	9211	W. E. Hart
1	MS 1110	9211	A. Johnson
1	MS 1110	9211	V. J. Leung
1	MS 1110	9211	C. A. Phillips
1	MS 0847	9211	J. R. Red-Horse
20	MS 0819	9211	T. G. Trucano
1	MS 0847	9211	B. G. vanBloemen Waanders
1	MS 1109	9212	R. J. Pryor
1	MS 1110	9214	J. DeLaurentis
1	MS 1110	9214	R. B. Lehoucq
1	MS 0310	9220	R. W. Leland
1	MS 0310	9220	J. A. Ang
1	MS 1110	9223	N. D. Pundit
1	MS 1110	9224	D. W. Doerfler
1	MS 1111	9226	B. A.Hendrickson
1	MS 0847	9226	P. Knupp
1	MS 0318	9227	P. D. Heermann
1	MS 0822	9227	C. F. Diegert
1	MS 0318	9230	P. Yarrington
1	MS 0819	9231	E. A. Boucheron
1	MS 0819	9231	K H. Brown
1	MS 0819	9231	K. G. Budge
1	MS 0819	9231	S. P. Burns
1	MS 0819	9231	D. E. Carroll
1	MS 0819	9231	M. A. Christon
1	MS 0819	9231	R. R. Drake
1	MS 0819	9231	A. C. Robinson
1	MS 0819	9231	M. K. Wong
1	MS 0820	9232	P. F. Chavez
1	MS 0820	9232	M. E. Kipp
1	MS 0820	9232	S. A. Silling
1	MS 0819	9232	R. M. Summers
1	MS 0820	9232	P. A. Taylor
1	MS 0820	9232	J. R. Weatherby
1	MS 0316	9233	S. S. Dosanjh
1	MS 0316	9233	D. R. Gardner
1	MS 0316	9233	S. A. Hutchinson
1	MS 1111	9233	A. G. Salinger
1	MS 1111	9233	J. N. Shadid
1	MS 0316	9235	J. B. Aidun
1	MS 0316	9235	H. P. Hjalmarson
1	MS 0660	9519	D. S. Eaton
1	MS 0660	9519	M. A. Ellis
1	MS 0139	9900	M. O. Vahle
1	MS 0139	9904	R. K. Thomas
1	MS 0139	9905	S. E. Lott
1	MS 0428	12300	D. D. Carlson
1	MS 0428	12301	V. J. Johnson
1	MS 0421	12323	J. M. Sjulin
1	MS 0829	12323	B. M. Rutherford
1	MS 0829	12323	F. W. Spencer
1	MS 0638	12326	M. A. Blackledge
1	MS 0638	12326	D. E. Peercy
1	MS 0638	12326	D. L. Knirk
1	MS 0492	12332	D. R. Olson
1	MS 0405	12333	T. R. Jones
1	MS 0405	12333	M. P. Bohn
1	MS 0405	12333	S. E. Camp
1	MS 0434	12334	R. J. Breeding
1	MS 0829	12335	K. V. Diegert
1	MS 1170	15310	R. D. Skocypec
1	MS 1176	15312	R. M. Cranwell
1	MS 1176	15312	D. J. Anderson
1	MS 1176	15312	J. E. Campbell
1	MS 1176	15312	L. P. Swiler
1	MS 1179	15340	J. R. Lee
1	MS 1179	15341	L. Lorence
1	MS 1164	15400	J. L. McDowell
1	MS 9018	8945-1	Central Technical Files
2	MS 0899	9616	Technical Library
1	MS 0612	9612	Review & Approval Desk For DOE/OSTI