ASME V&V 10-2006

REAFFIRMED 2016

Guide for Verification and Validation in Computational Solid Mechanics

AN AMERICAN NATIONAL STANDARD

The American Society of Mechanical Engineers

Intentionally left blank

ASME V&V 10-2006

Guide for Verification and Validation in Computational Solid Mechanics

AN AMERICAN NATIONAL STANDARD

The American Society of Mechanical Engineers

Three Park Avenue • New York, NY 10016

Date of Issuance: December 29, 2006

This Guide will be revised when the Society approves the issuance of a new edition. There will be no addenda issued to ASME V&V 10-2006.

ASME issues written replies to inquiries concerning interpretations of technical aspects of this document. Periodically certain actions of the ASME V&V Committees will be published as cases. Cases and interpretations are published on the ASME Web site under the Committee Pages at http://cstools.asme.org as they are issued.

This Guide was developed under procedures accredited as meeting the criteria for American National Standards. The Standards Committee that approved the Guide was balanced to assure that individuals from competent and concerned interests have had an opportunity to participate. The proposed code or standard was made available for public review and comment that provides an opportunity for additional public input from industry, academia, regulatory agencies, and the public-at-large.

ASME does not “approve,” “rate,” or “endorse” any item, construction, proprietary device, or activity.

ASME does not take any position with respect to the validity of any patent rights asserted in connection with any items mentioned in this document, and does not undertake to insure anyone utilizing a standard against liability for infringement of any applicable letters patent, nor assumes any such liability. Users are expressly advised that determination of the validity of any such patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

Participation by federal agency representative(s) or person(s) affiliated with industry is not to be interpreted as government or industry endorsement of this code or standard.

ASME accepts responsibility for only those interpretations of this document issued in accordance with the established ASME procedures and policies, which precludes the issuance of interpretations by individuals.

The American Society of Mechanical Engineers
Three Park Avenue, New York, NY 10016-5990

CONTENTS

Figures

Table

Mandatory Appendices

FOREWORD

Since the mid-1960s, computer simulations have come to dominate engineering mechanics analysis for all but the simplest problems. With today's increasing reliance on complicated simulations using computers, it is necessary to use a systematic program of verification and validation (V&V) to ensure the accuracy of these simulations. This document is intended to describe such a program.

The concept of systematic V&V is not a new one. The software development community has long recognized the need for a quality assurance program for scientific and engineering software. The Institute of Electrical and Electronic Engineers, along with other organizations, has adopted guidelines and standards for software quality assurance (SQA) appropriate for developers. SQA guidelines, while necessary, are not sufficient to cover the nuances of computational physics and engineering or the vast array of problems to which end-users apply the codes. To fill this gap, the concept of application-specific V&V was developed.

Application-specific V&V has been the focus of attention for several groups in scientific and engineering communities since the mid-1990s. The Department of Defense's Defense Modeling and Simulation Office (DMSO) produced recommended practices suitable for large-scale simulations. However, the DMSO guidelines generally do not focus on the details of first-principles–based computational physics and engineering directly. For the area of computational fluid dynamics (CFD), the American Institute of Aeronautics and Astronautics (AIAA) produced the first V&V guidelines for detailed, first-principle analyses.

Recognizing the need for a similar set of guidelines for computational solid mechanics (CSM), members of the CSM community formed a committee under the auspices of the United States Association for Computational Mechanics in 1999. The American Society of Mechanical Engineers (ASME) Board on Performance Test Codes (PTC) granted the committee official status in 2001 and designated it as the PTC 60 Committee on Verification and Validation in Computational Solid Mechanics. The PTC 60 committee undertook the task of writing these guidelines. Its membership consists of solid mechanics analysts, experimenters, code developers, and managers from industry, government, and academia. Industrial representation includes the aerospace/defense, commercial aviation, automotive, bioengineering, and software development industries. The Department of Defense, the Department of Energy, and the Federal Aviation Administration represent the government.

Early discussions within PTC 60 revealed an immediate need for a common language and process definition for V&V appropriate for CSM analysts, as well as their managers and customers. This document describes the semantics of V&V and defines the process of performing V&V in a manner that facilitates communication and understanding among the various performers and stakeholders. Because the terms and concepts of V&V are numerous and complex, it was decided to publish this overview document first, to be followed in the future by detailed treatments of how to perform V&V for specific applications.

Several experts in the field of CSM who were not part of PTC 60 reviewed a draft of this document and offered many helpful suggestions. The final version of this document was approved by PTC 60 on May 11, 2006 and was approved and adopted by the American National Standards Institute on November 3, 2006.

PERFORMANCE TEST CODE COMMITTEE 60
Verification and Validation in Computation Solid Mechanics

(The following is the roster of the Committee at the time of approval of this Guide.)

OFFICERS

L. E. Schwer, *Chair*
H. U. Mair, *Vice Chair*
R. L. Crane, *Secretary*

COMMITTEE PERSONNEL

M. C. Anderson, Los Alamos National Laboratory
J. M. Burns, *Liaison Member,* Burns Engineering
J. A. Cafeo, General Motors Corporation
R. L. Crane, The American Society of Mechanical Engineers
S. W. Doebling, Los Alamos National Laboratory
J. H. Fortna, ANSYS
M. E. Giltrud, Defense Threat Reduction Agency
J. K. Gran, SRI International
T. K. Hasselman, Acta, Inc.
H. M. Kim, Boeing
R. W. Logan, Lawrence Livermore National Laboratory
H. U. Mair, Institute for Defense Analyses
A. K. Noor, Old Dominion University
W. L. Oberkampf, Sandia National Laboratories
J. T. Oden, University of Texas
D. K. Pace, Consultant
T. Paez, Sandia National Laboratories
A. B. Pifko, Consultant
L. Proctor, MSC Software
J. N. Reddy, Texas A & M University
P. J. Roache, Consultant
L. E. Schwer, Schwer Engineering
P. E. Senseny, Consultant
M. S. Shephard, Rensselaer Polytechnic Institute
D. A. Simons, Northrop Grumman
W. G. Steele, Jr., *Liaison Member,* Mississippi State University
B. H. Thacker, Southwest Research Institute
T. G. Trucano, Sandia National Laboratories
R. J. Yang, Ford Motor Co.
Y. Zhao, St. Jude Medical

The Committee acknowledges the contributions of the following individuals associated with the Committee:

G. Gray, *Former Member,* Los Alamos National Laboratory
F. Hemez, *Alternate,* Los Alamos National Laboratory
R. Lust, *Alternate,* General Motors Corp.
C. Nitta, *Alternate,* Lawrence Livermore National Laboratory
V. Romero, *Alternate,* Sandia National Laboratories

PERFORMANCE TEST CODES STANDARDS COMMITTEE

OFFICERS

J. G. Yost, *Chair*
J. R. Friedman, *Vice Chair*
J. Karian, *Secretary*

COMMITTEE PERSONNEL

P. G. Albert
R. P. Allen
J. M. Burns
W. C. Campbell
M. J. Dooley
A. J. Egli
J. R. Friedman
G. J. Gerber
P. M. Gerhart
T. C. Heil
R. A. Johnson
D. R. Keyser
S. J. Korellis
M. P. McHale
P. M. McHale
J. W. Milton
S. P. Nuspl
A. L. Plumley
R. R. Priestley
J. A. Rabensteine
J. W. Siegmund
J. A. Silvaggio
W. G. Steele
J. C. Westcott
W. C. Wood
J. G. Yost

HONORARY MEMBERS

W. O. Hays
F. H. Light

MEMBERS EMERITI

R. L. Bannister
R. Jorgensen
G. H. Mittendorf
R. E. Sommerlad

CORRESPONDENCE WITH THE PTC 60 COMMITTEE

General. ASME Codes are developed and maintained with the intent to represent the consensus of concerned interests. As such, users of this Guide may interact with the Committee by requesting interpretations, proposing revisions, and attending Committee meetings. Correspondence should be addressed to:

Secretary, PTC 60 Committee
The American Society of Mechanical Engineers
Three Park Avenue
New York, NY 10016-5990

Proposing Revisions. Revisions are made periodically to the Guide to incorporate changes that appear necessary or desirable, as demonstrated by the experience gained from the application of the Guide. Approved revisions will be published periodically.

The Committee welcomes proposals for revisions to this Guide. Such proposals should be as specific as possible, citing the paragraph number(s), the proposed wording, and a detailed description of the reasons for the proposal, including any pertinent documentation.

Interpretations. Upon request, the PTC 60 Committee will render an interpretation of any requirement of the Guide. Interpretations can only be rendered in response to a written request sent to the Secretary of the PTC 60 Committee.

The request for interpretation should be clear and unambiguous. It is further recommended that the inquirer submit his/her request in the following format:

Subject: Cite the applicable paragraph number(s) and the topic of the inquiry.

Edition: Cite the applicable edition of the Guide for which the interpretation is being requested.

Question: Phrase the question as a request for an interpretation of a specific requirement suitable for general understanding and use, not as a request for an approval of a proprietary design or situation. The inquirer may also include any plans or drawings which are necessary to explain the question; however, they should not contain proprietary names or information.

Requests that are not in this format will be rewritten in this format by the Committee prior to being answered, which may inadvertently change the intent of the original request.

ASME procedures provide for reconsideration of any interpretation when or if additional information that might affect an interpretation is available. Further, persons aggrieved by an interpretation may appeal to the cognizant ASME Committee or Subcommittee. ASME does not "approve," "certify," "rate," or "endorse" any item, construction, proprietary device, or activity.

Attending Committee Meetings. The PTC 60 Committee regularly holds meetings, which are open to the public. Persons wishing to attend any meeting should contact the Secretary of the PTC 60 Committee.

PREFACE

This document provides general guidance for implementing verification and validation (V&V) of computational models for complex systems in solid mechanics. The guidance is based on the following key principles:

(a) Verification (addressing programming errors and estimating numerical errors) must precede validation (assessing a model's predictive capability by comparing calculations with experiments).

(b) The need for validation experiments and the associated accuracy requirements for computational model predictions are based on the intended use of the model and should be established as part of V&V activities.

(c) Validation of a complex system should be pursued in a hierarchical fashion from the component level to the system level.

(d) Validation is specific to a particular computational model for a particular intended use.

(e) Simulation results and experimental data must have an assessment of uncertainty to be meaningful.

Although the state of the art of V&V does not yet lend itself to writing a step-by-step performance code/standard, this guide provides the computational solid mechanics (CSM) community with a common language and conceptual framework to enable managers and practitioners of V&V to better assess and enhance the credibility of CSM models. Implementation of a range of V&V activities is discussed, including model development for complex systems, verification of numerical solutions to governing equations, attributes of validation experiments, accuracy requirements, and quantification of uncertainties. Remaining issues for further development of a V&V protocol are identified.

GUIDE FOR VERIFICATION AND VALIDATION IN COMPUTATIONAL SOLID MECHANICS

1 EXECUTIVE SUMMARY

Program managers need assurance that computational models of engineered systems are sufficiently accurate to support programmatic decisions. This document provides the technical community — engineers, scientists, and program managers — with guidelines for assessing the credibility of computational solid mechanics (CSM) models.

Verification and validation (V&V) are the processes by which evidence is generated, and credibility is thereby established, that computer models have adequate accuracy and level of detail for their intended use. Definitions of V&V differ among segments of the practicing community. The PTC 60 committee has chosen definitions consistent with those published by the Defense Modeling and Simulation Office of the Department of Defense (DoD) [1] and by the American Institute of Aeronautics and Astronautics (AIAA) in their 1998 Guide for the Verification and Validation of Computational Fluid Dynamics [2], which the present American Society of Mechanical Engineers (ASME) document builds upon. Verification assesses the numerical accuracy of a computational model, irrespective of the physics being modeled. Both code verification (addressing errors in the software) and calculation verification (estimating numerical errors due to under-resolved discrete representations of the mathematical model) are addressed. Validation assesses the degree to which the computational model is an accurate representation of the physics being modeled. It is based on comparisons between numerical simulations and relevant experimental data. Validation must assess the predictive capability of the model in the physical realm of interest, and it must address uncertainties that arise from both experimental and computational procedures.

Although the state of the art of V&V does not yet lend itself to writing a step-by-step performance code/standard, the guidance provided here will enable managers and practitioners of V&V to better assess and enhance the credibility of CSM models. The PTC 60 Committee recognizes that program needs and resources vary and that the application of the guidance provided in this document to specific cases must accommodate specific budget and risk considerations. The scope of this document is to explain the principles of V&V so that practitioners can better appreciate and understand how decisions made during V&V can impact their ability to assess and enhance the credibility of CSM models.

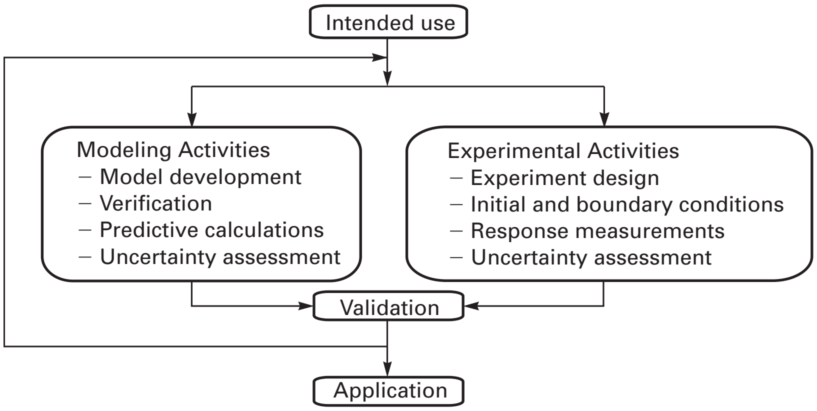
As suggested by Fig. 1, the V&V processes begin with a statement of the intended use of the model so that the relevant physics are included in both the model and the experiments performed to validate the model. Modeling activities and experimental activities are guided by the response features of interest and the accuracy requirements for the intended use. Experimental outcomes for component-level, subsystem-level, or system-level tests should, whenever possible, be provided to modelers only after the numerical simulations for them have been performed with a verified model. For a particular application, the V&V processes end with acceptable agreement between model predictions and experimental outcomes after accounting for uncertainties in both, allowing application of the model for the intended use. If the agreement between model and experiment is not acceptable, the processes of V&V are repeated by updating the model and performing additional experiments.

Finally, the importance of documentation in all of the V&V activities should be emphasized. In addition to preserving the compiled evidence of V&V, documentation records the justifications for important decisions, such as selecting primary response features and setting accuracy requirements. Documentation thereby supports the primary objective of V&V: to build confidence in the predictive capability of computational models. Documentation also provides a historical record of the V&V processes, provides traceability during an engineering audit, and captures experience useful in mentoring others.

2 INTRODUCTION

CSM is playing an increasingly important role in the design and performance assessment of engineered systems. Automobiles, aircraft, and weapon systems are examples of engineered systems that have become more and more reliant on computational models and simulation results to predict their performance, safety, and reliability. Although important decisions are made based on CSM, the credibility (or trustworthiness) of these models and simulation results is oftentimes not questioned by the general public, the technologists who design and build the systems, or the decision makers

Fig. 1 Elements of V&V



who commission their manufacture and govern their use.

What is the basis for this trust? Both the public and decision makers tend to trust graphical and numerical presentations of computational results that are plausible and that make sense to them. This trust is also founded on faith in the knowledge and abilities of the scientists and engineers who develop, exercise, and interpret the models. Those responsible for the computational models and simulations on which society depends so heavily are, therefore, keepers of the public trust with an abiding responsibility for ensuring the veracity of their simulation results.

Scientists and engineers are aware that the computational models they develop and use are approximations of reality and that these models are subject to the limitations of available data, physical theory, mathematical representations, and numerical solutions. Indeed, a fundamental approximation in solid mechanics is that the nonhomogeneous microstructure of materials can be modeled as a mathematical homogeneous continuum. Further approximations are commonly made, such as assuming the sections of a beam to remain plane during bending. Additionally, characterization of complex material behavior subject to extreme conditions is a significant approximation that must be made. The use of these approximations, along with their attendant mathematical formulations and numerical solution techniques, has proved to be a convenient and acceptably accurate approach for predicting the behavior of many engineered structures.

Analysts always need to ensure that their approximations of reality are appropriate for answering specific questions about engineered systems. Primarily, an analyst should strive to establish that the accuracy of a computational model is adequate for the model's intended use. The required accuracy is related to the ability of a simulation to correctly answer a quantitative question — a question that requires a numerical value as opposed to one that requires a simple "yes" or "no" response. Accuracy requirements vary from problem to problem and can be influenced by public perception and economic considerations, as well as by engineering judgment.

The truth of a scientific theory, or of a prediction made from the theory, cannot be proven in the sense of deductive logic. However, scientific theories and subsequent predictions can and should be tested for trustworthiness by the accumulation of evidence. The evidence collected, corroborative or not, should be organized systematically through the processes of computational model V&V. V&V address the issue of trustworthiness by providing a logical framework for accumulating and evaluating evidence and for assessing the credibility of simulation results to answer specific questions about engineered systems.

2.1 Purpose and Scope

The purpose of this document is to provide the computational solid and structural mechanics community with a common language, a conceptual framework, and general guidance for implementing the processes of computational model V&V. To this end, the reader will find a glossary of terms, figures illustrating the recommended overall approach to V&V activities, and discussions of factors that should be considered in developing and executing a V&V program. In creating this document, the PTC 60 committee benefited from the earlier contributions to the field of V&V by other groups, especially Reference 2 as well as References 3 and 4. Although the state of the art of V&V does not yet lend itself to writing a step-by-step performance code/standard, the guidance provided here will enable managers and practitioners of V&V to better assess and enhance the credibility of CSM models.

To maximize the value to the engineering community, the PTC 60 committee chose to write from the perspective of V&V for high-consequence computational predictions of complex engineering systems. However, the guidance provided here is also appropriate for simpler applications, recognizing that smaller budgets and lower risks will reduce the scope of the V&V effort. Also, while the concepts and terminology presented here are applicable to all applied mechanics, the focus is on CSM.

2.2 General Concepts of V&V

To avoid confusion, the terms "code," "model," and "simulation results" are defined here as follows:

code: the computer implementation of algorithms developed to facilitate the formulation and approximate solution of a class of problems.

model: the conceptual, mathematical, and numerical representations of the physical phenomena needed to represent specific real-world conditions and scenarios. Thus, the model includes the geometrical representation, governing equations, boundary and initial conditions, loadings, constitutive models and related material parameters, spatial and temporal approximations, and numerical solution algorithms.

simulation results: the output generated by the computational model.

The terms "verification" and "validation" have been used interchangeably in casual conversation as synonyms for the collection of corroborative evidence. Definitions that have been adopted in this document are largely consistent with those published by the DoD [1] and the AIAA [2].

validation: the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

verification: the process of determining that a computational model accurately represents the underlying mathematical model and its solution.

In essence, "verification" is the process of gathering evidence to establish that the computational implementation of the mathematical model and its associated solution are correct. "Validation," on the other hand, is the process of compiling evidence to establish that the appropriate mathematical models were chosen to answer the questions of interest by comparing simulation results with experimental data.

Readers might find it helpful at this point to look at definitions of the terms used throughout this document. The glossary (Mandatory Appendix I) defines terms that form part of the shared language for V&V as used herein.

The objective (i.e., desired outcome) of V&V is to validate a model for its intended use. From the perspective of the model builder, the model is considered validated for its intended use once its predetermined requirements for demonstration of accuracy and predictive capability have been fulfilled. From the perspective of the decision maker or stakeholder, the intended use also defines the limitations imposed on the applicability of the model. An example of an intended use would be to predict the response of a particular make and model of automobile in frontal impacts against a wall at speeds up to 30 mph. Validation might consist of predicting the compaction of the front end and the acceleration of the occupant compartment to within 20% for tests at 10, 20, and 30 mph. The validated model could then be used to predict the same response features at any speed up to 30 mph. However, it would not be validated for other makes or models of automobiles, for higher speeds, or for rear-end or side collisions.

A detailed specification of the model's intended use should include a definition of the accuracy criteria by which the model's predictive capability will be assessed. The accuracy criteria should be driven by application (i.e., intended use) requirements. For instance, in the previous example, 20% accuracy is based on consideration of how the predictions will be used. Although accuracy criteria and other model requirements may have to be changed before, during, or after validation assessments of the entire system, it is best to specify validation and accuracy criteria prior to initiating model-development and experimental activities in order to establish a basis for defining "how good is good enough?"

The recommended approach to conducting model V&V emphasizes the need to develop a plan for conducting the V&V program. For complex, high-consequence engineered systems, the initial planning will preferably be done by a team of experts. The V&V plan should be prepared before any validation experiments are performed. The plan, at a minimum, should include a detailed specification of the intended use of the model to guide the V&V effort; a detailed description of the full physical system, including the behavior of the system's parts both in isolation and in combination; and a list of the experiments that need to be performed. The plan may also provide details about the approach that will be taken to verify the model, as well as information related to such program factors as schedule and cost. Key considerations in developing the V&V plan are discussed in para. 2.5, following presentation of the V&V processes.

2.3 Approach to Modeling Complex Systems

Many real-world physical systems that would be the subject of model V&V are inherently complex. To address this complexity and prepare a detailed description of the full system, it is helpful to recognize that the real-world physical system being modeled is hierarchical in nature. As illustrated in Fig. 2, the hardware of a physical system is typically composed of assemblies, each of which consists of two or more subassemblies; a subassembly, in turn, consists of individual components.

The top-level reality of interest in Fig. 2 can be viewed as any level of a real physical system. For example, it could be a complete automobile, or it could be the drive

Fig. 2 Hierarchical Structure of Physical Systems

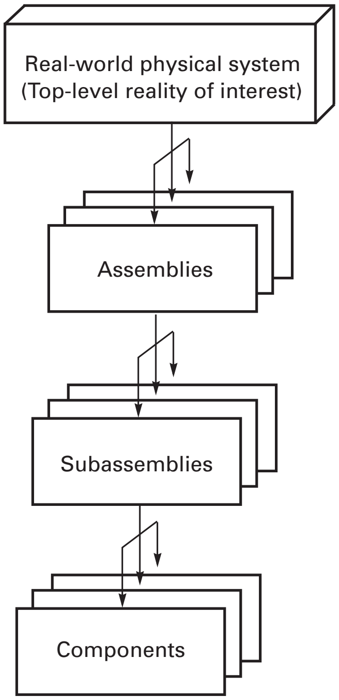
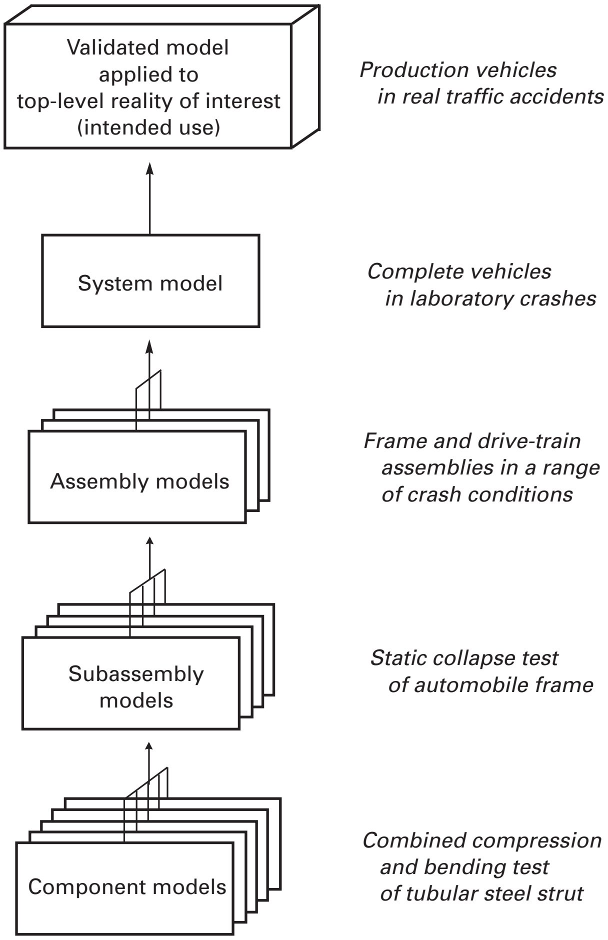


Fig. 3 Example of Bottom-Up Approach to V&V


Validated model applied to top-level reality of interest (intended use)
Production vehicles in real traffic accidents
System model
Complete vehicles in laboratory crashes
Assembly models
Frame and drive-train assemblies in a range of crash conditions
Subassembly models
Static collapse test of automobile frame
Component models
Combined compression and bending test of tubular steel strut

train of an automobile. If an automobile is the top-level reality of interest, it might be composed of assemblies such as the drive train, the frame, the body, and the interior. Considering the drive train as an assembly, it might be composed of subassemblies like the engine, the transmission, the drive shaft, and the rear differential. Similarly, a subassembly such as the engine might contain components like the engine block and the radiator. In terms of V&V, the requirements on the model for the top-level reality of interest, as well as for all lower levels, depend on the intended use for which the model is being developed.

2.4 Bottom-Up Approach to V&V

A top-down decomposition of the physical system into its hardware constituents, as discussed above, serves as the basis for developing a model of this system. However, the recommended approach to V&V is to develop such a hierarchy and then work from the bottom up to identify and describe the physical phenomena at each level of the hierarchy that must be accurately simulated with the model, beginning at the lowest tier in the hierarchy (i.e., the component level). Component phenomena could include fundamental behavior such as deformation, natural frequencies, and buckling loads. The bottom-up approach recognizes that some of the physical responses of components may be representative of a single physical phenomenon, while at levels of the hierarchy above that of components, interaction effects that are not exhibited by the individual components are likely, such as effects of frictional interfaces and joints. For example, a model of a subassembly consisting of a welded automobile frame could introduce behavior that is not present when individual struts are modeled separately.

Building a model from the bottom up will result in a multitiered set of individual models (a system-level model and its embedded submodels) and form the basis for defining validation experiments that need to be conducted at each tier of the hierarchy to ensure that the constituent models at the particular tier function appropriately. Models for components, subassemblies, and assemblies that have been validated previously can and should be reused if the response mechanisms they have been shown to exhibit, and the predictive accuracy they have demonstrated, clearly meet the requirements of the new system. Fig. 3 depicts an overview of the bottom-up approach to validation. The left side of the figure identifies the models that would be constructed at each tier, and the right side of the figure provides examples of the types of experiments and predictions that might be performed at the respective tiers. In this example, validation of the system model will be achieved, by consensus of the program experts, if the responses of complete vehicles in laboratory crashes are successfully predicted. It is common that the highest-tier validation experiments are either special cases of the expected operating conditions or idealized versions of the real-world system. It is important to complete V&V with computations and experiments at the system level to assess whether the bottom-up approach adequately considered complex nonlinear interactions at all levels of the hierarchy (i.e., that the appropriate hierarchical decomposition was used).

It may be tempting to perform validation of system models directly from data taken from tests of the complete system without new or archived validation at lower levels in the hierarchy. This can be problematic for a large number of components or if the subsystem models contain complex connections or interfaces, energy dissipation mechanisms, or highly nonlinear behavior. If there is poor agreement between the simulation results and the experiment, it is often difficult, if not impossible, to isolate which subsystem model is responsible for the discrepancy. Even if good agreement between calculation and experiment is observed, it is still possible that the model quality could be poor because of error cancellation among the subsystem models. Therefore, a better strategy is to conduct a sequence of experiments that builds confidence in the model's ability to produce accurate simulations at multiple levels in the hierarchy.

2.5 V&V Activities and Products

Once the elements of the physical system's hierarchy (whether one or many tiers) have been defined and prioritized, a systematic approach can be followed for quantifying confidence in model predictions through the logical combination of hierarchical model building, focused laboratory and field experimentation, and uncertainty quantification. This process is discussed below.

Figure 4 identifies the activities and products in the recommended V&V approach for CSM. The activities are denoted by simple text, such as "mathematical modeling" and "physical modeling"; the products of these activities are highlighted in rounded boxes (e.g., the mathematical model is the product of the mathematical modeling activity). Modelers follow the left branch to develop, exercise, and evaluate the model. Experimenters follow the right branch to obtain the relevant experimental data via physical testing. Modelers and experimenters collaborate in developing the conceptual model, conducting preliminary calculations for the design of experiments, and specifying initial and boundary conditions for calculations for validation.

The process shown in Fig. 4 is repeated for each member of every tier of the hierarchy in the system decomposition exercise discussed previously, starting at the component level and progressing upward through the system level. Thus, the reality of interest is an individual subsystem each time this approach is followed. Ultimately, the reality of interest at the top of Fig. 4 would be the complete system. However, in the bottom-up approach, both preliminary conceptual model development and V&V planning for all levels in the hierarchy, especially the system level, are performed before the main validation activities for components, subassemblies, and assemblies begin.

Abstraction of the reality of interest into the conceptual model requires identifying the domain of interest, important physical processes and assumptions, and system-response quantities of interest. The abstraction essentially produces the modeling approach based on these considerations. It is also intimately connected to the development of the overall V&V plan that establishes the validation requirements, including the types of experiments to be performed and the required level of agreement between the experimental outcomes and the simulation outcomes. Thus, this activity is typically iterative and involves modelers, experimenters, and decision makers.

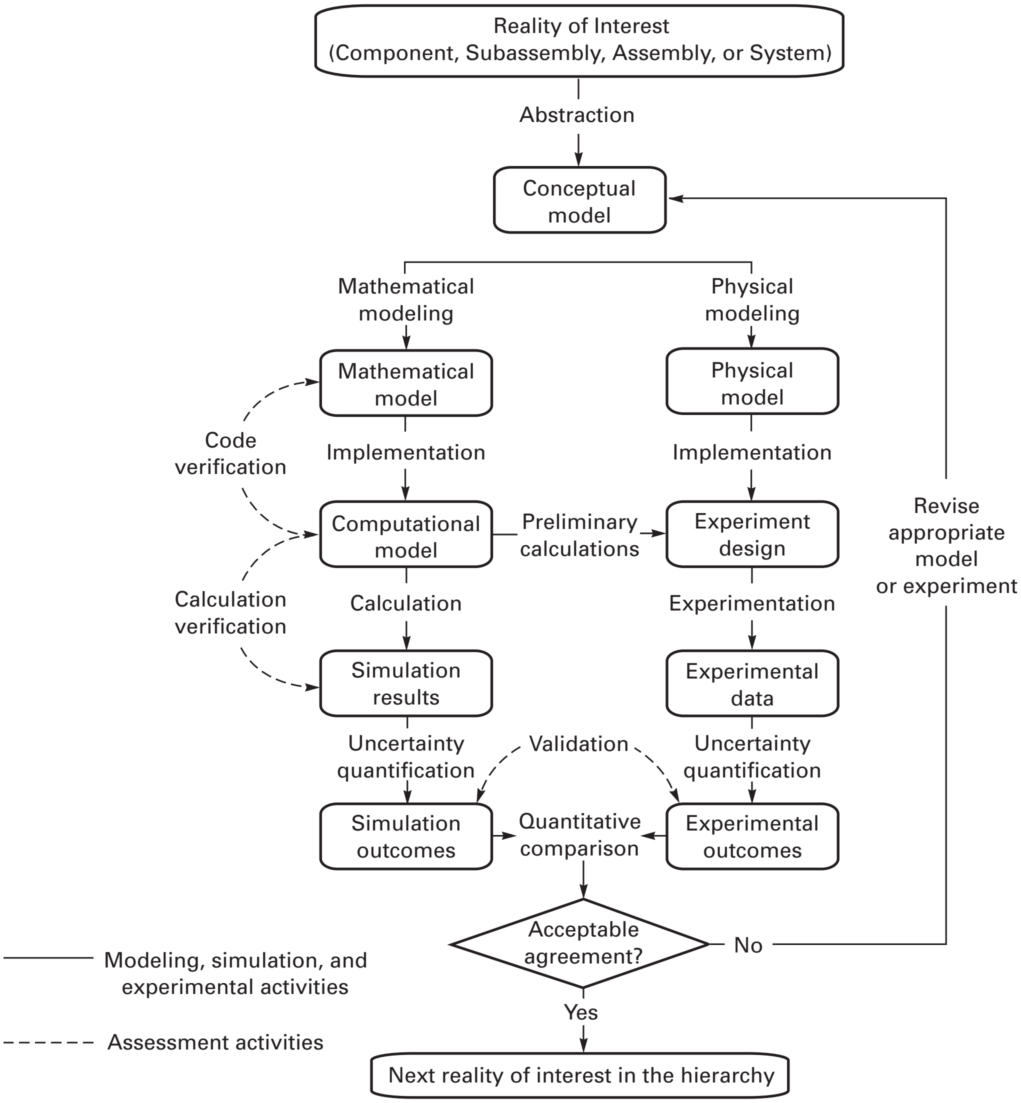
2.5.1 The Modeling Branch.

In the mathematical modeling activity, the modeler constructs a mathematical interpretation of the conceptual model. The resulting mathematical model is a set of equations and modeling data that describe physical reality, including the geometric description, governing equations, initial and boundary conditions, constitutive equations, and external forces.

During the subsequent implementation activity, the modeler develops the computational model, which is the software implementation on a specific computing platform of the equations developed in the mathematical model, usually in the form of numerical discretization, solution algorithms, and convergence criteria. The computational model includes numerical procedures, such as finite element or finite difference, for solving the equations prescribed in the mathematical model with specific computer software.

In the assessment activity of code verification, the modeler uses the computational model on a set of problems with known solutions. These problems typically have much simpler geometry, loads, and boundary conditions than the validation problems, to identify and eliminate algorithmic and programming errors. Then, in the subsequent activity of calculation verification, the version of the computational model to be used for validation problems (i.e., with the geometries, loads, and boundary conditions typical of those problems) is used for identifying sufficient mesh resolution to produce an adequate solution tolerance, including the effects of finite precision arithmetic. Calculation verification

Fig. 4 V&V Activities and Products



yields a quantitative estimate of the numerical precision and discretization accuracy for calculations made with the computational model for the validation experiments.

In the calculation activity, the modeler runs the computational model to generate the simulation results for validation experiments. The simulation results can also be postprocessed to generate response features for comparison with experimental data. A feature could be as simple as the maximum response for all times at a specific location in the object being tested, or it could be as complex as a fast Fourier transform of the complete response history at that location.

In the subsequent uncertainty quantification activity, the modeler should quantify the uncertainties in the simulation results that are due to the inherent variability in model parameters or to lack of knowledge of the parameters or the model form. The results of the parameter and model-form uncertainty quantification should be combined with those of the calculation verification to yield an overall uncertainty estimate associated with simulation results. Features of interest extracted from simulation results and estimates of uncertainty combine to form the simulation outcomes that are used for comparison with the experimental outcomes.

2.5.2 The Experimental Branch. In the first two activities of the right branch of Fig. 4, validation experiments are conceived via the physical modeling activity and designed as part of the implementation activity. The purpose of validation experiments is to provide information needed to assess the accuracy of the mathematical model; therefore, all assumptions should be understood, well defined, and controlled. To assist with experiment design, preliminary calculations (including sensitivity and uncertainty analyses) are recommended, for example, to identify the most effective locations and types of measurements needed from the experiment. These data should include not only response measurements, but also measurements needed to define model inputs and model input uncertainties associated with loading, initial conditions, boundary conditions, etc. The modeler and the experimenter should continue to work together, so that they are both continually aware of assumptions in the models or the experiments. By observing the preparations for the experiment, for example, the modeler will frequently detect incorrect assumptions in the model. However, experimental results should not be given to the modeler to preclude inadvertent or intentional tuning of the model to match experimental results.

The experimentation activity involves the collection of raw data from various instruments used in the experiment, such as strain and pressure gauges and high-speed cameras, and the generation of processed data such as time integrals, averages, or the determination of velocity from high-speed video. As necessary, the experimental data can be transformed into experimental features that are more useful for direct comparison with simulation outcomes. Repeat experiments are generally required to quantify uncertainty due to lack of repeatability and inherent variability.

The experimenter then performs uncertainty quantification to quantify the effects of various sources of uncertainty on the experimental data. Among these sources are measurement error, design tolerances, manufacturing and assembly variations, unit-to-unit fabrication differences, and variations in performance characteristics of experimental apparatuses. Experimental outcomes, which are the product of this uncertainty quantification activity, will typically take the form of experimental data plus uncertainty bounds as a function of time or load.

2.5.3 Obtaining Agreement. Once experimental outcomes and simulation outcomes for the actual test conditions have been generated, the modeler and experimenter perform the validation assessment activity by comparing these two sets of outcomes.

The metrics for comparing experimental outcomes and simulation outcomes as well as the criteria for acceptable agreement will have been specified during the formulation of the V&V plan. The degree to which the model accurately predicts the data from system-level validation experiments is the essential component of the overall assessment of the model's predictive capability. Note, however, that the diamond symbol denoting "acceptable agreement" at the bottom of Fig. 4 provides an objective decision point for initiating improvements in the conceptual, mathematical, and computational models and in the experimental designs.

The block at the bottom of Fig. 4 denotes that the process repeats for the next submodel to be validated, either at the same tier or at the next higher tier of the hierarchy. Thus, as V&V is performed, the results of the component-level activities are propagated to the next higher tier of the hierarchy, and so on up to the full-system level.

2.6 Development of the V&V Plan

As mentioned previously, a V&V program should be thoughtfully planned before the major activities in model development and experimentation are initiated. In particular, it is essential to define the requirements for system-level validation in the V&V plan.

2.6.1 Validation Testing. In many cases, the most difficult part of V&V planning is to establish the relationship between validation experiments and the reality of interest. That is, for what set of cases should the model have to demonstrate predictive capability so that the user will have sufficient confidence that the model can predict the reality of interest with the required accuracy? In some cases, this is a matter of interpolation or perhaps minor extrapolation. In other cases, however, it may not be possible either to test the complete system or to test over the full range of the reality of interest, such as for a model whose intended use is to predict the response of a high-rise building to an earthquake. Still, by a consensus of experts, a plan must always be developed that defines the set of conditions for which the system model's predictive capability is to be demonstrated in order to be accepted for its intended use.

2.6.2 Selection of Response Features. Complex physical systems and the model simulations that predict their behavior encompass an enormous array of response features. And because only a limited number of measurements can be made in validation experiments, it is important to identify the features of interest before the experiments are designed. Selecting which response features to measure and compare with predictions should first be driven by application requirements. At the system level, this may require product safety or reliability parameters to be defined in engineering terms. For example, occupant injury in automobile crashes may be related to occupant–compartment accelerations and protrusions, and thus those features should be measured and predicted. The appropriate response features of assemblies, subassemblies, and components depend on how their responses affect the critical features of the system response. Specifications should also be made for the metrics used for comparisons of outcomes, such as root-mean-square (RMS) differences of simulation and experimental acceleration histories.

2.6.3 Accuracy Requirements. The accuracy requirements for predicting the response features of interest with the system-level model are based on the intended use and may rely on engineering judgment or a formal risk analysis. Specification of accuracy requirements allows the "acceptable agreement" question to be answered quantitatively. Only with accuracy requirements can the decision be made about whether to accept or revise a model. Without accuracy requirements, the question "how good is good enough?" cannot be answered.

System-level accuracy requirements are used to establish accuracy requirements for each submodel in the V&V hierarchy. These requirements should be established such that models for assemblies, subassemblies, and components are refined at least to the degree required to meet the accuracy goal of the system-level model. A sensitivity analysis of the complete system can be used to estimate the contribution of each model; the estimated contributions can then be used to establish commensurate accuracy requirements. It is reasonable to expect that the accuracy requirement for component behavior will be more stringent than the accuracy requirements for the complete system due to the simpler nature of problems at the component level and the compounding effect of propagating inaccuracy up through the hierarchy. For example, a 10% accuracy requirement might be established for a model that calculates the axial buckling strength of a tubular steel strut in order to achieve 20% accuracy of the collapse strength of a frame made of many such components.

2.7 Documentation of V&V

It is important to document both the results and the rationale of V&V not only for the current intended use, but also for potential future uses. V&V allow a knowledge base to be built from the various levels in the hierarchy and later reused in subsequent applications. For example, in many applications, derivative or closely related product designs are used in the development of future designs. If a thorough execution and documentation of hierarchical V&V has been performed for the model of the basic design, many of the hierarchical elements for V&V of the model for the derivative design might be reusable. In this way, the value of investment in hierarchical V&V can be leveraged to reduce V&V costs for future projects. Documentation also provides the basis for possible limitations on reuse and thus prevents unjustifiable extrapolations. The V&V documentation should be comprehensive, self-contained, retrievable, and citable.

2.8 Overview of Subsequent Sections

Section 2 has outlined the basic principles and characteristics of a careful and logical approach to implementing model V&V for CSM. The guidelines for accomplishing the various activities in V&V form the contents of sections 3 through 5. Model development activities are the focus of section 3. In section 4, the two assessment activities of code verification and calculation verification are described. Section 5 discusses the experimental and assessment activities involved in validating a model. The concluding remarks in section 6 identify issues that need to be addressed so that V&V for CSM can evolve into a more robust and quantitative methodology. The concluding remarks are followed by mandatory appendices, which comprise a glossary of V&V terms (Mandatory Appendix I), nomenclature (Mandatory Appendix II), and a bibliography (Mandatory Appendix III).

3 MODEL DEVELOPMENT

This section describes the activities involved in developing the computational model, starting with the formulation of the conceptual and mathematical models, then revising these models during V&V, and, finally, quantifying the uncertainty in the resulting model. The description of the model development activities begins with the assumption that the reality of interest, the intended use of the model, the response features of interest, and the accuracy requirements have been clearly defined. However, there will be some interplay between the development of the conceptual model and the V&V plan.

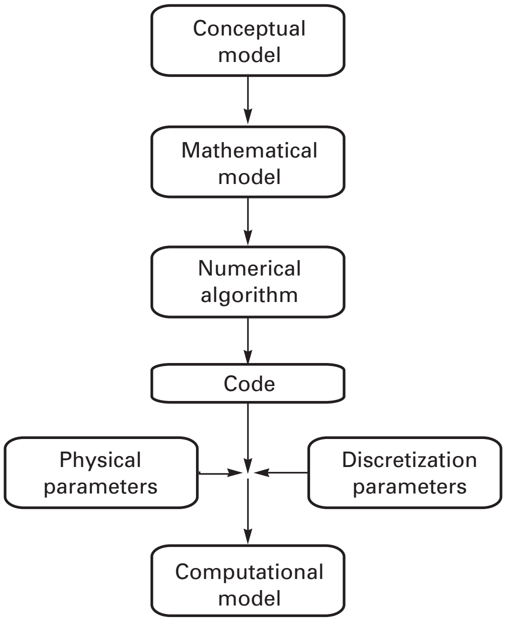
In general, the system model (conceptual to computational) is built up from subassembly, assembly, and component models, as illustrated in Fig. 2. At the highest level, the "reality of interest" within Figs. 3 and 4 will be the real-world system under the intended range of realistic operating conditions; the corresponding "intended use" of the model is to predict system behavior for cases that cannot, or will not, be tested.

Figure 5 illustrates the path from a conceptual model to a computational model. An example of a conceptual model is a classical Bernoulli–Euler beam with the assumptions of elastic response and plane sections. This conceptual model can be described with differential calculus to produce a mathematical model. The equations can be solved by various numerical algorithms, but typically in CSM they would be solved using the finite element method. The numerical algorithm is programmed into a software package, here called a "code." With the specification of physical and discretization parameters, the computational model is created.

3.1 Conceptual Model

The conceptual model is defined as the idealized representation of the solid mechanics behavior of the reality of interest. This model should therefore include those mechanisms that impact the key mechanical and physical processes that will be of interest for the intended use of the model. The activity of conceptual model development involves formulating a mechanics-based representation of the reality of interest that is amenable to mathematical and computational modeling, that includes the appropriate level of detail, and that is expected to produce results with adequate accuracy for

Fig. 5 Path From Conceptual Model to Computational Model



the intended use. Essentially, it is defining the modeling approach.

The formulation of the conceptual model is important to the overall model-development process because many fundamental assumptions are made that influence interpretation of the simulation results. These assumptions include the determination of how many separate parts or components will be included in the model, the approach to modeling the material behavior, the elimination of unimportant detail features in the geometry, and the selection of interface and boundary types (e.g., fixed, pinned, contact, friction, etc.). If an important mechanical phenomenon is omitted from the conceptual model, the resulting simulations might not be adequate for the intended use of the model.

An essential step in developing the conceptual model is to identify which physical processes in the reality of interest are anticipated initially to have significant effects on the system's response. Likewise, it is important to identify which physical processes do not have a significant effect and to note that such mechanics will be ignored in the conceptual model. Identifying the essential physical processes will help to ensure that the computational model sufficiently represents the mechanics involved and does not waste computational effort modeling physical effects that do not affect the responses of interest. Development of the conceptual model also requires knowledge of the range of operating environments that are relevant to the model's intended use. The environments affect choices in the modeling, such as whether to include plasticity or thermal softening.

Table 1 PIRT Example

Phenomenon	Type of Phenomenon	Importance to Response of Interest	Level of Confidence in Model
A	Interface	High	Medium
B	Plasticity	Medium	High
C	Loads	Medium	Low
D	Fracture	Low	Low

Response features of interest are the characteristics of the response of the physical system that the computational model has to predict for the intended use. They could include characteristics such as the maximum tensile stress in bolts, the peak acceleration of the center of a floor, the average value of pressure in a chamber, the deflection of the center of a glass window, the modal frequencies of a radio tower, or the strain energy release rate at the tip of a fracture. Knowledge of the features of interest is important in the conceptual modeling activity because interest in certain features may influence decisions that are made during the mathematical and computational modeling activities. For example, if the deflections of a particular part are of interest, the compliance of materials surrounding that part probably should not be neglected.

During development of the conceptual model, the best tools available for identification of the key physical processes are engineering expertise and judgment. Thorough documentation of the rationale for what is included in — or excluded from — the conceptual model is an important part of proper model development. Note that once the computational model has been developed, a sensitivity analysis can be used to investigate the importance of a physical process to the response of the system (see para. 3.5).

Constructing a Phenomena Identification and Ranking Table (PIRT) can be a useful exercise for identifying the key physical processes [5]. The PIRT is both a process and a product. The exercise involves gathering a diverse group of subject-matter experts together to rank the physical processes according to their importance to the system responses of interest. The product is the table itself, which presents a summarized list of the physical phenomena, along with a ranking (e.g., high, medium, low) of the importance of each phenomenon to the system responses of interest. Sample entries in a PIRT are shown in Table 1. The PIRT can be used either to construct a conceptual model (starting from scratch) or to prioritize the conceptual model of a large general-purpose code that may have the ability to model hundreds of phenomena, only a subset of which are relevant to the subject model.

In addition, the PIRT could also include at this stage of model development a qualitative judgment regarding the ability of either existing computational models or to-be-developed computational models to describe the physical processes accurately (last column in Table 1). This information is useful to help prioritize which physical processes will be investigated experimentally during validation (i.e., this is part of the interplay between the development of the conceptual model and the development of the V&V plan). For the example in Table 1, phenomenon B would have a low priority for validation because it can already be modeled with high confidence. Similarly, phenomenon D would have a low priority because of its low importance to the model response of interest.

3.2 Mathematical Model

The development of the mathematical model consists of specifying the mathematical descriptions of the mechanics represented in the conceptual model. In the mathematical model, principles of mechanics, the material behavior, interface properties, loads, and boundary conditions are cast into equations and mathematical statements. For example, if the property of an interface between two bodies is to be described with Coulomb friction, the mathematical model would be $\tau = \mu \sigma$, where τ is the shear stress, μ is the Coulomb friction coefficient, and σ is the normal stress.

The specification of the mathematical model then allows the model input parameters to be defined. The model input parameters describe the various user-specified inputs to the model, such as material constants, applied loads, and the Coulomb friction coefficient in the above example. The domain of interest can then be expressed in terms of these parameters. For example, if the application domain specifies a range of applied loads, a specific parameter (or set of parameters) in the mathematical model can be used to define that range of loads.

3.3 Computational Model

The computational model is the numerical implementation of the mathematical model that will be solved on a computer to yield the computational predictions (simulation results) of the system response. As defined herein, the computational model includes the type and degree of spatial discretization of the geometry (e.g., into finite elements), the temporal discretization of the governing equations, the solution algorithms to be used to solve the governing equations, and the iterative convergence criteria for the numerical solutions. With this inclusive definition, models employing solution-adaptive mesh-generation methods are defined by their adaptive control parameters.

The computational model can be simple or complicated, and it can employ in-house or commercial finite-element software to develop and solve the numerical equations. An analyst is often tempted to jump directly from a geometric description of the reality of interest to the development of a computational mesh, especially given the availability of highly automated preprocessing software. Meshing, however, is not modeling. The analyst must understand the underlying conceptual model and mathematical model in order to understand the effects on the model outcomes that are caused by the assumptions and mathematical simplifications inherent in the computational model. Without this understanding, it is difficult to know whether the computational model is inadequate or inappropriate for the intended use. For example, the analyst must consider the type of boundary conditions to be imposed in buckling problems, because buckling results are sensitive to the end conditions used in the model.

3.4 Model Revisions

Commonly, at some stage of validation, the modeler will find that the computational model needs revisions to achieve the desired accuracy or to account for new requirements. In a general sense, there are two classes of possible revisions to the mathematical and computational models. The first class of revisions covers updates to parameters in the mathematical or computational model that are determined by calibrating the computational model to experimental data (e.g., apparent material parameters, modal damping coefficients for linear vibration, or friction coefficients for a mechanical interface). The second class of revisions covers changes to the form of the mathematical or conceptual model to improve the description of the mechanics of interest so that better agreement with the reference experimental data can be achieved. The two classes of revisions are discussed below.

3.4.1 Updates to Model Parameters by Calibration. Revision by parametric model calibration is extensively used in the field of linear structural dynamics to bring computational predictions into better agreement with measured response quantities such as modal frequencies and mode shapes. This revision process is commonly known as "model updating," "model tuning," "parameter calibration," and "parameter estimation." The process allows the most common sources of modeling (and experimental) difficulties in linear structural dynamics — compliance in joints, energy loss/damping, unmeasured excitations, uncertain boundary conditions — to be represented as simple mechanical models and calibrated so that the global response of the computational model is in agreement with the experimental data.

Parametric model calibration, however, determines only the model's fitting ability, not its predictive capability. A model calibrated to experimental data may not yield accurate predictions over the range of its intended use. This means that the model should not be used as a calibration framework for some uncertain parameters if these parameters can be evaluated in an independent test. *Data used for model calibration must remain independent of data used to assess the predictive capability of the model.*

The type of experiment used to determine the values of unknown or uncertain model input parameters is generally referred to as a "calibration experiment." A calibration experiment is distinct from a validation experiment. The purpose of a calibration experiment is to generate values or quantified probability distributions for model input parameters under specific types of experimental conditions. For example, an optimization approach may be used to determine the parameter values using a computational model of the calibration experiment and the measured data from the calibration experiment. In contrast to calibration experiments, validation experiments are designed and performed to provide an independent, objective assessment of the predictive capabilities of the computational model.

It is a reality of modeling, given cost and schedule constraints, that model calibration is often performed after an initial validation assessment has been made and acceptable agreement (as indicated in Fig. 4) has not been achieved. That is, the modeler finds a set of parameter values that provides acceptable agreement with the validation test data, but only after failing to achieve that agreement with a prediction. Unfortunately, to then assess predictive capability (outside the now updated domain of the validation referent data), subsequent validation against other independent experiments is still necessary. Any revisions to the parameter values after V&V are applied signifies new model-development activity, triggering a repetition of some model V&V.

3.4.2 Updates to Model Form. The second class of model revisions consists of changes to the form of the conceptual model and, in turn, the mathematical model and the computational model. Typically, the need to revise the model form is observed during the quantitative comparison activity, when some characteristics in the response of the structure are not consistent with the corresponding characteristics of the model output, and the differences are not attributable to reasonable uncertainties in the model parameters.

Many common types of deficiencies in model form can be responsible for inaccurate simulation results: two-dimensional models that cannot represent three-dimensional response effects; inappropriate form for representation of material behavior; assumptions about contacting surfaces being tied when in reality a gap develops between the parts; assumptions that two parts do not move relative to one another when in reality they do, resulting in development of significant friction forces; assumed rigid boundary conditions that turn out to have significant compliance, etc. It is important to look for possible violation of the assumptions of the form of the mathematical model when reconciling the measured data with the results of the computational simulation. As with parameter calibration, any revisions to the model after V&V are applied signifies new model-development activity, triggering a repetition of some model V&V.

3.5 Sensitivity Analysis

One way besides intuition and experience to identify important phenomena is to perform a sensitivity analysis using the computational model. Sensitivity analysis is the general process of discovering the effects of model input parameters on the response features of interest using techniques such as analysis of variance (ANOVA) [6]. When performed before the computational model is validated (but not before it is verified), a sensitivity analysis can provide important insight into the characteristics of that computational model and can assist in the design of experiments as part of the PIRT process. Model sensitivities, however, must eventually be subject to the same scrutiny of V&V as the main parameters of interest. As with engineering judgment or even the initial PIRT prioritization, unvalidated model sensitivities may be wrong in magnitude or even in sign. Thus, model sensitivity analysis should be revisited after model revision.

Local sensitivity analysis is used to determine the character of the response features with respect to the input parameters in a local region of the parameter space (i.e., in the vicinity of a single point). Finite difference techniques or adjoint methods are used to determine the local gradients at points in the design space. Global sensitivity analysis is concerned with some type of average behavior of the response features over a large domain of the parameters and is often used to select a subset of the parameters for detailed local sensitivity analysis.

3.6 Uncertainty Quantification for Simulations

Validation for computational mechanics models must take into account the uncertainties associated with both simulation results and experimental data. The uncertainties associated with experimental data are discussed in section 5. Throughout the modeling process (left branch of Fig. 4), and especially during the uncertainty quantification activity, all significant sources of uncertainty in model simulations must be identified and treated to quantify their effects on predictions made with the model. It is useful to categorize uncertainties as being either irreducible or reducible.

Irreducible uncertainty (also called "aleatory uncertainty") refers to inherent variations in the physical system being modeled. This type of uncertainty always exists and is an intrinsic property of the system. Examples of irreducible uncertainty are variations in geometry, material properties, loading environment, and assembly procedures. The inherent variability in model parameters is typically characterized by performing replicate component-level tests that cover the range of conditions over which the individual parameters will be exercised in the intended use of the model. If no component-level validation testing is performed, estimates of the inherent variability in model parameters should be based on prior experience and engineering judgment. However, even the most complete set of test information will not eliminate irreducible uncertainty — it can only be better quantified, for example, by determining a parameter's mean value, distribution, and distribution form (e.g., normal, uniform, log-normal).

Using probabilistic analysis, inherent variability can be propagated through the simulation to establish an expected variability in the simulation output quantities. Sampling-based propagation methods, such as Monte Carlo and Latin Hypercube, are straightforward techniques for propagating variability [7]. Sampling-based methods draw samples from the input parameter populations, evaluate the deterministic model using these samples, and then build a distribution of the appropriate response quantities. Well-known sensitivity-based methods include the first-order reliability method (FORM) [8], advanced mean value (AMV) [9], and adaptive importance sampling (AIS) [10].

Reducible uncertainty (also called "epistemic uncertainty") refers to deficiencies that result from a lack of complete information or knowledge. Two important sources of reducible uncertainty are statistical uncertainty and model form uncertainty. Statistical uncertainty arises from the use of limited samples. For example, if the mean value of a material property is calculated with only two or three measurements of the material property, then the mean value will contain statistical uncertainty, which can be reduced by considering additional measurements of the material property. Model form uncertainty refers to the uncertainty associated with modeling assumptions, such as a constant parameter assumption (regardless of its assigned numerical value) in the partial differential equations (PDEs). In other words, a parameter in an equation in the computational model could be defined as having a constant value, whereas in reality the value of the parameter varies with time, temperature, or position. In general, model form uncertainty is extremely difficult to quantify, but some innovative approaches to this problem have been developed [11, 12].

3.7 Documentation of Model Development Activities

It is important that model development activities be documented to facilitate reuse of the model. The documentation should explain the rationale for model development (e.g., modeling assumptions) and describe the conceptual, mathematical, and computational models. The description of the mathematical model should include assumptions about the mechanics of interest and the sources of information for the model parameters. The description of the computational model should include discretization assumptions, computational parameters, and other parameters of interest.

4 VERIFICATION

The process of verification assesses the fidelity of the computational model to the mathematical model. The mathematical model is commonly a set of PDEs and the associated boundary conditions, initial conditions, and constitutive equations. The computational model is the numerical implementation of the mathematical model, usually in the form of numerical discretization, solution algorithms, and convergence criteria. Verification assessments consider issues related to numerical analysis, software quality engineering (SQE), programming errors in the computer code, and numerical error estimation. Verification precedes validation, which assesses the predictive capability of the computational model by comparisons with experimental data.

Verification is composed of two fundamental activities: code verification and calculation verification. Code verification is the assessment activity for ensuring, to the degree necessary, that there are no programming errors in a computer code and that the numerical algorithms for solving the discrete equations yield accurate solutions with respect to the true solutions of the PDEs. Calculation verification is the assessment activity for estimating the numerical solution errors that are present in every simulation result; examples include temporal and spatial discretization error, iterative error, and round-off error. Calculation verification is also referred to as numerical error estimation. References 13 and 14 discuss the differences and emphases of code verification and calculation verification.

Mathematically rigorous verification of a code would require proof that the algorithms implemented in the code correctly approximate the underlying PDEs and the stated initial conditions and boundary conditions. In addition, it would also have to be proven that the algorithms converge to the correct solutions of these equations in all circumstances under which the code will be applied. Such proofs are currently not available for general-purpose computational physics software. Executing the elements of code verification and calculation verification that are identified as necessary in this document is critical for V&V, but not sufficient in the sense of mathematical proof [15].

4.1 Code Verification

The assessment activity of code verification can be logically segregated into the following two parts:

(a) numerical code verification, which focuses on the underlying mathematical correctness and specific implementations of discrete algorithms for solving PDEs

(b) SQE or software quality assurance (SQA), which address such matters as configuration management, version control, code architecture, documentation, and regression testing [15]

Although CSM code users are typically not directly involved in developing and producing the modeling software they use, it is important that these users provide feedback to the developers to ensure that the best software engineering practices are consistently employed for the codes they use. Otherwise, unnecessary faults in the code may impact simulation results intermittently and unpredictably.

4.1.1 Numerical Code Verification. The objective of numerical code verification is to verify that the numerical solution algorithms are correctly implemented (programmed) in the code and that these algorithms are functioning as intended. Numerical code verification relies on careful investigations of topics such as spatial and temporal convergence rates, iterative convergence rates, independence of numerical solutions to coordinate transformations, and appropriate preservation of symmetry related to various types of initial and boundary conditions. In CSM, the primary solution algorithms are the finite-element method and the finite-difference method. Although the formal (theoretical) order of accuracy of these algorithms may be known from power series expansions of the discrete equations, the observed order of accuracy can be different. Thus, an important part of code verification is to determine the observed order of accuracy of the solution algorithm, which is the rate at which the solution asymptotically approaches the exact solution as the discretization is refined. This can be done by comparing two or more computational results with different discretizations to an exact solution and observing the rate of convergence.

Many factors can degrade the observed order of accuracy relative to the formal order of accuracy that is reported as a mathematical feature of an algorithm. These factors include programming errors, insufficient mesh resolution to achieve the asymptotic range, mixed accuracy issues, singularities, discontinuities, contact surfaces, mesh clustering, inadequate iterative convergence, and over-specified boundary conditions [13, 16]. In verification, all of these reasons for degradation in the order of accuracy are evidence of possible algorithmic or code errors and must be understood.

The primary tasks in numerical code verification are to define appropriate test problems for evaluating the accuracy of the numerical algorithms and to assess the performance of these algorithms on the test problems. Numerical code verification depends on comparing computational solutions to the "correct answer," which is provided by analytical solutions or highly accurate numerical solutions for a set of well-chosen test problems. The correct answer to a physically meaningful problem can only be known in a relatively small number of simple cases that generally exercise only a limited portion of the code. Fortunately, the method of manufactured solutions (MMS) offers a technique for deriving a mathematically exact solution to a closely related problem in order to exercise all aspects of the code that would be activated by the physical problems of interest. Because such cases assume a very important role in verification, they should be carefully formulated to provide a comprehensive set of test problems for verification of the code.

Given the paucity of good benchmarks for complex mathematical models, two points must be made. The first point is that some solutions are better than others; therefore, a hierarchy of confidence should be recognized. Similar to the AIAA Guide [2], the following organization of confidence (from highest to lowest) for the testing of algorithms is advocated:

(a) exact analytical solutions (including manufactured solutions)

(b) semianalytic solutions [reduction to numerical integration of ordinary differential equations (ODEs), etc.]

(c) highly accurate numerical solutions to PDEs

The second point is that some test problems are more appropriate than others, so application-relevant test problems should be used. These test problems could be ones with which users have a great deal of experience, or they could be ones that are constructed to address specific needs that arise when planning the verification activities.

Paragraphs 4.1.1.1 through 4.1.1.4 provide additional information on the kinds of tests and techniques employed in numerical code verification.

4.1.1.1 Analytical Solutions. Two categories of analytical solutions are of interest in code verification. First, there are those that correspond to plausible — if often greatly simplified — real-world physics. Second, there are manufactured solutions, which are defined and discussed in para. 4.1.1.2.

"Physically plausible" analytical solutions are solutions to the mathematical model's PDEs, with initial conditions and boundary conditions that can realistically be imposed, such as uniform pressure on a simply supported elastic plate. These solutions are sometimes exact (requiring only arithmetic evaluations of explicit mathematical expressions), but are often semianalytic (represented by infinite series, complex integrals, or asymptotic expansions). Difficulties can arise in computing any of these semianalytic solutions, especially infinite series. The analyst must ensure that when used for code verification, numerical error has been reduced to an acceptable level.

Typically for problems that allow analytical solutions, whether exact or semianalytic, pass/fail criteria can be stated in terms of the following two types of comparison:

(a) the agreement between the observed order of accuracy and the formal order of accuracy of the numerical method

(b) the agreement of the converged numerical solution with the analytical solution using specified norms

When computational solutions are compared with analytic solutions, either the comparisons should be examined in the regions of interest or the error norms should be computed over the entire solution domain. The accuracy of each of the dependent variables or functionals of interest should be determined as part of the comparison.

4.1.1.2 Method of Manufactured Solutions (MMS). The MMS is a technique for developing a special type of analytical solution [13, 17]. To apply it, an analyst prescribes solution functions for the PDEs and finds the forcing functions that are consistent with the prescribed solution. That is, the prescribed solution functions are inserted into the PDEs, and the equations are rearranged such that all remaining terms in excess of the terms in the original PDEs are grouped into forcing functions or source terms. Initial conditions and boundary conditions are similarly derived, based on the prescribed solution on the boundary. For example, for the simply supported plate problem, one could prescribe a solution of displacements that requires a highly variable pressure distribution or even applied internal moments. If this pressure and moment "forcing function" can be derived, it can then be applied using a computational model for the plate, and the computed displacement field can be compared to the prescribed solution.

The advantages of the MMS are many. It can be applied to a wide variety of highly nonlinear problems. It can test a large number of numerical features in the code, such as the numerical method, the spatial-transformation technique for mesh generation, the mesh distribution technique, and the correctness of algorithm coding [13]. The MMS provides a clear assessment because unless there are software errors, the computational results must agree with the solution used to derive the forcing function.

The MMS, however, is not without its disadvantages. In any nontrivial application of this method, the algebra and calculus required to derive the forcing function can become very complex, and symbolic manipulation software may offer the only practical recourse. Using the MMS can also require special coding and compilation if the code does not admit separate externally applied nodal forces for every degree of freedom at every node, each with its own time history. While the MMS can efficiently highlight the presence of errors, it cannot point to the sources of these errors and cannot identify mistakes in algorithm efficiency [13, 17].

4.1.1.3 Numerical Benchmark Solutions. When analytic solutions cannot be found or derived, the only other option for benchmark solutions is numerically derived ones. There are two distinct categories of highly accurate numerical benchmark solutions. One category consists of solutions in which the PDEs have been reduced by similarity transformations or other means to one or more ODEs that must be integrated numerically. The other category consists of solutions in which the PDEs have been solved directly by numerical methods. The accuracy of such numerical benchmark solutions has to be critically assessed to qualify them for use in code verification. For the numerical integration of ODEs, well-established standard methods are available for assessing accuracy. In the case of numerically integrated PDEs, no published solution can be considered a benchmark until the code used in producing that solution has been thoroughly verified and documented. In addition, comprehensive numerical error estimation must be reported. Credibility will be enhanced if independent investigators, preferably using different numerical approaches and computer software, produce multiple solutions that agree. Using multiple independent sources for the solutions will mitigate the risk of errors in the verification benchmark.

4.1.1.4 Consistency Tests. Consistency tests can be used to verify numerical algorithms. Global as well as local tests should be made for the conservation of mass, momentum, and energy [18]. An algorithm can satisfy the conservation laws exactly, or it can satisfy the laws in the limit of infinite resolution; this distinction should be considered when assessing the accuracy of an algorithm. Consistency tests can also be made that involve geometry (e.g., checking that the same numerical solution is obtained in different coordinate systems or determining whether specific symmetry features are preserved in the solution). Consistency tests should be considered complementary to the other types of algorithm tests described herein for numerical algorithm verification. If they can be devised, consistency tests are especially important because the failure of these tests indicates that there are unacceptable errors in the code.

4.1.2 Software Quality Engineering (SQE). The SQE part of code verification refers to procedures used to provide evidence that the software implementation of the numerical algorithms is free of programming errors and implementation faults. Most commonly, such errors reside in the source code, but occasionally flaws in the compiler introduce them. Evidence of error-free software from SQE is a necessary element of verification. SQE determines whether the software system is reliable and produces reliable results on specified computer hardware with a specified software environment (compilers, libraries). To optimize its influence on code verification, SQE should be planned and used during the development of the software product, not as a retrospective activity for a fielded software implementation [19]. However, feedback from users to developers is highly encouraged.

4.2 Calculation Verification

Calculation verification is applied to a computational model that is intended to predict validation results.

Thus, each computational model developed in a validation hierarchy would be subject to calculation verification. The goal of calculation verification is to estimate the numerical error associated with the discretization. In most cases, exercising the computational model with multiple meshes is required to estimate this error. Another source of error is mesh bias, wherein the arrangement of the elements can influence the results, especially if the mesh is coarse.

The two basic categories of approaches for estimating the error in a numerical solution to a complex set of PDEs are a priori and a posteriori. A priori approaches use only information about the numerical algorithm that approximates the partial differential operators and the given initial and boundary conditions [13, 20, 21]. A posteriori error estimation approaches use all of the a priori information plus the results from two or more numerical solutions to the same problem that have different mesh densities and/or different time steps. The discussion here focuses on a posteriori error estimates because they can provide quantitative assessments of numerical error in practical cases of nonlinear PDEs.

4.2.1 A Posteriori Error Estimation. A posteriori error estimation has primarily been approached using either finite-element–based error estimation techniques [22, 23] or multiple-mesh solutions combined with Richardson extrapolation and extensions thereof [13].

Two fundamentally different types of finite-element–based discretization error estimators have been developed. The most commonly used are recovery methods, which involve postprocessing of either solution gradients or nodal values in patches of neighboring elements. These provide direct error estimates only in the global energy norm; however, they provide ordered error estimates for specific field quantities of interest (i.e., the estimate improves with mesh refinement).

The second class of finite-element–based error estimators consists of residual-based methods. Like recovery methods, residual methods were originally formulated to provide error estimates in the global energy norm. Extension to error estimates in quantities of interest, such as deflections or stresses, generally requires additional solutions [24].

Single-mesh finite-element–based error estimates, when applicable, offer a great advantage by reducing mesh-generation and computational effort. However, the estimates require that the convergence rate be assumed. Calculation of an observed convergence rate always requires the generation of multiple meshes. The single-mesh a posteriori methods are also important for finite element adaptivity, where both the spatial mesh density (known as h-adaptivity) and the order of the finite element scheme (known as p-adaptivity) can be adapted [22, 23].

Standard Richardson extrapolation assumes that

(a) the observed order of accuracy (rate of convergence) is known

(b) two numerical solutions at different mesh resolutions have been computed

(c) both solutions are in the asymptotic convergence regime

To estimate a bound on the numerical error, the method then extrapolates to a more accurate value against which to compare the original solution. Various elaborations of Richardson extrapolation use three or more meshes to calculate an observed order of accuracy [13]. The observed order of accuracy can be used to verify a theoretical order of accuracy, test whether the solution is in the asymptotic regime, and estimate a zero-mesh-size converged solution using extrapolation. A grid convergence index (GCI) based on Richardson extrapolation has been developed and advocated to assist in estimating bounds on the mesh convergence error [13, 25]. The GCI can convert error estimates that are obtained from any mesh-refinement ratio into an equivalent mesh-doubling estimate. More generally, the GCI produces an error-bound estimate through an empirically based factor of safety applied to the Richardson error estimate [13].

4.2.2 Potential Limitations. The assumption of smoothness in solutions (i.e., the absence of singularities and discontinuities), underlies much of the theory of existing error estimation techniques and is quite demanding in estimating local errors in the solution domain; however, this assumption does not prevent the use of an empirical approach to error estimation based on observed convergence rates. Experience shows that an empirical approach is more dependable when more than three meshes are used with a least squares evaluation of observed convergence rates and when functionals rather than point values are considered.

Singularities and discontinuities commonly occur in solid mechanics; the crack tip singularity is an example. The difficulties of singularities and discontinuities are compounded in very complex conceptual models, where multiple space and time scales may be important and very strong nonlinearities may be present. Ideally, calculation verification should be able to confront these complexities. However, the "pollution" of particular regions of a calculation by the presence of singularities such as shock waves, geometrical singularities, or crack propagation is a subject of concern in error estimation [13, 23, 26], and there is a lack of rigorous theory for guidance in these situations.

Another complexity in numerical error estimation is the coupling that can occur between numerical error and the spatial and temporal scales in certain types of physical models. Refining the mesh does not ensure that the physics modeled will remain unchanged as the mesh is resolved. For example, an insufficiently refined mesh in buckling problems will prevent the model from exhibiting higher modes of buckling. This observation regarding mesh refinement directly influences the accuracy and reliability of any type of a posteriori error estimation method, especially extrapolation methods.

4.3 Verification Documentation

Documentation needs to be an integral part of the verification process to facilitate reuse of the model. The documentation should explain the rationale and limitations of the code verification and calculation verification activities. It should include descriptions of the error estimation techniques employed, the results of consistency tests, and the analytical solutions, manufactured solutions, and numerical benchmark solutions used. SQE and SQA, configuration management, and acceptable computational systems should also be described.

5 VALIDATION

The activities described in this section are performed for each reality of interest in the validation hierarchy developed during preparation of the V&V plan.

The goal of validation is to determine the predictive capability of a computational model for its intended use. This is accomplished by comparing computational predictions (simulation outcomes) to observations (experimental outcomes). Three prerequisites for meaningful validation are

(a) having a clear definition of the model's intended use

(b) having already conducted code verification and calculation verification activities sufficiently so that the errors discovered through validation can be isolated from those errors discovered through verification

(c) quantifying uncertainties in both the simulation outcomes and the experimental outcomes

The approach of validation is to measure the agreement between the simulation outcomes from a computational model and the experimental outcomes from appropriately designed and conducted experiments. These outcomes should incorporate the experimental and modeling uncertainties in dimensions, materials, loads, and responses. In most cases, assessing the predictive capability of a computational model over the full range of its intended use cannot be based solely upon data already available at the beginning of the V&V program. Not only might existing data inadequately represent the intended use of the model, it may also have been used in model calibration during the development of the computational model. In such cases, new experiments and computational predictions are required. The challenge is to define and conduct a set of experiments that will provide a stringent enough test of the model that the decision maker will have adequate confidence to employ the model for predicting the reality of interest. If the model predicts the experimental outcomes within the predetermined accuracy requirements, the model is considered validated for its intended use.

5.1 Validation Experiments

Validation experiments are performed to generate data for assessing the accuracy of the mathematical model via simulation outcomes produced by the verified computational model. A validation experiment is a physical realization of a properly posed applied mathematics problem with initial conditions, boundary conditions, material properties, and external forces. To qualify as a validation experiment, the geometry of the object being tested (e.g., a component, subassembly, assembly, or full system), the initial conditions and the boundary conditions of the experiment, and all of the other model input parameters must be prescribed as completely and accurately as possible. Ideally, this thoroughness on the part of the experimenter will provide as many constraints as possible, requiring few assumptions on the part of the modeler. All of the applied loads, multiple response features, and changes in the boundary conditions should be measured; and uncertainties in the measurements should be reported.

5.1.1 Experiment Design. Generally, data from the literature are from experiments performed for other purposes and thus do not meet the requirement of a validation experiment. Experiments can have many purposes and are often focused on assessing component performance relative to safety criteria or on exploring modes of system response. Consequently, the measurement set in many experiments may differ from the measurements needed for model validation. For example, a test may show that a component fails at a load higher than an acceptable threshold and thereby establish that the component is acceptable for use. However, the test may not have measured the deformation as the force was applied because that measurement was not needed for the purpose of the experiment. If both the component-failure measurement and the deformation measurement were necessary to validate a computational model, the test measuring only component failure could not be used for validation. Furthermore, it is essentially impossible to make blind predictions of experiments whose results are known prior to the validation effort because the results guide, if even subconsciously, modelers' assumptions and their selection of unmeasured quantities. For these reasons, it is usually necessary to perform experiments that are dedicated to model validation [3].

The modeler should have input to the design of the validation experiments. The experimenter and the modeler need to share an understanding of the responses that are difficult to measure or predict. Additionally, the modeler needs to be certain that all the inputs (especially for constitutive models), boundary conditions, and imposed loads are being measured. The modeler should perform a parametric study with the verified model to determine model sensitivities that need to be investigated experimentally. Additionally, pretest analyses should be conducted to uncover potential problems with the design of the experiment. However, credibility of the validation process will be greatly enhanced if the modeler does not know the test results before the prediction is complete, with the exception that the modeler must be provided material properties, applied loads, and initial and boundary conditions.

In summary, the validation experiments and measurement set should be designed to leave as few unknown parameters as possible. In the all-too-common case that some significant parameters are not measured, the modeler has to perform multiple calculations to compare with the experiments by varying the values of those parameters. The modeler cannot arbitrarily select a parameter value within its accepted range and base the validation comparison on that selection because doing so can result in either false validation or false invalidation. If all of the calculation results using a realistic range of the parameters are within the acceptable tolerance for validation, then validation may be claimed, even though the experiment had uncontrolled variables. But if the calculation results for a significant portion of the realistic parameter range lie outside this tolerance, validation cannot be claimed, and progress can only be made by the experimenter constraining the range of the unmeasured or poorly measured parameters.

5.1.2 Measurement Selection. Selecting the quantities to measure should be based primarily on the response features of interest. When possible, these features should be measured directly rather than derived from other measurements. For example, if strain is the feature of interest, it would probably be better to use a strain gauge instead of multiple measurements of displacement. Similarly, if velocity can be measured directly, that approach would be better than integrating a measurement of acceleration or differentiating a measurement of displacement. On the other hand, consistency of the test data is an important attribute that increases confidence in the data. Data consistency can be established by independent corroborative measurements (e.g., measuring displacement or acceleration to corroborate measurements of velocity). Measurements of point quantities made in families that allow fields to be estimated are also useful; for example, a displacement field can be used to corroborate measurements of strain [27].

Another reason that variables or locations in the model other than those specified in the validation requirements should be measured is that agreement between these measurements and the simulation results can contribute significantly to overall confidence in the model. Although some quantities may be of secondary importance, accurate calculations of these quantities provide evidence that the model accurately calculates the primary response for the right reason. For example, confidence in a model that matches the central deflection of a beam is greatly enhanced if it also matches the displacements or strains all along the length of the beam — even if central deflection is the only quantity of interest for the intended use. This can qualitatively or even quantitatively build confidence that the model can be used to make accurate predictions for problem specifications that are different from those included in model development and validation. Thus, validation experiments should produce a variety of data so that multiple aspects of the model can be assessed.

5.1.3 Sources of Error. It is important to calibrate the gauges that will be used in validation experiments and to document their inaccuracies related to nonlinearity, repeatability, and hysteresis. Many things can influence the output of a gauge. Pressure transducers, for example, should be calibrated in an environment similar to that of the validation experiment (e.g., at elevated temperature). If a transducer is sensitive to the environment and the environment changes significantly during a validation test, the transducer's sensitivity to the environment must already have been established (during previous calibration of the gauge) so that the resulting data can be corrected to account for the transducer's sensitivity to the environment [28].

In addition, the experimenter needs to determine and account for effects such as the compliance or inertia of any test fixtures if these effects contribute to the measurement of displacement or force, respectively. For example, the mass of a piston in a hydraulic testing machine can affect the measurement of the force applied to the specimen and, if ignored, can contribute to lack of agreement between the simulation results and the experimental data. Reporting the details of operating, calibrating, and installing the gauges used in an experiment helps the modeler understand the relationship between gauge output and model output. It may even be necessary in some cases for a modeler to build a model that includes such parts as the test fixtures or measurement fixtures to accurately predict the measurements.

5.1.4 Redundant Measurements. For validation experiments, redundant measurements are needed to establish the precision (scatter) in the validation test results and thus improve the quantification of uncertainty in experimental measurements. One approach for obtaining redundant measurements is to repeat the test using different specimens. The test-to-test scatter could then have contributions from differences in specimens (initial conditions) or material properties, specimen installation (boundary conditions), gauges, gauge installation, and data acquisition. An example would be to perform bending tests on several members of a set of beams and to measure the response with strain gauges mounted on the tension and compression surfaces. Not only would the beams be different, they might be off center in the testing machine by different amounts. In addition, the strain gauges would have different scatter in location and orientation, and the signal-wire resistances would differ. Another approach for obtaining redundant measurements is to repeat the test using the same specimen. This approach might be taken if the cost of testing is high or the availability of test specimens is limited. Of course, specimen-to-specimen response variability would not be obtained. Still another approach for obtaining redundant measurements is to place similar transducers at symmetrical locations (if the test has adequate symmetry) to assess scatter. The data from these transducers could also be used to determine whether the expected symmetry was indeed obtained.

5.2 Uncertainty Quantification in Experiments

In the uncertainty quantification activity for experiments, the effects of measurement error, design tolerances, construction uncertainty, and other uncertainties are quantified, resulting in the experimental outcomes. Although published experimental results often do not include an assessment of uncertainty, it is necessary to estimate and report the uncertainty in the measurements in validation experiments so that simulation results can be judged appropriately.

In experimental work, errors are usually classified as being either random error (precision) or systematic error (bias). An error is classified as random if it contributes to the scatter of the data in redundant measurements or repeat experiments at the same facility. Random errors are inherent to the experiment, produce nondeterministic effects, and cannot be reduced with additional testing, although they can be better quantified with additional testing. Sources of random error include dimensional tolerances on test parts or measurement locations, variability of material properties, and mechanical equipment variances due to friction. Systematic errors can produce a bias in the experimental measurements that is difficult to detect and estimate. Sources of systematic error include transducer calibration error, data acquisition error, data reduction error, and test technique error [29].

Either the experimenter or an independent reviewer must provide an uncertainty assessment of the results. The assessment should consider all sources of experimental uncertainty, whether the sources were measured or estimated. When possible, the uncertainties should take the form of mean values with standard deviations or distributions [30]. But even when statistics are not available, an estimate of experimental uncertainty based on previous experience or expert opinion is necessary before proceeding to comparisons with simulation outcomes. A common pitfall is to neglect important contributions to modeling uncertainty, experimental uncertainty, or both, and then to try to draw conclusions about predictive accuracy based on inadequate information. Improper or inappropriate inferences could thus be made about the accuracy of the computational model.

5.3 Accuracy Assessment

Following uncertainty quantification of the experimental data that resulted in the experimental outcomes, the final step in validation consists of

(a) comparing values of the metrics chosen to measure the agreement between simulation outcomes and experimental outcomes

(b) making an assessment of the accuracy of the computational model relative to the goals provided in the V&V plan for the model's intended use

Recall that an accuracy assessment is made for each component, subassembly, and assembly in every level of the validation hierarchy for which validation data are produced (Fig. 3). The determination of the system-level model's accuracy is made after the hierarchy of validation experiments has been performed and the composite computational model has been validated through the various hierarchical tiers.

5.3.1 Validation Metrics. A validation metric provides a method by which the simulation outcomes and the experimental outcomes can be compared. A metric is a mathematical measure of the difference between the two outcomes such that the measure is zero only if the two outcomes are identical. Validation metrics should accommodate computational and experimental uncertainty to the extent possible.

Features of the experimental outcomes and the simulation outcomes compared using validation metrics should be carefully selected. A feature and its associated metric may be simple. For example, a simple binary metric would be the following: Is the material's yield stress exceeded in the simulation? More typical is a quantification metric (e.g., the difference between the yield stress and the calculated von Mises stress). Most often, a quantification metric is normalized by dividing the difference by the measurement; this is often referred to as the "relative error." Frequently, spatial or temporal distributions need to be compared (e.g., strain as a function of distance, or velocity at a point as a function of time). In this case, a weighted sum of squares provides a convenient measure of the magnitude difference. In addition to the magnitude difference, such distributions may have phase differences. For example, the time of arrival for velocity waveforms can affect the phase difference; a phase metric has been proposed for this kind of error [31]. Metrics of different types like magnitude and phase can be combined using a simple sum of squares [32] or a weighted average, where the weights are assigned by subject-matter experts or by individual engineering judgment.

Validation metrics can sometimes be devised to incorporate the uncertainties associated with the experimental outcomes and the uncertainties associated with the simulation outcomes (e.g., the input parameter uncertainties propagated through the computational model). When multiple (repeat) experiments have been performed, the mean and variance of the system response of interest can be quantified. A metric for the special case of multiple experiments, but no uncertainty in the simulation outcomes, has been proposed [33]. For the general case, where both the measurement and simulation are expressed as a mean with variance, some research has been performed [34], but this and other aspects of validation metrics are still active areas of research.

5.3.2 Accuracy Adequacy. It is possible that the model will be demonstrated to fulfill only a portion of the validation requirements. The accuracy may fall short of the requirements in general or for a certain portion of the intended use. For example, a 10% accuracy goal may be unmet, but 15% accuracy may be established. Alternately, the 10% accuracy may be met for loads under or over a given level or for all but a particular type, such as thermal. Assuming that the original criteria were properly established for the intended use, this implies that further model improvements are needed. In the meantime, the model may have utility on a limited basis (i.e., it may be validated to a lower standard than that specified in the V&V plan, or it may be partially validated). In such cases, the technical experts and decision makers have the shared burden of establishing partial acceptance criteria. They could establish a new and less-ambitious definition of the acceptable level of agreement for validation, or they could define the limitations of the model's use. Partial validation is not uncommon, and this underscores the point that a verdict or claim of "validation" is never meaningful without reporting the accuracy criteria and the uncertainties in experiments and calculations.

Confidence in the model's predictions decreases as the conditions of application deviate from those used in the validation process. For example, a model of an engine block that has been developed to accurately predict the stresses on the cylinder surfaces may not give adequately accurate predictions of the stress near an internal cooling channel in the same model. Confidence in the model's output is limited to applications that are judged to be sufficiently similar to that for which validation was performed. Confident use for other purposes requires additional validation.

5.4 Validation Documentation

Documentation of the overall validation process and the specific validation activity (for each reality of interest in the hierarchy) conveys an understanding of the predictive capability of the model for its intended use and supports the conclusion about whether or not the model was successfully validated for its intended use. The documentation also facilitates reuse of the knowledge base by enabling subsequent users to build upon the established validation activity, regardless of whether the model was successfully validated for its original intended use.

For each reality of interest, the validation documentation should build upon the documentation of the conceptual model and the documentation describing the verification process of the computational model. Accordingly, the validation documentation could be couched in terms of answering the following questions: Are the approximations and uncertainties inherent in the modeling approach appropriate for the intended application? Are the model and experiments adequate to predict the intended system responses for the reality of interest?

Also, for each reality of interest, the validation documentation should describe all experiments that were performed to test the associated computational model. The description of an experiment should specify the dimensions, the manner in which the boundary conditions were applied, the method by which the measurements were taken, the material properties, the types of equipment that were used, and the uncertainties in calibrations. Documentation of the experiments is usually the primary mode of communication between the experimenter and the modeler, so thoroughness here is especially important.

At the global level, as it applies to the model as a whole, the validation documentation should present not only the arguments for (or against) accepting the model as validated for its intended use, but also the recommended limits of its use. Idealizations and limitations present in the system-level validation experiments should be explained, as should the applicability of lower-tier validation data to the intended use. The range of system configurations, loading environments, materials, etc., for which predictions are expected to have adequate accuracy should be delineated. Similar to the V&V planning process, this requires the expert judgment of engineers and other informed participants.

6 CONCLUDING REMARKS

A summary of the guidelines presented in this document that are designed to help assess accuracy and enhance the credibility of CSM models is as follows: a V&V plan should be developed to guide model development and validation experiment design. Accuracy requirements also should be established in this plan. A computational model is validated for its intended use by demonstrating its ability to predict validation experiments with acceptable accuracy. Although calibration of a model to experimental data may demonstrate the model's fitting ability, calibration does not demonstrate its predictive capability. Thus, calibration is not validation. Code verification and calculation verification must precede validation calculations. Confidence in the model is best achieved when validation is pursued in a hierarchical fashion from the component level to the system level. Validation experiments should leave as few unknown parameters as possible, but multiple calculations of an experiment with a range of values for unmeasured parameters are usually needed. For meaningful comparisons of simulation and experiment, assessments of uncertainty in both simulation results and experimental data have to be performed. The rationale and results of a V&V program should be well-documented not only for enhancing credibility of the model for its present intended use, but also for building a knowledge base for future applications.

The remainder of this section briefly discusses issues that need to be addressed so that V&V for CSM can further evolve into a more robust and quantitative methodology. These issues include both technical and managerial challenges.

With the ever-increasing complexity in CSM models, especially constitutive models, the task of verification becomes more difficult because of a lack of relevant analytical solutions. The danger is that without adequate verification, any lack of accuracy in a model's predictive capability cannot be isolated to either model implementation errors (the role of verification) or inadequate representation of the physics (the role of validation). Thus, there is an ongoing need to support verification with the generation of analytical solutions and manufactured solutions.

In general, V&V should support risk assessments across the spectrum of decisions (especially high-consequence decisions) that are intended to be based, at least in part, on simulation in lieu of experimental data before a system is constructed, during its test and evaluation, during its operation, or when full system testing is not achievable. The challenge is to quantify the reduction in risk that can be achieved with V&V. For example, enhanced confidence through V&V can reduce the risk that safety factors employed in the design are inadequate. Similarly, enhanced confidence in the computational model used to predict the system response may allow for a reduction in conservatism in the design. Quantifying this risk reduction or cost savings can then guide the establishment of accuracy requirements and appropriate monetary investment in a V&V program.

The recommended V&V activities cannot guarantee that accuracy requirements will be achieved within the time and budget for V&V. Decision makers will sometimes have to address the trade-off between additional cost and additional risk when a demonstration of accuracy does not completely fulfill a requirement. V&V activities might expose a level of risk that is unacceptable to the decision maker and force the investment in model improvement; alternately, the decision maker might have to accept a higher level of risk.

Investment in V&V for a computational model will take various forms, of which the most visible and costly is likely to be experimentation. Decades of experience — successes and failures — in computational mechanics have brought about fundamental changes in experimentation, and the validation of computational models has evolved as a significant justification for experimentation. This evolution in experimentation continues, and it is highlighting the need for validation experiments, which are designed to produce experimental data of high-enough quality to quantify the uncertainties in the experimental approach. Dedicated validation experiments thus contribute directly to the confidence in the model. However, it is difficult to quantify the need for new experiments, especially in continuing product development programs with reliable legacy experimental data for previous product generations. Because the value added by formal validation experiments and the associated blind pretest calculations is difficult to quantify, it can be difficult to decide how much to invest in new experimentation.

Robust V&V is facilitated by devoting more effort to (and placing more significance on) estimating the uncertainty in validation experiments. In a similar vein, it is appropriate to generate multiple predictions for an experiment by using a range of parameter values within their range of uncertainty and to then compare a family of predictions to the family of experimental results, as opposed to making one prediction for one experiment. However, the science of quantitatively comparing two distributions of outcomes is neither simple nor mature.

Perhaps the overarching challenge in validation is to establish a framework for quantifying a model's predictive capabilities in regions of the reality of interest that are farthest from the validation tests. This can be particularly problematic when system-level experimental data are insufficiently representative of the full intended use; only subsystem-level data are available (e.g., either before system construction or if full-system testing is not achievable); or system-level data are not supported by a hierarchy of subsystem data.

Currently, a model's predictive capability can be quantitatively demonstrated for the validation test database, but the predictive capability for other cases cannot be quantified. As it stands, the task of establishing accuracy requirements and estimating the accuracy of extrapolations is based on expert engineering judgment. Advances in physics modeling, decision theory, and mathematics are needed to meet this challenge.

In closing, it is important to clarify the division of responsibilities among management, code developers, code users, data users, and experimenters to enhance the efficiency of the V&V program. Given the V&V guidelines recommended in this document, the following responsibilities are implied: code developers (whether commercial, industrial, government, or university) should take increasing responsibility for code verification; modelers should shoulder the primary obligation for calculation verification; modelers and experimenters should be jointly responsible for validation activities; and management and policy makers who base their decisions on computational simulations, and who are ultimately accountable for assessing risk, should have primary responsibility at the initiation of the V&V program for setting the accuracy requirements.

Intentionally left blank

MANDATORY APPENDIX I
GLOSSARY

Various groups working in the field of V&V have described the semantics of terms related to V&V in a variety of ways. Thus, it is necessary for the CSM community to specify which of the existing definitions of the terms will be used or to independently develop a new definition. Wherever possible, existing definitions are used herein to preserve consistency.

adequacy: the condition of satisfying all requirements for model acceptance, including those for model accuracy and for programmatic constraints such as implementation, cost, maintenance, and ease of use.

calculation verification: the process of determining the solution accuracy of a particular calculation.

calibration: the process of adjusting physical modeling parameters in the computational model to improve agreement with experimental data.

calibration experiment: an experiment performed to improve estimates of some parameters in the mathematical model.

code: the computer implementation of algorithms developed to facilitate the formulation and approximate solution of a class of problems.

code verification: the process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software.

computational model: the numerical implementation of the mathematical model, usually in the form of numerical discretization, solution algorithm, and convergence criteria.

conceptual model: the collection of assumptions and descriptions of physical processes representing the solid mechanics behavior of the reality of interest from which the mathematical model and validation experiments can be constructed.

error: a recognizable deficiency in any phase or activity of modeling or experimentation that is not due to lack of knowledge (e.g., choosing an incorrect material property for use in the computational model, recording a gain incorrectly during a sensor calibration, incorrectly defining a data format statement in the code).

experimental data: raw or processed observations (measurements) obtained from performing an experiment.

experimental outcomes: features of interest extracted from experimental data that will be used, along with estimates of the uncertainty, for validation comparisons.

formal order of accuracy: values of the exponents of the leading terms of the power series expansion of the truncation error in the discrete equations that represent the PDEs of interest. This is the theoretical convergence rate of the discretization method in both space and time.

intended use: the specific purpose for which the computational model is to be used.

irreducible uncertainty: inherent variation associated with the physical system being modeled. Also called "aleatory uncertainty." For example, the yield strength of the steel in a production lot of beams follows a probability distribution. With many observations, this distribution can be defined, but it cannot be eliminated.

mathematical model: the mathematical equations, boundary values, initial conditions, and modeling data needed to describe the conceptual model.

model: the conceptual, mathematical, and numerical representations of the physical phenomena needed to represent specific real-world conditions and scenarios. Thus, the model includes the geometrical representation, governing equations, boundary and initial conditions, loadings, constitutive models and related material parameters, spatial and temporal approximations, and numerical solution algorithms.

model update: the process of changing the basic assumptions, structure, parameter estimates, boundary conditions, or initial conditions of a model to improve model accuracy.

observed order of accuracy (or convergence): the empirically determined rate of convergence of the solution to a set of discrete equations as the spatial and temporal discretizations approach zero. This rate can be obtained by comparing multiple computational solution results that use different levels of discretization.

PIRT (Phenomena Identification and Ranking Table): a list of the physical processes that influence the system responses of interest, along with a ranking (e.g., high, medium, low) of the importance of each process.

prediction: the output from a model that calculates the response of a physical system before experimental data are available to the user.

reality of interest: the physical system and its associated environment to which the computational model will be applied.

reducible uncertainty: the potential deficiency that can be reduced by gathering more data, observations, or information. Also called "epistemic uncertainty." An example is the mathematical form used to describe how force in a spring develops when the spring is deformed. The spring may be linear or nonlinear with respect to deformation. Some simple tests will reveal the nature.

referent: data, theory, or information against which simulation results will be compared.

simulation: the computer calculations performed with the computational model (i.e., "running the model").

simulation outcomes: features of interest extracted from simulation results that will be used, along with estimates of the uncertainty, for validation comparisons.

simulation results: output generated by the computational model.

uncertainty: a potential deficiency in any phase or activity of the modeling, computation, or experimentation process that is due to inherent variability or lack of knowledge.

uncertainty quantification: the process of characterizing all uncertainties in the model or experiment and of quantifying their effect on the simulation or experimental outcomes.

validated: through V&V, a model can be declared "validated for its intended use" once it has met the validation requirements.

validation: the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

validation experiment: an experiment that is designed and performed to generate data for the purpose of model validation.

validation metric: mathematical measure that quantifies the level of agreement between simulation outcomes and experimental outcomes.

validation requirements: specifications and expectations that a computational model must meet to be acceptable for its intended use. These requirements are established in the V&V plan and often include descriptions of specific tests that need to be run or other referents that need to be gathered, as well as accuracy statements about the agreement between simulation outcomes and experimental outcomes or referents.

verification: the process of determining that a computational model accurately represents the underlying mathematical model and its solution.

MANDATORY APPENDIX II
NOMENCLATURE

The following commonly used terms, which appear throughout this Guide, are defined below:

AIAA	=	American Institute of Aeronautics and Astronautics
AIS	=	adaptive importance sampling
AMV	=	advanced mean value
ANOVA	=	analysis of variance
ASME	=	American Society of Mechanical Engineers
CSM	=	computational solid mechanics
DoD	=	Department of Defense
FORM	=	first-order reliability method
GCI	=	grid convergence index
MMS	=	method of manufactured solutions
ODE	=	ordinary differential equation
PDE	=	partial differential equation
PIRT	=	Phenomena Identification and Ranking Table
PTC	=	Performance Test Codes
RMS	=	root-mean-square
SQA	=	software quality assurance
SQE	=	software quality engineering
V&V	=	verification and validation

MANDATORY APPENDIX III
BIBLIOGRAPHY

The following sources were cited using numbered references throughout this Guide:

[1] U.S. Department of Defense, May 13, 2003, "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)," DoD Instruction 5000.61, Defense Modeling and Simulation Office, Washington, DC.

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

[3] Oberkampf, W. L., Trucano, T. G., and Hirsch, C., 2004, "Verification, Validation, and Predictive Capability in Computational Engineering and Physics," *Applied Mechanics Reviews*, 57(5), pp. 345–384.

[4] Thacker, B. H., Doebling, S. W., Hermez, F. M., Anderson, M. C., Pepin, J. E., and Rodriguez, E. A., 2004, "Concepts of Model Verification and Validation," LA-14167-MS, Los Alamos National Laboratory, Los Alamos, NM.

[5] Trucano, T., Pilch, M., and Oberkampf, W., 2002, "General Concepts for Experimental Validation of ASCI Code Applications," SAND2002-0341, Sandia National Laboratories, Albuquerque, NM.

[6] Mason, R. L., Gunst, R. F., and Hess, J. L., 2003, *Statistical Design and Analysis of Experiments: With Applications to Engineering and Science*, 2nd Edition, John Wiley & Sons, New York.

[7] Ang, W., and Tang, W., 1975, *Probability Concepts in Engineering Planning and Design, Basic Principles*, John Wiley & Sons, New York.

[8] Madsen, H. O., Krenk, S., and Lind, N. C., 1986, *Methods of Structural Safety*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

[9] Wu, Y.-T., Millwater, H. R., and Cruse, T. A., 1990, "Advanced Probabilistic Structural Analysis Methods for Implicit Performance Functions," *AIAA Journal*, 28(9), pp. 1663–1669.

[10] NESSUS User's Manual, Version 8.2, 2004, Southwest Research Institute, San Antonio, TX.

[11] Hasselman, T. K., 2001, "Quantification of Uncertainty in Structural Dynamics Models," *ASCE Journal of Aerospace Engineering*, 14(4), pp. 158–165.

[12] Hasselman, T. K., Yap, K. C., Wathugala, G. W., and Anderson, M. C., 2005, "A Top-Down Method for Uncertainty Quantification and Predictive Accuracy Assessment," *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, and 7th AIAA Non-Deterministic Approaches Forum*, Paper No. AIAA-2005-1903, Austin, TX, April 18–21, 2005.

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

[14] Roache, P. J., 1998, "Verification of Codes and Calculations," *AIAA Journal*, 36(5), pp. 696–702.

[15] Oberkampf, W. L., and Trucano, T. G., 2002, "Verification and Validation in Computational Fluid Dynamics, *Progress in Aerospace Sciences*, 38(3), pp. 209–272.

[16] Axelsson, O., 1996, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK.

[17] Knupp, P., and Salari, K., 2003, *Verification of Computer Codes in Computational Science and Engineering*, Chapman & Hall/CRC Press, Boca Raton, FL.

[18] Haworth, D. C., Tahry, S. H. E., and Huebler, M. S., 1993, "A Global Approach to Error Estimation and Physical Diagnostics in Multidimensional Computational Fluid Dynamics," *International Journal for Numerical Methods in Fluids*, 17(1), pp. 75–97.

[19] Schulmeyer, G. G., and Mackenzie, G. R., 2000, *Verification and Validation of Modern Software-Intensive Systems*, Prentice-Hall, Englewood Cliffs, NJ.

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

[21] Oden, J. T., 1994, "Error Estimation and Control in Computational Fluid Dynamics," in *The Mathematics of Finite Elements and Applications*, J. R. Whiteman, ed., John Wiley & Sons, New York, pp. 1–23.

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

[23] Babuska, I., and Strouboulis, T., 2001, *The Finite Element Method and Its Reliability*, Oxford University Press, Oxford, UK.

[24] Roy, C. J., 2004, "Verification of Codes and Solutions in Computational Simulation," *Proceedings of CHT-04 CHMT International Symposium on Advances in Computational Heat Transfer*, International Centre for Heat and Mass Transfer, Bergen, Norway, April 19–24, 2004.

[25] Roache, P. J., 1994, "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, 116(3), pp. 405–413.

[26] Oden, J. T., Feng, Y., and Prudhomme, S., 1998, "Local and Pollution Error Estimation for Stokesian Flow," *International Journal of Numerical Methods in Fluids*, 27(1–4), pp. 33–39.

[27] Senseny, P. E., and Pucik, T. A., 1999, "Development and Validation of Computer Models for Structures in Jointed Rock," *International Journal for Numerical and Analytical Methods in Geomechanics*, 23(8), pp. 751–778.

[28] Senseny, P. E., 1987, "Deformation Measurements in Rock Mechanics Using Direct-Contact Extensometers," *Proceedings, 4th Annual Hostile Environments and High Temperature Measurement Conference*, Society for Experimental Mechanics, Windsor Locks, CT, pp. 94–99.

[29] Coleman, H.W., and Steele, Jr., W. G., 1989, *Experimentation and Uncertainty Analysis for Engineers*, John Wiley & Sons, New York.

[30] American Society of Mechanical Engineers (ASME), 1998, *Test Uncertainty, Instruments and Apparatus*, PTC 19.1-1998, ASME, New York.

[31] Russell, D. M., 1997, "Error Measures for Comparing Transient Data: Part I: Development of a Comprehensive Error Measure," *Proceedings of the 68th Shock and Vibration Symposium*, Shock and Vibration Information Analysis Center, Baltimore, MD, pp. 175–184.

[32] Sprague, M. A., and Geers, T. L., 2004, "A Spectral-Element Method for Modeling Cavitation in Transient Fluid-Structure Interaction," *International Journal for Numerical Methods in Engineering*, 60(15), pp. 2467–2499.

[33] Oberkampf, W. L., and Barone, M. F., 2005, "Measures of Agreement between Computation and Experiment: Validation Metrics," SAND2005-4302, Sandia National Laboratories, Albuquerque, NM.

[34] Rebba, R., Huang, S., Liu, Y., and Mahadevan, S., 2006, "Statistical Validation of Simulation Models," *International Journal of Materials and Product Technology*, 25(1/2/3), pp. 164–181.

Intentionally left blank

Intentionally left blank

ASME V&V 10-2006

ISBN 0-7918-3042-X

9 780791 830420

C06906