2

Fundamental concepts and terminology

This chapter discusses the fundamental concepts and terminology associated with verification and validation (V&V) of models and simulations. We begin with a brief history of the philosophical foundations so that the reader can better understand why there are a wide variety of views toward V&V principles and procedures. Various perspectives of V&V have also generated different formal definitions of the terms *verification* and *validation* in important communities. Although the terminology is moving toward convergence within some communities, there are still significant differences. The reader needs to be aware of these differences in terminology to help minimize confusion and unnecessary disagreements, as well as to anticipate possible difficulties in contractual obligations in business and government. We also discuss a number of important and closely related terms in modeling and simulation (M&S). Examples are predictive capability, calibration, certification, uncertainty, and error. We end the chapter with a discussion of a conceptual framework for integrating verification, validation, and predictive capability. Although there are different frameworks for integrating these concepts, the framework discussed here has proven very helpful in understanding how the various activities in scientific computing are related.

2.1 Development of concepts and terminology

Philosophers of science have been struggling with the fundamental concepts underlying verification and validation (V&V) for at least two millennia. During the twentieth century, key philosophical concepts of epistemology were fundamentally altered (Popper, 1959; Kuhn, 1962; Carnap, 1963; Popper, 1969). These changes were heavily influenced by the experiments and theories associated with quantum mechanics and the theory of relativity. Usurping the throne of Newtonian mechanics, which had reigned for 300 years, did not come easily or painlessly to modern physics. Some researchers in engineering and the applied sciences have used several of the modern concepts in the philosophy of science to develop the fundamental principles and terminology of V&V. See Kleindorfer *et al.* (1998) for an excellent historical review of the philosophy of science viewpoint of validation. When this viewpoint is carried to the extreme (Oreskes *et al.*, 1994), one is left with the following position: one can only *disprove* or *fail to disprove* theories and laws of

nature. Stated differently, theories and laws cannot be proved; they can only be falsified (Popper, 1969). This position is valuable for philosophy of science, but it is nearly useless for assessing the credibility of computational results in engineering and technology. Engineering and technology must deal with practical decision making that is usually focused on requirements, schedule, and cost. During the last two decades a workable and constructive approach to the concepts, terminology, and methodology of V&V has been developed, but it was based on practical realities in business and government, *not* the issue of absolute truth in the philosophy of nature.

2.1.1 Early efforts of the operations research community

The first applied technical discipline that began to struggle with the methodology and terminology of V&V was the operations research (OR) community, also referred to as systems analysis or modeling and simulation (M&S) community. Some of the key early contributors in the OR field in the 1960s and 1970s were Naylor and Finger (1967); Churchman (1968); Klir (1969); Shannon (1975); Zeigler (1976); Oren and Zeigler (1979); and Sargent (1979). See Sheng *et al.* (1993) for a historical review of the development of V&V concepts from the OR viewpoint. For a conceptual and theoretical discussion of V&V in modern texts on M&S, see Bossel (1994); Zeigler *et al.* (2000); Roza (2004); Law (2006); Raczynski (2006). In the OR activities, the systems analyzed could be extraordinarily complex, e.g., industrial production models, business or governmental organizations, marketing models, national and world economic models, and military conflict models. These complex models commonly involve a strong coupling of complex physical processes, human behavior in a wide variety of conditions, and computer-controlled systems that adapt to changing system characteristics and environments. For such complex systems and processes, fundamental conceptual issues immediately arise with regard to (a) defining the system as opposed to its external influences, (b) issues of causality, (c) human behavior, (d) measuring system responses, and (e) assessing the accuracy of the model.

A key milestone in the early work by the OR community was the publication of the first definitions of V&V by the Society for Computer Simulation (SCS) in 1979 (Schlesinger, 1979).

Model verification: substantiation that a computerized model represents a conceptual model within specified limits of accuracy.

Model validation: substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model.

The SCS definition of verification, although brief, is quite informative. The main implication is that the computerized model, i.e., the computer code, must accurately mimic the model that was originally conceptualized. The SCS definition of validation, although instructive, appears somewhat vague. Both definitions, however, contain a critical concept: *substantiation* or evidence of correctness.

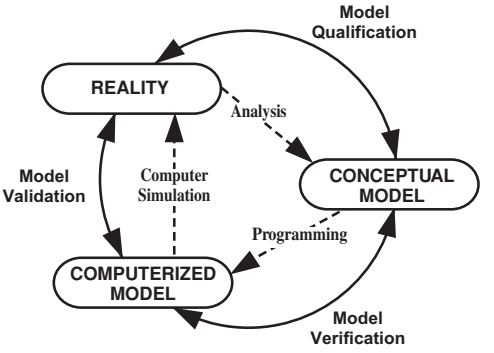

Figure 2.1 Phases of M&S and the role of V&V (Schlesinger, 1979 © 1979 by Simulation Councils, Inc.).

Along with these definitions, the SCS published the first useful diagram depicting the role of V&V within M&S (Figure 2.1).

Figure 2.1 identifies two types of model: a conceptual model and a computerized model. The *conceptual model* comprises all relevant information, modeling assumptions, and mathematical equations that describe the physical system or process of interest. The conceptual model is produced by analyzing and observing the system of interest. The SCS defined *qualification* as "Determination of adequacy of the conceptual model to provide an acceptable level of agreement for the domain of intended application." The *computerized model* is an operational computer program that implements a conceptual model using computer programming. Modern terminology typically refers to the computerized model as the computer model or code. Figure 2.1 emphasizes that *verification* deals with the relationship between the conceptual model and the computerized model and that *validation* deals with the relationship between the computerized model and reality. These relationships are not always recognized in other definitions of V&V, as will be discussed shortly.

The OR community clearly recognized, as it still does today, that V&V are tools for assessing the accuracy of the conceptual and computerized models. For much of the OR work, the assessment was so difficult, if not impossible, that V&V became more associated with the issue of credibility, i.e., whether the model is worthy of belief, or a model's power to elicit belief. In science and engineering, however, quantitative assessment of accuracy for important physical cases related to the intended application *is* mandatory. In certain situations, accuracy assessment can only be conducted using subscale physical models, a subset of the dominant physical processes occurring in the system, or a subsystem of the complete system. As will be discussed later in this chapter, the issue of extrapolation of models is more directly addressed in recent developments.

2.1.2 IEEE and related communities

During the 1970s, computer-controlled systems started to become important and widespread in commercial and public systems, particularly automatic flight-control systems for aircraft

and high-consequence systems, such as nuclear power reactors. In response to this interest, the Institute of Electrical and Electronics Engineers (IEEE) defined verification as follows (IEEE, 1984; IEEE, 1991):

> *Verification:* the process of evaluating the products of a software development phase to provide assurance that they meet the requirements defined for them by the previous phase.

This IEEE definition is quite general, but it is also strongly referential in the sense that the value of the definition directly depends on the specification of "requirements defined for them by the previous phase." Because those requirements are not stated in the definition, the definition does not contribute much to the intuitive understanding of verification or to the development of specific methods for verification. While the definition clearly includes a requirement for the consistency of products (e.g., computer programming) from one phase to another, the definition does not contain any indication of what the requirement for correctness or accuracy might be.

At the same time, IEEE defined validation as follows (IEEE, 1984; IEEE, 1991):

> *Validation*: the process of testing a computer program and evaluating the results to ensure compliance with specific requirements.

Both IEEE definitions emphasize that both V&V are processes, that is, ongoing activities. The definition of validation is also referential because of the phrase "compliance with specific requirements." Because specific requirements are not defined (to make the definition as generally applicable as possible), the definition of validation is not particularly useful by itself. The substance of the meaning must be provided in the specification of additional information.

One may ask why the IEEE definitions are included, as they seem to provide less understanding and utility than the earlier definitions of the SCS. First, these definitions provide a distinctly different perspective toward the entire issue of V&V than what is needed in scientific computing. The IEEE perspective asserts that because of the extreme variety of requirements for M&S, the requirements should be defined in a separate document for each application, not in the definitions of V&V. For example, the requirement of model accuracy as measured by comparisons with experimental data could be placed in a requirements document. Second, the IEEE definitions are the more prevalent definitions used in engineering as a whole. As a result, one must be aware of the potential confusion when other definitions are used in conversations, publications, government regulations, and contract specifications. The IEEE definitions are dominant because of the worldwide influence of this organization and the prevalence of electrical and electronics engineers. It should also be noted that the computer science community, the software quality assurance community, and the International Organization for Standardization (ISO) (ISO, 1991) use the IEEE definitions.

In addition, and more importantly for scientific computing, the IEEE definitions of V&V have been used by the American Nuclear Society (ANS) (ANS, 1987). However, in 2006 the ANS formed a new committee to reconsider their use of the IEEE definitions for V&V.

2.1.3 US Department of Defense community

In the early 1990s, the US Department of Defense (DoD) began to recognize the importance of putting into place terminology and procedures for V&V that would serve their very broad range of needs in M&S (Davis, 1992; Hodges and Dewar, 1992). The DoD tasked the Defense Modeling and Simulation Office (DMSO) to study the terminology put into place by the IEEE and to determine if the IEEE definitions would serve their needs. The DMSO obtained the expertise of researchers in the fields of OR, operational testing of combined hardware and software systems, man-in-the-loop training simulators, and warfare simulation. They concluded that the IEEE definitions were too restricted to software V&V instead of their need for much broader range of M&S. In 1994, the DoD/DMSO published their basic concepts and definitions of V&V (DoD, 1994).

Verification: the process of determining that a model implementation accurately represents the developer's conceptual description of the model.

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.

From a comparison of these definitions with those codified by the IEEE, it is clear there was a major break in conceptualization of V&V by the DoD. The DoD definitions could be referred to as *model V&V*, as opposed to the IEEE definitions of *software V&V*. The DoD definitions are actually similar to those formed by the SCS in 1979.

As noted in the discussion of the IEEE definitions, the DoD definitions also stress that both V&V are "process[es] of determining." V&V are ongoing activities that do not have a clearly defined completion point, unless additional specifications are given in terms of intended uses of the model and adequacy. The definitions include the ongoing nature of the process because of an unavoidable but distressing fact: the veracity, correctness, and accuracy of a computational model cannot be demonstrated for all possible conditions and applications, except for trivial models. For example, one cannot prove that even a moderately complex computer code has no errors. Likewise, models of physics cannot be proven correct; they can only be proven incorrect.

The key feature of the DoD definitions, which is not mentioned in the IEEE definitions, is the emphasis on *accuracy*. This feature assumes that a measure of accuracy can be determined. Accuracy can be measured relative to any accepted referent. In verification, the referent could be either well-accepted solutions to simplified model problems or expert opinions as to the reasonableness of the solution. In validation, the referent could be either experimentally measured data or expert opinions as to what is a reasonable or credible result of the model.

2.1.4 AIAA and ASME communities

Most science and engineering communities focus on their particular types of application, as opposed to the very broad range of DoD systems. Specifically, scientific computing

concentrates on modeling physical systems that have limited aspects of human interaction with the system. Typically, the mathematical model of the system of interest is dominated by physical processes that are described by partial differential equations (PDEs) or integro-differential equations. Human interaction with the system, as well as the effect of computer control systems, is explicitly given by way of the boundary conditions, initial conditions, system excitation, or other auxiliary submodels. The effect of this narrow focus of the science and engineering communities will be apparent in the further development of V&V terminology, concepts, and methods.

The computational fluid dynamics (CFD) community, primarily through the American Institute of Aeronautics and Astronautics (AIAA), was the first engineering community to seriously begin developing concepts and procedures for V&V methodology. Some of the key early contributors were Bradley (1988); Marvin (1988); Blottner (1990); Mehta (1990); Roache (1990); and Oberkampf and Aeschliman (1992). For a more complete history of the development of V&V concepts in CFD, see Oberkampf and Trucano (2002).

2.1.4.1 AIAA Guide

In 1992, the AIAA Committee on Standards for Computational Fluid Dynamics (AIAA COS) began a project to formulate and standardize the basic terminology and methodology in V&V for CFD simulations. The committee was composed of representatives from academia, industry, and government, with representation from the US, Canada, Japan, Belgium, Australia, and Italy. After six years of discussion and debate, the committee's project culminated in the publication of *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* (AIAA, 1998), referred to herein as the *AIAA Guide*. The AIAA Guide defines a number of key terms, discusses fundamental concepts, and specifies general procedures for conducting V&V in CFD.

The AIAA Guide (AIAA, 1998) modified slightly the DoD definition for verification, giving the following definition:

Verification: the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.

The DoD definition of verification did not make it clear that the accuracy of the numerical solution to the conceptual model should be included in the definition. Science and engineering communities, however, are keenly interested in the accuracy of the numerical solution – a concern that is common to essentially all of the fields in scientific computing.

Although the AIAA Guide adopted verbatim the DoD definition of *validation*, there are important differences in interpretation. These will be discussed in the next section, as well as in Section 2.2.3.

The fundamental strategy of verification is the identification, quantification, and reduction of errors in the computer code and the numerical solution. Verification provides evidence or substantiation that the conceptual (continuum mathematics) model is solved accurately by the discrete mathematics model embodied in the computer code. To quantify computer coding errors, a highly accurate, reliable benchmark solution must be available.

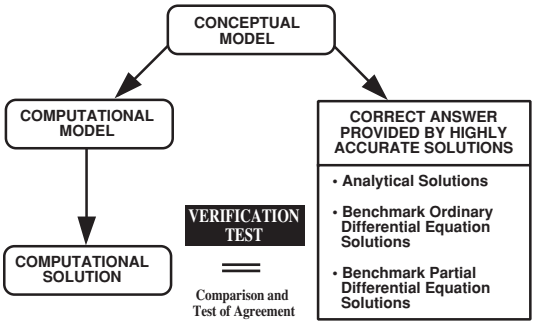

Figure 2.2 Verification process (AIAA, 1998).

Highly accurate solutions, unfortunately, are only available for simplified model problems. Verification does *not* deal with the relationship of the conceptual model to the real world. As Roache (1998) lucidly points out: "Verification is a mathematics issue; not a physics issue." *Validation* is a physical science issue. Figure 2.2 depicts the verification process of comparing the numerical solution from the code in question with various types of highly accurate solutions.

In the AIAA Guide, a significant break was made from the DoD perspective on validation in terms of the types of comparison allowed for accuracy assessment with respect to "the real world." The AIAA Guide specifically required that assessment of the accuracy of computational results be *only* allowed using experimental measurements. The fundamental strategy of validation involves identification and quantification of the error and uncertainty in the conceptual and mathematical models. This involves the quantification of the numerical error in the computational solution, estimation of the experimental uncertainty, and comparison between the computational results and the experimental data. That is, accuracy is measured in relation to experimental data, our best measure of reality. This strategy *does not* assume that the experimental measurements are more accurate than the computational results; it only asserts that experimental measurements are the most faithful reflections of reality for the purposes of validation. Figure 2.3 depicts the validation process of comparing the computational results with experimental data from various sources.

Because of the infeasibility and impracticality of conducting true validation experiments on most complex systems, the recommended method is to use a building block, or system complexity hierarchy, approach. This approach was originally developed by Sindir and his colleagues (Lin *et al.*, 1992; Sindir *et al.*, 1996), as well as Cosner (1995); and Marvin (1995). It divides the complex engineering system of interest into multiple, progressively simpler tiers; e.g., subsystem cases, benchmark cases, and unit problems. The strategy in the tiered approach is to assess how accurately the computational results compare with the experimental data (with quantified uncertainty estimates) at multiple degrees of physics coupling and geometric complexity (Figure 2.4). The approach is clearly constructive in that it (a) recognizes a hierarchy of complexity in systems and simulations, (b) recognizes that the quantity and accuracy of information obtained from experiments varies radically

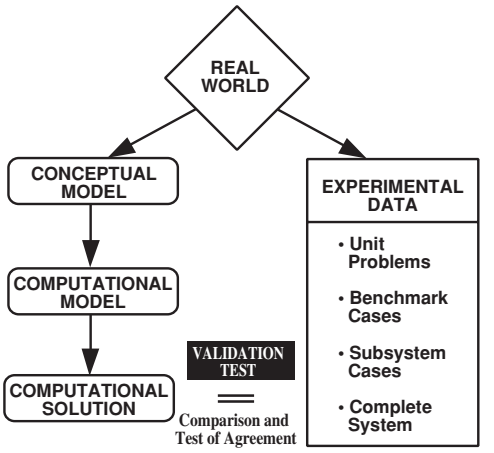

Figure 2.3 Validation process (AIAA, 1998).

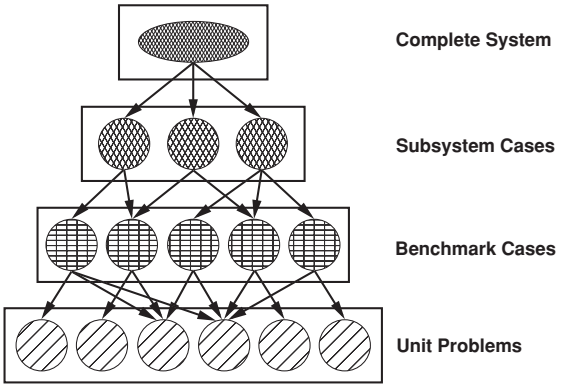

Figure 2.4 Validation tiers of a system hierarchy (AIAA, 1998).

over the range of tiers, and (c) guides the accumulation of validation evidence with the focus always being on the complete system. It should also be noted that additional building-block tiers beyond the four discussed here could be defined; however, additional tiers would not fundamentally alter the recommended methodology.

In the AIAA Guide's discussion of validation (depicted in Figure 2.3) is the concept that validation is the comparison of computational results with experimental measurements for the purpose of model accuracy assessment. Thinking of validation in this way requires one to then deal with the issue of *prediction*. The AIAA Guide gives the following definition:

Prediction: use of a computational model to foretell the state of a physical system under conditions for which the computational model has not been validated.

A prediction refers to the computational simulation of a specific case of interest that is *different* in some way from cases that have been validated. This definition differs from

the common-language meaning of prediction and refers only to *pre*diction, not *post*diction (replication of previously obtained results). If this restriction is not made, then one is only demonstrating previous agreement with experimental data in the validation database. The results of the validation process should be viewed as historical statements of model comparisons with experimental data. Stated differently, the validation database represents reproducible evidence that a model has achieved a given level of accuracy in the solution of specified problems. From this perspective, it becomes clear that validation comparisons do not directly make claims about the accuracy of predictions generally; rather, they allow inferences from the model concerning responses of similar systems. The issue of segregating validation (in the sense of model accuracy assessment) and inferred accuracy in prediction is a major conceptual issue that will resurface in several chapters.

2.1.4.2 ASME Guide

In the late 1990s, members of the solid mechanics community became interested in the concepts and methodology of V&V. The first V&V committee within the Codes and Standards branch of the American Society of Mechanical Engineers (ASME) was formed in 2001 and designated Performance Test Codes 60, Committee on Verification and Validation in Computational Solid Mechanics. Under the leadership of the committee chair, Leonard Schwer, the committee painstakingly debated and struggled with the subtleties of the terminology and appropriate methodology for V&V. Late in 2006, the *ASME Guide for Verification and Validation in Computational Solid Mechanics*, herein referred to as the *ASME Guide*, was completed (ASME, 2006).

The ASME Guide slightly modified the definition of verification as formulated by the AIAA Guide:

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

The ASME Guide adopted the definition of *validation* as formulated by the DoD and used by the AIAA Guide. Key issues of interpretation will be given below and in Section 2.2.3. Building on many of the concepts described in the AIAA Guide, in addition to newly published methods in V&V, the ASME Guide significantly extended the engineering standards literature in V&V.

Instead of graphically showing V&V as separate entities, as in the AIAA Guide, the AMSE Guide constructed a comprehensive diagram showing both activities, along with other complementary activities (Figure 2.5). It is important to recognize that the diagram and all of the activities shown can be applied to any tier of a system hierarchy. The analogous activities in both the mathematical modeling and the physical modeling are clearly shown, along with their interactions. The conceptual model, the mathematical model, and the computational model are all shown in the figure, as well as being defined in the ASME Guide. The separation of the concepts and activities in each of these three types of models significantly improved the understanding of not only the V&V process, but also the M&S process. Two elements of verification are identified: code verification and

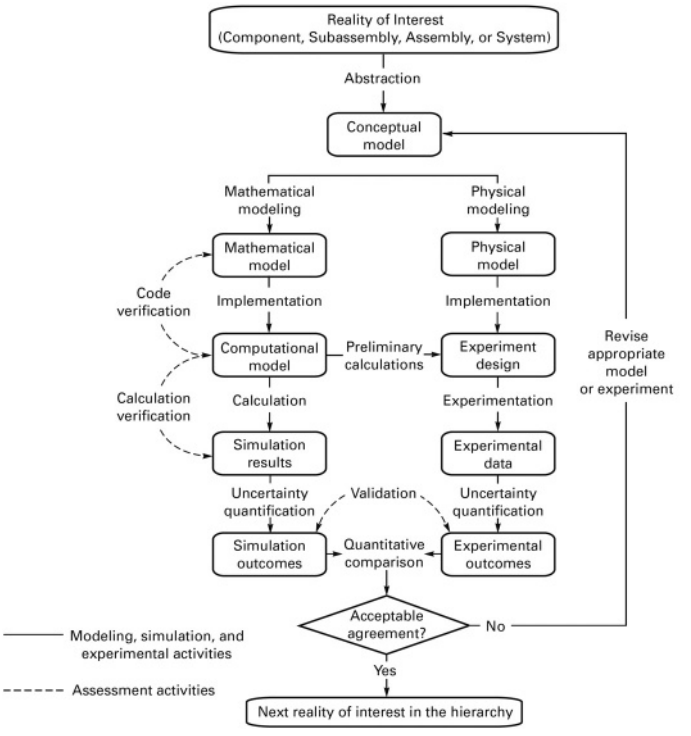

Figure 2.5 Verification and validation activities and products (ASME, 2006).

calculation verification. This separation of verification activities followed the pioneering work of Blottner (1990) and Roache (1995). With this separation of verification activities, a much clearer set of techniques could be discussed to improve coding reliability and numerical accuracy assessment of the computational model.

As shown at the bottom of Figure 2.5, an important decision point in the V&V activities is the answer to the question: Is there acceptable agreement between the computational results and the experimental measurements? The ASME Guide discusses how this decision should be made, specifically with regard to the key phrase in the definition of validation: "intended uses of the model." In some communities the phrase that is used is "fit for purpose," instead of "intended uses of the model." In the formulation of the conceptual model, several tasks are defined, among them: (a) identify which physical processes in the reality of interest are anticipated to have significant effects on the responses of interest and which processes are not expected to be important, (b) determine requirements for demonstrating the accuracy and predictive capability of the model, and (c) specify the model's domain of intended use. The specification of accuracy requirements for responses of interest predicted by the model allows the "acceptable agreement" question to be answered. Only with accuracy requirements can the decision be made whether to accept or revise a model. Without accuracy requirements, the question: "How good is good enough?" cannot be answered.

The emphasis in the specification of the model's domain of intended use deals with the operating conditions under which the model is to be used, e.g., range of boundary conditions, initial conditions, external system excitation, materials, and geometries. The ASME Guide, as well as the DoD community, recognizes the importance, and difficulty, of specifying the model's domain of intended use.

2.1.5 Hydrology community

The hydrology community, particularly surface and subsurface transport, has also been actively developing concepts and methods concerning V&V. Most of this work, however, has been essentially developed independently of many of the activities discussed earlier in this chapter. Some of the early key contributors to this work were Beck (1987); Tsang (1989); LeGore (1990); Davis *et al.* (1991); and Konikow and Bredehoeft (1992). The work of the hydrology community is significant for two reasons. First, it addresses validation for complex processes in the physical sciences where validation of models is extremely difficult, if not impossible. One reason for this difficulty is remarkably limited knowledge of the specific underground transport characteristics and material properties associated with the validation database. For such situations, one must deal more explicitly with calibration or parameter estimation in models instead of the concepts of validation developed by the AIAA and ASME. This critical issue of validation versus calibration of models will be dealt with in several chapters in this book. Second, because of the limited knowledge about the physical characteristics of the system under consideration, the hydrology field has strongly adopted statistical methods of calibration and validation assessment. In hydrology, it is not just calibration of scalar parameters, but also scalar and tensor fields. For a good review of the state of the art in hydrology V&V, see Anderson and Bates (2001).

In more recent work, the hydrology community in Europe (Rykiel, 1996; Beven, 2002; Refsgaard and Henriksen, 2004) has independently developed ideas about V&V that are very similar to those being developed in the United States. Rykiel (1996) makes an important practical point, especially to analysts and decision makers, about the difference between the philosophy-of-science viewpoint and the practitioner's view of validation: "Validation is not a procedure for testing scientific theory or for certifying the 'truth' of current scientific understanding. . . . Validation means that a model is acceptable for its intended use because it meets specified performance requirements." Refsgaard and Henriksen (2004) recommended terminology and fundamental procedures for V&V that are very similar to the AIAA Guide and ASME Guide. They define model validation as "Substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model." Refsgaard and Henriksen (2004) also stressed another crucial issue that is corroborated by the AIAA Guide and the ASME Guide: "Validation tests against independent data that have not also been used for calibration are necessary in order to be able to document the predictive capability of a model." In other words, the major challenge in validation is to perform an assessment of the model in a blind-test with experimental data, whereas the key issue in calibration is to adjust the physical modeling

parameters to improve agreement with experimental data. It is difficult, and sometimes impossible, to make blind-test comparisons; e.g., when well-known benchmark validation data are available for comparison. As a result, one must be very cautious in making conclusions about the predictive accuracy of models when the computational analyst has seen the data. Knowing the correct answer beforehand is extremely seductive, even to a saint.

2.2 Primary terms and concepts

This section will discuss in more detail the concepts and underlying principles behind the formal definitions of V&V. This book will use the definitions of V&V as given by the ASME Guide. Also defined and discussed in this section are the terms code verification, solution verification, predictive capability, calibration, certification, and accreditation. Definitions for these terms will be primarily drawn from ASME, AIAA, IEEE, and DoD.

The modern scientific method is very much aligned with the philosophy of nature approach referred to as deductivism. Deductivism is the method of drawing conclusions by logically combining new ideas with facts that have been accepted as true. Deductivism argues from the general to the particular, or reasons from known general principles to deduce previously unobserved or unknown phenomena. This perspective can be most clearly seen in the manner in which scientists and engineers are trained, as well as in mathematical modeling of physical processes. V&V, however, is aligned with inductive reasoning processes, i.e., processes that present the correctness of individual pieces of evidence to support the conclusion of correctness of the generalization. The philosophical perspective of V&V is one of *fundamental skepticism*: if any claim cannot be demonstrated or proven, then it is not accepted as true. The dichotomy of perspectives between the training of scientists and engineers, as opposed to the philosophy of V&V, is sometimes at the root of the lack of interest, or open resistance, to many V&V activities by some scientists and engineers.

2.2.1 Code verification

The ASME Guide (ASME, 2006) defines *code verification* as:

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

Code verification can be segregated into two activities: numerical algorithm verification and software quality assurance (SQA), as shown in Figure 2.6. Numerical algorithm verification addresses the mathematical correctness in the software implementation of all the numerical algorithms that affect the numerical accuracy of the computational results. The major goal of numerical algorithm verification is to accumulate evidence that demonstrates that the numerical algorithms in the code are implemented correctly and that they are functioning as intended. As an example, numerical algorithm verification would demonstrate that a spatial discretization method would produce the expected convergence rate, as the mesh is refined for the specific PDE being tested. The emphasis in SQA is on determining whether

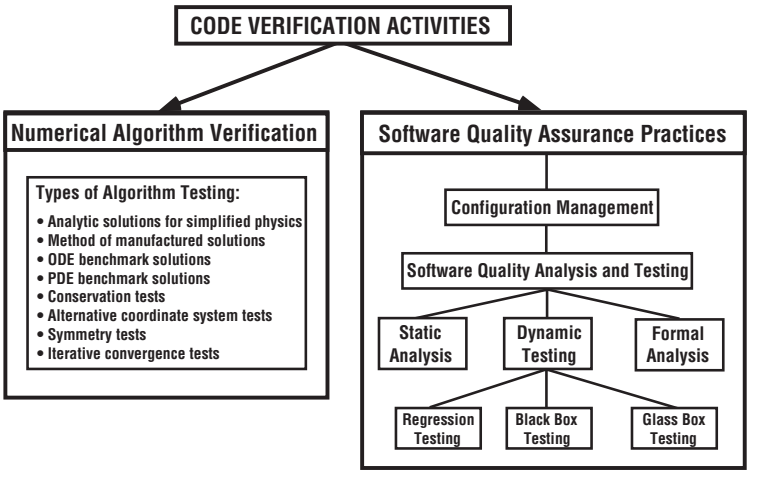

Figure 2.6 Integrated view of code verification in M&S (Oberkampf *et al*., 2004).

or not the code, as part of a software system, is implemented correctly and that it produces repeatable results on specified computer hardware and in a specified software environment. Such environments include computer operating systems, compilers, function libraries, etc. Although there are many software system elements in modern computer simulations, such as pre- and post-processor codes, focus in this book will be on SQA practices applied to the source code associated with scientific computing.

Numerical algorithm verification is fundamentally empirical. Specifically, it is based on testing, observations, comparisons, and analyses of code results for individual executions of the code. It focuses on careful investigations of numerical aspects, such as spatial and temporal convergence rates, spatial convergence in the presence of discontinuities, independence of solutions to coordinate transformations, and symmetry tests related to various types of boundary conditions (BCs). Analytical or formal error analysis is inadequate in numerical algorithm verification because the code itself must *demonstrate* the analytical and formal results of the numerical analysis. Numerical algorithm verification is usually conducted by comparing computational solutions with highly accurate solutions, which are commonly referred to as verification benchmarks. Oberkampf and Trucano (2007) divided the types of highly accurate solution into four categories (listed from highest to lowest in accuracy): manufactured solutions, analytical solutions, numerical solutions to ordinary differential equations, and numerical solutions to PDEs. Methods for numerical algorithm verification will be discussed in detail in Chapters 5 and 6.

SQA activities consist of practices, procedures, and processes that are primarily developed by researchers and practitioners in the computer science and software engineering communities. Conventional SQA emphasizes processes (i.e., management, planning, design, acquisition, supply, development, operation, and maintenance), as well as reporting, administrative, and documentation requirements. A key element or process of SQA

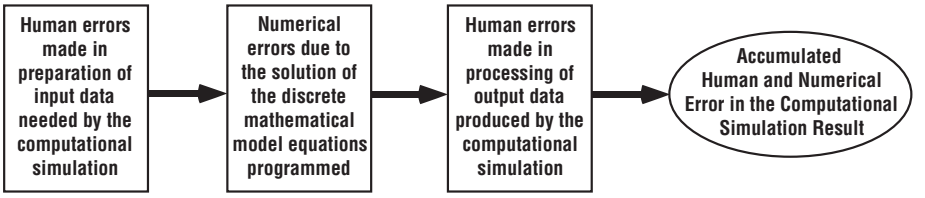

Figure 2.7 Error sources addressed in solution verification.

is software configuration management, which is composed of configuration identification, configuration and change control, and configuration status accounting. As shown in Figure 2.6, software quality analysis and testing can be divided into static analysis, dynamic testing, and formal analysis. Dynamic testing can be further divided into such elements of common practice as regression testing, black box testing, and glass box testing. From an SQA perspective, Figure 2.6 could be reorganized so that all types of algorithm testing categorized under numerical algorithm verification could be moved to dynamic testing. Although this perspective is useful, it fails to stress the importance of numerical algorithm verification that is critical in the numerical solution of PDEs. We stress that SQA is a necessary element of code verification. SQA methods will be discussed in Chapter 4, Software engineering for scientific computing.

2.2.2 Solution verification

Solution verification, also called calculation verification, is defined as:

Solution verification: the process of determining the correctness of the input data, the numerical accuracy of the solution obtained, and the correctness of the output data for a particular simulation.

Solution verification attempts to identify and quantify three sources of errors that can occur in the exercise of the computer simulation code (Figure 2.7). First, errors, blunders, or mistakes made by the computational analysts in preparation of the input for the computer simulation code. Second, numerical errors resulting from computing the discretized solution of the mathematical model on a digital computer. Third, errors, blunders or mistakes made by the computational analysts in any processing of the output data that is produced by the simulation code. The first and third sources of errors are of a very different type than the second. The first error source does *not* refer to errors or approximations made in the formulation or construction of the mathematical model. The first and third sources refer to human errors exclusive of any other sources. Human errors can be very difficult to detect in large-scale computational analyses of complex systems. Even in relatively small-scale analyses, human errors can go undetected if procedural or data-checking methods are not employed to detect possible errors. For example, if a solid mechanics analysis contains tens of CAD/CAM files, perhaps hundreds of different materials, and thousands of Monte Carlo

simulation samples, human errors, even by the most experienced and careful practitioners, commonly occur.

The second source, numerical solution errors, is primarily concerned with (a) spatial and temporal discretization errors in the numerical solution of PDEs, and (b) iterative solution errors usually resulting from a chosen solution approach to a set of nonlinear equations. There are other sources of numerical solution errors and these will be discussed in Chapter 3, Modeling and computational simulation. The importance and difficulty of numerical error estimation has increased as the complexity of the physics and mathematical models has increased, e.g., mathematical models given by nonlinear PDEs with singularities and discontinuities. It should be noted that the ASME Guide definition of calculation verification is not used in this book because it only refers to the second source of error, as opposed to all three sources.

The two basic approaches for estimating the error in the numerical solution of a PDE are *a priori* and *a posteriori* error estimation techniques. An *a priori* technique only uses information about the numerical algorithm that approximates the given PDE and the given initial conditions (ICs) and BCs. *A priori* error estimation is a significant element of classical numerical analysis for linear PDEs, especially in analyzing finite element methods. An *a posteriori* approach can use all the *a priori* information as well as the computational results from previous numerical solutions, e.g., solutions using different mesh resolutions or solutions using different order-of-accuracy methods. During the last decade it has become clear that the only way to achieve a useful quantitative estimate of numerical error in practical cases for nonlinear PDEs is by using *a posteriori* error estimates. Estimation of numerical solution errors will be discussed in detail in Chapters 8 and 9.

2.2.3 Model validation

Even though the DoD, the AIAA Guide, and the ASME Guide use the same formal definition for validation, our discussion in Section 2.1 hinted at differences in the interpretation and implications of the term. For example, it was pointed out that the AIAA Guide and the ASME Guide require *experimental measured data* when comparisons are made with simulations, whereas in the DoD interpretation this is not required. The recent paper of Oberkampf and Trucano (2008) clearly addressed the three aspects of validation and how different communities view each. Figure 2.8 depicts these three aspects as follows.

- Quantification of the accuracy of the computational model results by comparing the computed system response quantities (SRQs) of interest with experimentally measured SRQs.
- Use of the computational model to make predictions, in the sense of interpolation or extrapolation of the model, for conditions corresponding to the model's domain of intended use.
- Determination of whether the estimated accuracy of the computational model results satisfies the accuracy requirements specified for the SRQs of interest.

As depicted in Figure 2.8, Aspect 1 deals with assessing the accuracy of results from the model by comparisons with available experimental data. The assessment could be conducted

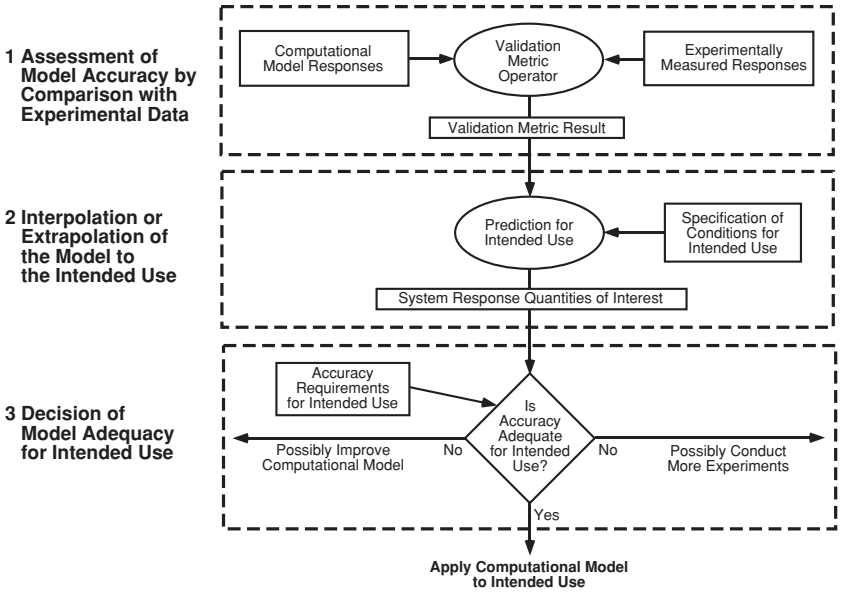

Figure 2.8 Three aspects of model validation (Oberkampf and Trucano, 2008).

for the actual system of interest or *any* related system. Some examples are (a) the actual system at operational conditions for the intended use of the system, (b) the actual system operating at lower than anticipated or less demanding conditions, or (c) subsystems or components of the actual system that have been identified in the system hierarchy. Model accuracy is quantitatively estimated using a *validation metric operator*. This operator computes a difference between the computational results and the experimental results for individual SRQs as a function of the input or control parameters in the validation domain. The operator can also be referred to as a *mismatch* operator between the computational results and the experimental results over the multidimensional space of all input parameters. In general, it is a statistical operator because the computational results and the experimental results are not single numbers but distributions of numbers (e.g., a cumulative distribution function) or quantities that are interval valued. This topic will be discussed in detail in Chapter 13, Model accuracy assessment.

Aspect 2 deals with a fundamentally and conceptually different topic: use of the model to make a prediction. As noted earlier, the AIAA Guide defined *prediction* (AIAA, 1998) as "foretelling the response of a system under conditions for which the model has not been validated." Prediction can also be thought of as interpolating or extrapolating the model beyond the specific conditions tested in the validation domain to the conditions of the intended use of the model. Several other authors have stressed the important aspect of extrapolation of the model and the attendant increase in uncertainty, which is usually referred to as *model form uncertainty*. See, for example, Cullen and Frey (1999) or Suter (2007). The important issue here is the estimated total uncertainty in the SRQs of interest as

a function of (a) the (in)accuracy in the model that was observed over the validation domain, and (b) the estimated model input parameters; both of these at the specified conditions of the intended use of the model. Stated differently, Aspect 2 does *not* deal with aspects of adequacy or accuracy requirements on the prediction, but focuses on the uncertainty in the SRQs of interest for the applications conditions of interest. The estimated total uncertainty is due to a wide variety of sources, such as inherent uncertainty in the system, lack of knowledge concerning the conditions of the intended use of the system, and model form uncertainty. The basic concepts in the topic of predictive uncertainty estimation will be summarized in Chapter 13, Predictive capability. See, for example, Morgan and Henrion (1990); Kumamoto and Henley (1996); Cullen and Frey (1999); Ayyub and Klir (2006); and Suter (2007).

Aspect 3 deals with (a) the comparison of the estimated accuracy of the model relative to the accuracy requirements of the model for the domain of the model's intended use, and (b) the decision of adequacy or inadequacy of the model over the domain of the model's intended use. The more general assessment of model adequacy or inadequacy typically depends on many factors, such as computer resource requirements, speed with which re-meshing can be done for a new geometry, and ease of use of the software for the given experience level of the analysts involved. The validation decision in Aspect 3 *only* refers to whether the model satisfies or does not satisfy the accuracy requirements specified. An accuracy requirement may be stated as: the estimated maximum allowable model form uncertainty for specified SRQs cannot exceed a fixed value over the domain of the model's intended use. The model form uncertainty will be a function of the input parameters describing the model's intended use, but model accuracy can also depend on uncertainty in the parameters themselves. The maximum allowable uncertainty over the parameter range of the intended use of the model would typically be an absolute-value quantity (i.e., the uncertainty cannot exceed a specified value) or a relative-uncertainty quantity (i.e., the uncertainty is scaled by the magnitude of the quantity). There are two types of *yes* decision that could occur in Aspect 3: (a) the estimated uncertainty is less than the maximum allowable uncertainty over the parameter range of the model's intended use, or (b) the parameter range of the model's intended use must be modified, e.g., restricted, such that the estimated uncertainty does not exceed the maximum allowable uncertainty.

A final important conceptual point should be mentioned in regard to Aspect 3. The decision governed by model adequacy assessment is only concerned with the adequacy of the computational model, *not* the performance of the engineering system being analyzed. Whether the system of interest, e.g., a gas turbine engine or a flight vehicle, meets its performance, safety, or reliability requirements is, of course, a completely separate topic from the aspects discussed relative to Figure 2.8. Simply put, a computational model of a system could be accurate, but the system itself could be lacking in performance, safety, or reliability because of inadequate design.

Understanding that there are three aspects to the term *validation* presented in Figure 2.8, there are two viewpoints underlying interpretation of the term. One interpretation is what is

called the *encompassing view of validation*. When employing this perspective, one means *all three aspects* discussed above. The DoD community usually takes the encompassing view of validation, although there is commonly confusion on this issue. The *restricted view of validation* considers each aspect of validation separately. That is, Aspect 1 is referred to as validation assessment, model accuracy assessment, or model validation. Aspect 2 is referred to as model prediction, predictive capability, or model extrapolation. Aspect 3 is referred to as model adequacy assessment or adequacy assessment for intended use. The AIAA Guide takes the restricted view of validation. The ASME Guide generally takes the encompassing view of validation, but in a few sections of this Guide, the concepts can only be understood using the restricted view of validation.

Either interpretation can be used in validation activities related to scientific computing. However, it is our view, and the experience of many, that an encompassing view of validation commonly leads to misunderstandings and confusion in discussions and in communication of computational results. The primary reason for this confusion is the dissimilarity between each of the three aspects discussed above. Misunderstandings and confusion can be particularly risky and damaging, for example, in communication of computational results to system designers, project managers, decision makers, and individuals not trained in science or engineering. As a result, *this book will use the restricted view of validation*. For this restricted view of validation, the terms *model validation*, *validation assessment*, and *model accuracy assessment* will also be used; all referring only to Aspect 1.

One term that has been used extensively is *model*, although this term has not yet been carefully defined. As is well known, there are many types of model used in scientific computing. The three major types of model are conceptual, mathematical, and computational. A *conceptual model* specifies (a) the physical system, the system surroundings, and the phenomena of interest, (b) the operating environment of the system and its domain of intended use, (c) the physical assumptions that simplify the system and the phenomena of interest, (d) the SRQs of interest, and (e) the accuracy requirements for the SRQs of interest. A *mathematical model* is derived from the conceptual model, and it is a set of mathematical and logical relations that represent the physical system of interest and its responses to the environment and the ICs of the system. The mathematical model is commonly given by a set of PDEs, integral equations, BCs and ICs, material properties, and excitation equations. A *computational model* is derived from the numerical implementation of the mathematical model, a process that results in a set of discretized equations and solution algorithms, and then these equations and algorithms are programmed into a computer. Another way to describe the computational model is that it is a mapping of the mathematical model into a software package that, when combined with the proper input, produces simulation results. Commonly the computational model is simply referred to as the *code*. The different types of model will be discussed in detail in Chapter 3.

When the term *model validation* is used, one is actually referring to validation of the mathematical model, even though the computational model results are compared with experimental data. The essence of what is being assessed in validation and the essence of what is making a prediction is embodied in the mathematical model. Viewing model

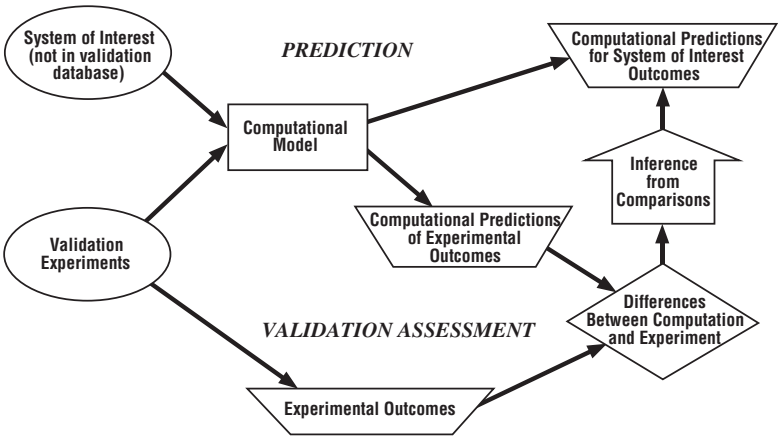

Figure 2.9 Relationship of model validation to prediction (Oberkampf and Trucano, 2002).

validation as *mathematical model validation* fundamentally relies on the following assumptions: (a) that the numerical algorithms are reliable and accurate, (b) the computer program is correct, (c) no human procedural errors have been made in the input or output for the simulation, and (d) the numerical solution error is small. Evidence for the veracity of these assumptions must be demonstrated by the activities conducted in code verification and solution verification.

2.2.4 Predictive capability

This book will use the definition of the term *prediction* as given by the AIAA Guide (AIAA, 1998):

Prediction: use of a computational model to foretell the state of a physical system under conditions for which the computational model has not been validated.

As discussed earlier, this definition is very specific and restrictive compared to common-language usage. The meaning of predictive capability is depicted as Aspect 2 of Figure 2.8, i.e., extrapolation or interpolation of the model to specific conditions defined by the intended use of the model. The results of the model validation process, Aspect 1, should be viewed as reproducible evidence that a model has achieved a given level of accuracy in the solution of specified problems. The evidence compiled allows inferences to be made concerning similar systems exposed to similar conditions. The strength of the inference depends on the explanatory power of the model as opposed to the descriptive power of the model. The suggested relationship between model validation and prediction is shown in Figure 2.9.

Figure 2.9 attempts to capture the distinction between model validation and prediction. The bottom portion of the figure represents the model validation process. Although it is not readily apparent, the validation process shown in Figure 2.9 is fundamentally the same as

that shown in Figure 2.3. In Figure 2.9, the block Validation Experiments produces one or more physical realizations of the "real world." The Experimental Outcomes are the experimental data measured in the experiment. The physical conditions from the actual validation experiments, i.e., model input parameters, initial conditions, and boundary conditions, are input to the Computational Model, which produces the Computational Results of Experimental Outcomes. These results are then compared with the experimentally determined outcomes in the block Differences Between Computation and Experiment. This block was referred to as Validation Metric Operator in Figure 2.8. Based on the magnitude of these differences in quantities of interest and on the depth of understanding of the physical process, an Inference from Comparisons is made.

The upper portion of Figure 2.9 represents the prediction process. The System of Interest should drive the entire scientific computing process, but some of the realizations of interest, i.e., predictions, are commonly not in the validation database. That is, when a physical realization is conducted as part of the validation database, regardless of the validation tier as discussed in Section 2.1.4 above, the realization becomes part of the Validation Experiments. Predictions for conditions of interest are made using the Computational Model, resulting in Computational Predictions of System of Interest Outcomes. The confidence in these predictions is determined by (a) the strength of the Inference from Comparisons, (b) the similarity of the complex system of interest to the existing validation experiments, and (c) the depth of understanding of the physical processes involved, i.e., the explanatory power of the mathematical model.

The process of logical and mathematical inference of accuracy of a computational model stemming from its associated validation database is analogous to similar processes and conclusions for classical scientific theories. However, the strength or confidence in the inference from scientific computing is, and should be, much weaker than traditional scientific theories. Computational simulation relies on the same logic as science theories, but it also relies on many additional issues that are not present in traditional scientific theories, such as, code verification, solution verification, and extrapolation of models that have varying degrees of calibrated parameters. One of the key theoretical issues is the state of knowledge of the process being modeled. Bossel (1994); Zeigler *et al.* (2000); and Roza (2004) give a discussion of hierarchical levels of knowledge of a system. For physical processes that are well understood both physically and mathematically, the inference can be quite strong. For complex physical processes, the inference can be quite weak. A general mathematical method for determining how the inference degrades as the physical process becomes more complex and less well understood has not been formulated. For example, in a complex physical process how do you determine *how nearby* the prediction case is from cases in the validation database? This could be viewed as a topological question in some type of high-dimensional space composed of both model form uncertainty and parameter uncertainty. Struggling with the rigorous specification of the strength or quantification of the inference in a prediction is, and will remain, an important topic of research (Bossel, 1994; Chiles and Delfiner, 1999; Zeigler *et al.*, 2000; Anderson and Bates, 2001).

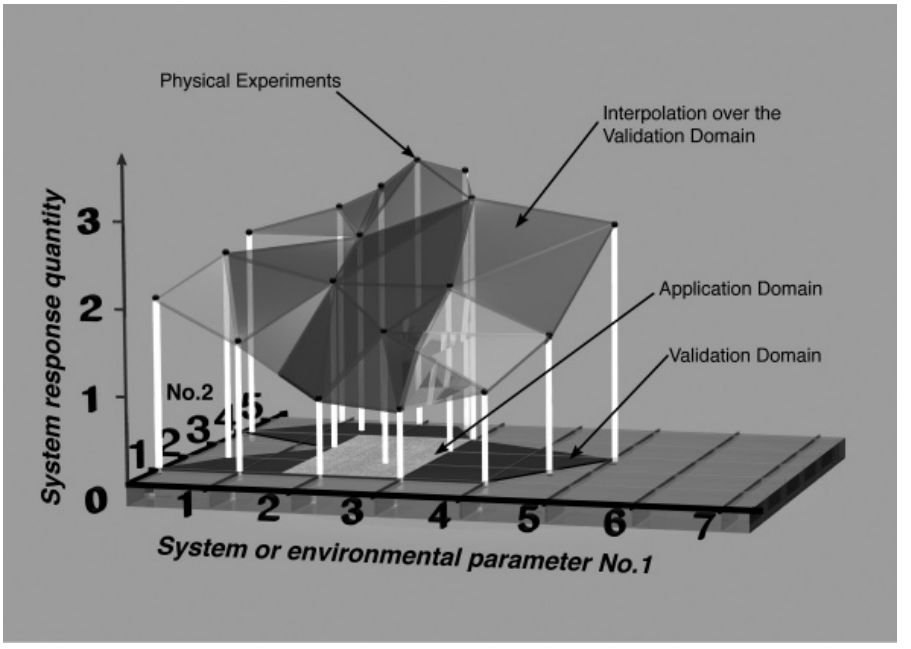

Figure 2.10 Possible relationships of the validation domain to the application domain. See color plate section. (a) Complete overlap of the application domain and the validation domain.

To better explain the important relationship of prediction to model validation, consider Figure 2.10. The two horizontal axes of the figure are labeled *system or environmental parameters No. 1 and No. 2*. These are parameters in the model of a physical system that typically come from the system itself, the surroundings, or the environment in which the system is operating. Examples of these parameters are (a) initial speed and angle of impact of an automobile in a crash environment, (b) Mach number and Reynolds number in a gas dynamics problem, (c) amplitude and frequency of vibrational excitation of a structure, and (d) damaged state of a system exposed to an accident or hostile environment. The vertical axis is the SRQ of interest. In most computational analyses, there is typically a group of SRQs of interest, each depending on several system or environmental parameters.

The values of the system or environmental parameters in the physical experiments are shown in Figure 2.10 as points, at the bottom of the white pillars, in the two dimensional space of the system response and system/environmental parameters. The validation domain is defined by the boundary of the physical experiments that have been conducted (the maroon colored region, including the interior region noted by the tan color). The tan-colored rectangular region indicates the application domain. The experimental measurements of the SRQ are shown as the black dots at the top of each white pillar. The SRQ over the validation domain is indicated as a response surface constructed using a piecewise linear interpolation (the light-blue colored surface). The SRQ over the application domain is indicated as the response surface colored either purple or green. For the purpose of discussion here,

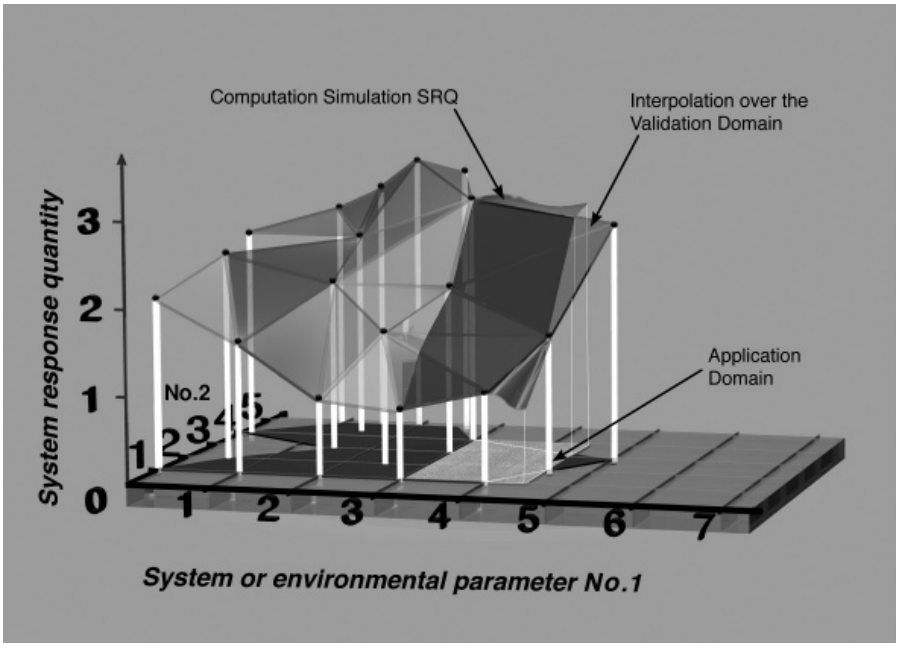

Figure 2.10(b) Partial overlap of the application domain and the validation domain.

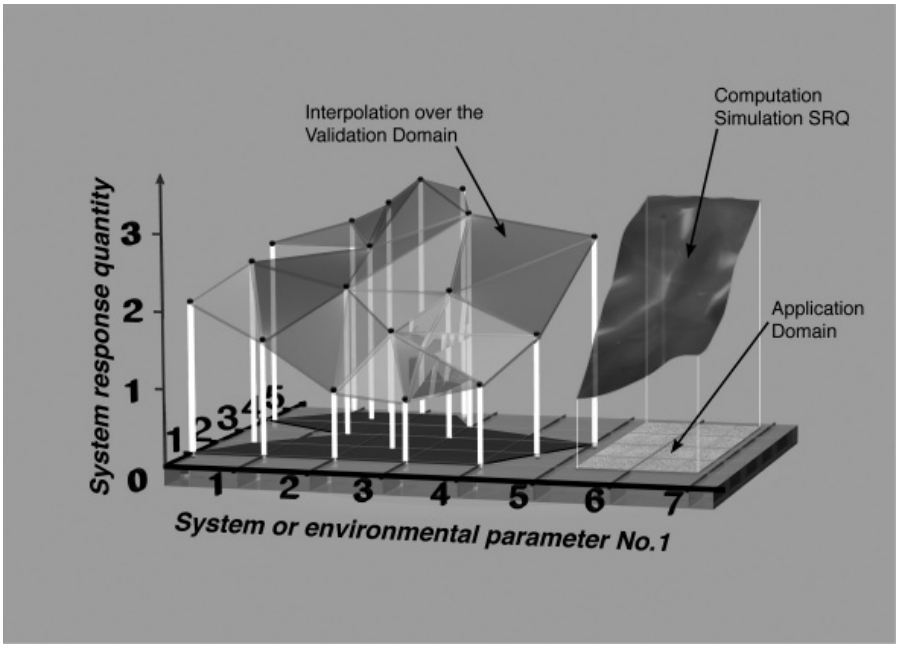

Figure 2.10(c) No overlap of the application domain and the validation domain.

we presume the following three features concerning the experimental and computational results obtained over the validation domain. First, in this region there is high confidence that the relevant physics is understood and modeled at a level that is commensurate with the needs of the application. Second, this confidence has been quantitatively demonstrated by satisfactory agreement between computations and experiments over the validation domain. Third, outside the validation domain there are physical and statistical reasons to expect degradation in confidence in the quantitative predictive capability of the model. Stated differently, if the model is physics-based, then the model has some credibility outside the validation domain. However, its quantitative predictive capability has not been assessed and, therefore, can only be estimated by extrapolation.

Figure 2.10a depicts the prevalent and desirable situation in engineering; that is, the complete overlap of the validation domain with the application domain. In this region the SRQ can be computed from interpolation of either the experimental measurements or the computational results, whichever is judged to be more accurate and/or reliable. The vast majority of modern engineering system design is represented in Figure 2.10a. Stated differently, engineering systems over the centuries have been predominately designed, and their performance determined, based on experimental testing.

Figure 2.10b represents the common engineering situation where there is significant overlap between the validation domain and the application domain. There are, however, portions of the application domain outside the validation domain, shown in green in Figure 2.10b. These regions primarily rely on extrapolation of the model to predict the SRQs of interest. Here we are *not* dealing with the question of whether the validation domain can or cannot be extended to include the application domain shown in green. Keep in mind that the number of the system or environmental parameters in a real engineering system commonly numbers ten to hundreds. For this high dimensional space, it is very common for portions of the application domain to be outside the validation domain in at least some of the parameter dimensions. In fact, in high dimensional spaces, it becomes very difficult to even determine if one is within a hyper-volume or not. Some examples of significant overlap of the validation domain and the application domain are: prediction of the crash response of automobile structures and occupants at conditions slightly different from the test database, prediction of aerodynamic drag on a vehicle design that is somewhat different than the test database on existing vehicles, and prediction of the performance of a gas turbine engine for flight conditions that are similar to, but not exactly attainable, using existing test facilities.

Figure 2.10c depicts the situation where there is not only no overlap between the validation domain and the application domain, but the application domain is *far* from the validation domain. This situation necessitates model extrapolation well beyond the demonstrated physical understanding and the statistical knowledge gained from the experimental data. Some examples are: entry of a spacecraft probe into the atmosphere of another planet; prediction of the fracture dynamics of an aircraft engine fan cowling made of new materials under operational conditions, such as the loss of a fan blade; and prediction of steam explosions in a severe accident environment of a nuclear power plant. For many high-consequence

systems, predictions are in this realm because experiments cannot be performed for closely related conditions. For the case suggested in Figure 2.10c, the strength of inference from the validation domain must rely primarily on the fidelity of the physics embodied in the model. The need to perform this extrapolation reinforces our need for models to be critically judged on the basis of achieving the right answers for the right reasons in the validation regime.

A detailed discussion of the procedural steps used in developing a predictive capability is given in Chapter 13.

2.2.5 Calibration

The ASME Guide (ASME, 2006) gives the definition of model *calibration* as:

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

Calibration is primarily directed toward improving the agreement of computational results with existing experimental data, *not* determining the accuracy of the results. Model calibration is also referred to as model updating or model tuning. Because of technical issues (such as limited experimental data or poor understanding of the physics), or practical issues (such as constraints in program schedules, fiscal budgets, and computer resources), calibration is a more appropriate process than is validation. If one were to concentrate on experiments and simulations for only one component or unit problem, then the distinction between calibration and validation would usually be clear. However, if one examines a complete system, it is found that some elements of the validation hierarchy involved calibration and some are focused on validation. As a result, both model calibration and validation commonly occur during different phases of the computational analysis of a complete system. Attempts should be made to recognize when calibration is done simply for expediency because it directly impacts the confidence in predictions from the model. Calibration of model parameters typically confounds a variety of weaknesses in a model, thereby resulting in decreased predictive capability of the model. *How* model calibration impacts the confidence in predictive capability is very difficult to determine and is presently an active research topic.

Model calibration can be considered as part of the broader field of parameter estimation. Parameter estimation refers to procedures for estimating any type of parameter in a model using supplied data, e.g., experimentally measured data or computationally generated data. The estimated parameter can be either a deterministic value, such as a single value determined by some optimization process, or a nondeterministic value, such as a random variable.

Calibration is generally needed in the modeling of complex physical processes, when one must deal with incomplete or imprecise measurements in the experiments, or when physical parameters cannot be directly measured in an experiment. Examples of technical fields that commonly use model calibration are multi-phase fluid flow, structural dynamics, fracture

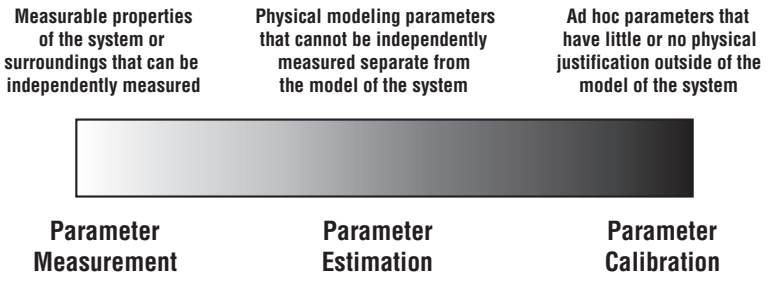

Figure 2.11 Spectrum of parameter measurement, estimation, and calibration.

mechanics, meteorology, hydrology, and reservoir engineering. Sometimes the parameters requiring calibration result from a phenomenological model or approximations made in mathematical modeling, e.g., effective quantities. Phenomenological models are those that express mathematically the results of observed phenomena without paying detailed attention to the physical causes. These types of model are commonly used as submodels in describing complex physical processes. Quite often the parameters that need calibration are not independently or physically measurable at all, but only exist as adjustable parameters in a mathematical model. Although the definition of calibration refers to a parameter, the parameter can be a scalar, a scalar field, a vector, or a tensor field.

Because of the wide range in which calibration and parameter estimation can enter a simulation, these procedures should be considered as a spectrum of activities. Figure 2.11 shows a three-level spectrum that can constructively segregate these different activities. At the left, more confident, end of the spectrum one has *parameter measurement*. By this we mean the determination of physically meaningful parameters that can be measured, in principle, using simple, independent models. There are many physical parameters in this category, for example: mechanical properties, such as Young's modulus, tensile strength, hardness, mass density, viscosity, and porosity; electrical properties, such as electrical conductivity, dielectric constant, and piezoelectric constants; thermal properties, such as thermal conductivity, specific heat, vapor pressure, and melting point; and chemical properties, such as pH, surface energy, and reactivity. In the middle of the spectrum one has *parameter estimation*. By this we mean the determination of physically meaningful parameters that can only, in practice, be determined using a complex model. Some examples are (a) internal dynamic damping in a material, (b) aerodynamic damping of a structure, (c) damping and stiffness of assembled joints in a multi-element structure, (d) effective reaction rate in turbulent reacting flow, and (e) effective surface area of droplets in multiphase flow. At the right end of the spectrum one has *parameter calibration*. By this we mean adjustment of a parameter that has little or no physical meaning outside of the model in which it is used. Some examples are (a) most parameters in fluid dynamic turbulence models, (b) parameters obtained by regression fits of experimental data, and (c) ad hoc parameters that are added to a model to simply obtain agreement with experimental data.

The spectrum shown in Figure 2.11 can aid in judging the physical soundness and trustworthiness of how parameters are determined. As one moves to the right in this spectrum, the confidence in extrapolating the model decreases significantly. Stated differently, the uncertainty in a prediction increases rapidly as one extrapolates a model that has heavily relied on parameter estimation and especially calibration. Concerning parameter adjustment in models versus blind prediction of models, Lipton made a graphic comparison:

> Accommodation [calibration] is like drawing the bull's-eye afterwards, whereas in prediction the target is there in advance (Lipton, 2005).

There are situations where the spectrum shown in Figure 2.11 is distorted because of the procedure that is used to determine a parameter. That is, when certain parameter adjustment procedures are used that are normally considered as parameter measurement or parameter estimation, one can cause a fundamental shift toward the parameter calibration category. Some examples are:

- well-known, physically meaningful parameters are changed simply to obtain agreement with newly obtained system-level experimental measurements;
- parameters are readjusted when unrelated submodels are changed;
- parameters are readjusted when spatial mesh refinement or discretization time step are changed;
- parameters are readjusted when numerical algorithms are changed;
- parameters are readjusted when a code bug is eliminated, and the code bug had nothing to do with the parameters being adjusted.

Such things as convenience, expediency, excessive experimental and simulation costs, and project schedule requirements commonly induce the above listed examples.

Consider the following three examples to help clarify the issues involved in parameter measurement, parameter estimation, and calibration. First, suppose one is interested in determining Young's modulus, also known as the modulus of elasticity of a material, in solid mechanics. Young's modulus, E, is defined as

$$E = \frac{\text{tensile stress}}{\text{tensile strain}}. \tag{2.1}$$

An experiment is conducted in which the tensile stress and tensile strain are measured over the linear elastic range of the material and then a value for E is computed. Although a mathematical model is used to define E, it would be inappropriate to say that E is calibrated because the physics of the process is very well understood. The appropriate term for this activity would be *measurement* of E. If a large number of material samples were drawn from some production batch of material, then parameter estimation methods could be used to characterize the variability of the production batch. The result would then be given as a probability distribution to describe the variability in E.

Second, suppose that a structural dynamics simulation is conducted on a structure that is constructed from several structural members and all of these members are bolted together. All of the structural members are made from the same batch of material as the previous experiment to measure E, which is needed in the simulation. A finite element model is

made for the structure and an experiment is conducted in which the structure is excited over the linear range. Various vibration modes of the structure are measured and a parameter optimization procedure is used to determine the joint stiffness and damping in the mathematical model that results in the best match of the experimental data. Assume that all of the attachment joints are of the same design and the pre-load torque on all of the bolts is the same. This procedure to determine the stiffness and damping in the bolted joints is referred to as *parameter estimation*. It is obvious that these two parameters, joint stiffness and damping, cannot be measured independently from the model for the vibration of the structure, i.e., the structural members must be bolted together before the structure exists. As a result, the term *parameter estimation* would properly characterize this procedure in the spectrum shown in Figure 2.11.

Third, consider a similar structural dynamics simulation to before, but now the geometry of the structure is more complex with many structural members of varying thicknesses and cross-sections, all bolted together. All of the structural members, however, are made from the same batch of material as the above experimental measurement of E. If the value of E in the simulation of the vibration of this structure were allowed to be an adjustable parameter, then E would be considered as a *calibrated* parameter. That is, the parameter is allowed to change simply due to expediency in the simulation. For this simple example, there is no physical reason to claim that E has changed. Confidence in the predictive capability of the calibrated model could be seriously questioned and the uncertainty in the predictions for similar structures would be difficult to estimate.

A more detailed discussion of the more common calibration procedures is given in Chapters 12 and 13.

2.2.6 Certification and accreditation

The IEEE (IEEE, 1991) defines *certification* as:

Certification: a written guarantee that a system or component complies with its specified requirements and is acceptable for operational use.

For our purposes, the "system or component" will be considered either a model, a code, or a simulation. For simplicity, all of these will be referred to as an *entity*. In certification of an entity, the written guarantee of acceptable performance can be generated by anyone who is willing to accept the responsibility or legal liability associated with the guarantee. Model developers, code developers, code assessors, or organizations could provide the written guarantee required for certification. For example, a national laboratory, a governmental organization, or a commercial code company could certify their own codes. The documentation for certification is normally done in a more formal manner than is the documentation for model validation. Thus, the team or organization conducting the certification would provide the detailed documentation for the simulations conducted, the experimental data used in the test cases, and the results from comparing simulations with highly-accurate solutions and experiments.

The DoD (DoD, 1994; DoD, 1996; DoD, 1997) defines *accreditation* as:

Accreditation: the official certification that a model or simulation is acceptable for use for a specific purpose.

The definition of accreditation uses the phrase "model or simulation," whereas the definition of certification uses the phrase "system or component." This, however, is not the crux of the difference between these two terms. The fundamental difference between the terms certification and accreditation is the phrase "written guarantee" versus "official certification" in certification and accreditation, respectively. As one might suspect, these terms suggest that the focus is changing from technical issues to legal, control authority, and liability issues when moving from certification to accreditation. Note that the DoD does not formally use the term *certification*, and the IEEE does not formally use the term *accreditation*.

In accreditation, *only* officially designated individuals or organizations can provide the guarantee of "acceptable for use for a specific purpose." Typically, the customer (or potential customer) or a separate legal representative has the authority to select the individual or organization that can accredit the entity. The accrediting authority is *never* the entity developers, anyone from the developers' organization, or anyone else who might have a vested interest in the performance, accuracy, or sale of the entity. Considering high-consequence public safety risks and environmental impact, one can make a convincing argument that accreditation of entities is plainly needed. The fundamental difference between accreditation and certification is the level of authority, independence, and responsibility to guarantee the performance or accuracy of the entity. In addition, when compared to certification, the accreditation of an entity is generally more formal, involves more in-depth entity testing, and requires more extensive documentation. Note that commercial software companies *never* make any statement of certification or accreditation. In fact, the *Conditions of Use* statement that one must agree to specifically states, "No warranty is expressed or implied with this product." It is doubtful, however, this would absolve the software company of complete legal liability.

Certification and accreditation can also be viewed as increasing levels of independence of assessment in V&V activities. A number of researchers and practitioners over a wide range of fields of scientific computing have pointed out the importance and value of independent V&V (see, e.g., Lewis, 1992; Gass, 1993; Arthur and Nance, 1996). The levels of independence in the V&V assessment of scientific computing entities can be viewed as a continuum (Figure 2.12). The least independent evaluation, i.e., no independence, occurs when the entity developer conducts assessment activities. Essentially all research activities are conducted at this first level of assessment. Some observers may question the adequacy of the first-level assessment, except possibly the developer. Only with some level of independence and objectivity of the assessors can one have the proper perspective for critical appraisal. For example, it is common that the developer's ego, his/her professional esteem or reputation, or the public image or future business opportunities of the sponsoring organization are intertwined with the entity. Evaluation only by developers is *never*

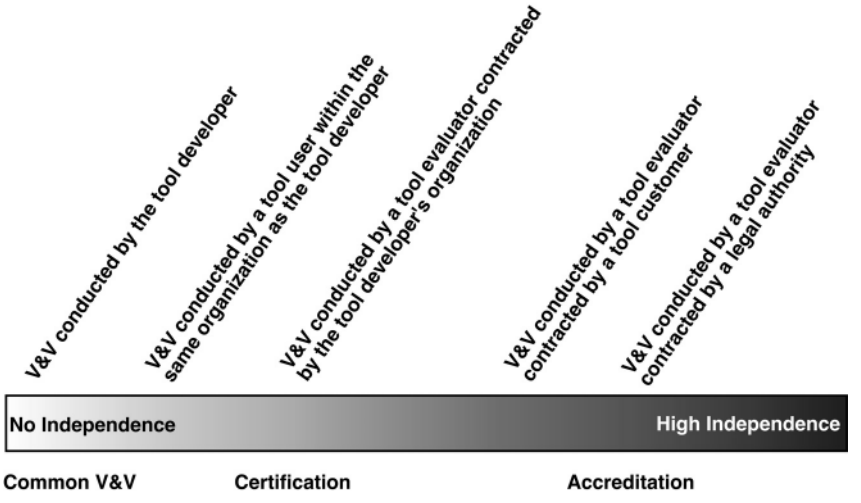

Figure 2.12 Spectrum of independence of V&V levels applied to scientific computing entities.

recommended for any production or commercial entity, or any computational results that can have significant organizational, safety, or security impact.

At the second level, the V&V evaluation is conducted by a user of the entity who is in the same or a closely related organization as the developer of the entity. Thus, a user can also be an entity evaluator, but the user *cannot* be one of the developers. This level of independence in evaluation is a major step commonly not appreciated by the management of research organizations that develop scientific computing entities. The entity evaluator at this second level can have various degrees of independence from the entity developer. If the entity evaluator is in the same group or team, e.g., the lowest-level organizational unit, then the independence of the entity evaluator is marginal. If the entity evaluator is in a group separated laterally by two or three lines of management from the entity developer's management, then the evaluation has much improved independence. For example, the entity evaluator could be a potential user of the entity in a design group that conducts computational analyses for product design or manufacturing processes. It is suggested that the minimum level of evaluation independence that should be considered for certification is separation of the entity developer and the entity evaluator by two to three lines of management.

At the third level, the V&V evaluation is conducted by an entity evaluator who is contracted by the entity developer's organization. This level of evaluation typically provides considerable independence for the entity evaluator because an external contractor is commonly hired for the task. At this level, and the higher levels to be discussed, information concerning the credentials of the contractor should be obtained to ensure that the contractor is objective in the evaluation and has the proper expertise for the evaluation. Occasionally, a monetary bonus is paid to the contractor if the contractor demonstrates exceptional thoroughness and vigor in evaluating the entity. For example, the contractor may be paid a bonus for each coding error, data input error, or failure to meet a specification. If national

security classification or extraordinary proprietary issues are a concern, the evaluator could be employed from a subsidiary or sister organization. For this case, the evaluator would not have any organizational connection to the entity developer's organization or to the anticipated users of the entity. This level of independent V&V provides fresh perspectives on the entity's performance, robustness, applicability, and reliability. In addition, this level of independent V&V commonly provides helpful and constructive ideas for significant improvements in the entity's performance or documentation. This level of independence could be viewed as strong certification, but not accreditation because the entity developer's organization is still in control of all of the information obtained in the assessment.

At the fourth level, the V&V evaluation is conducted by an entity evaluator who is contracted by the customer or potential customer of the entity. By *customer* we mean a user of the entity that is an independent organization from the developer's organization. This shift is a significant increase in the level of independent assessment and would normally be considered part of accreditation. Here, the authority to guarantee the performance, accuracy, or quality of the entity has moved from the entity developer's organization to a customer-oriented organization. This amount of insulation between the entity developer and the entity evaluator is appropriate for certain situations, such as those mentioned previously, but can also cause technical and practical problems. These problems are discussed as part of the next level of independence. The interpretation of accreditation commonly assumes that the assessment authority moves from the developer to the customer of the entity. If the developer and customer of the entity are essentially the same, then our assumption of independence does not apply. For example, in many DoD simulation activities the developer and the customer are essentially the same, or very closely related. As a result, this arrangement would not adhere to our interpretation of accreditation independence.

At the fifth level of independence, the V&V evaluation is conducted by an entity evaluator who is contracted by an *independent legal authority* or *governmental organization*. The evaluation authority has now moved not only further from the entity developer, but also moved from the entity customer, i.e., the user. The amount of insulation between the entity developer and the entity evaluator at the fifth level can be quite beneficial to the independent legal authority or governmental organization responsible for performance assessment of high-consequence systems. However, this insulation can have a detrimental effect on the quality of the computational analysis desired by the scientific and engineering community. Weakening the scientific quality of the entity is clearly not the intent of accreditation, but it can be a by-product. For example, any changes to the entity, even those intended to improve accuracy, efficiency, or robustness, cannot be made unless the entity is re-accredited. As a result, modifying an entity becomes a very time-consuming and expensive process. The degree to which the accreditation procedure can weaken the quality of computational analyses is illustrated, in our view, by the history of the safety assessment of nuclear power reactors in the United States. Currently, it is not clear how to achieve a better balance between the need for improving the quality of entities and the need for adequate assurance of public safety.

2.3 Types and sources of uncertainties

Computational simulation attempts to bring together what is known in terms of certainty and what is uncertain in the analysis of a system. Science and engineering has strongly tended to emphasize what we know, or think we know, instead of what is uncertain. There can be many different types of uncertainties that occur in computational analyses. A large number of researchers and practitioners in risk assessment (Morgan and Henrion, 1990; Kumamoto and Henley, 1996; Cullen and Frey, 1999; Suter, 2007; Vose, 2008; Haimes, 2009), engineering reliability (Melchers, 1999; Modarres *et al.*, 1999; Ayyub and Klir, 2006), information theory (Krause and Clark, 1993; Klir *et al.*, 1997; Cox, 1999), and philosophy of science (Smithson, 1989) have dealt with categorizing types of uncertainty. Many of the categorizations that have been constructed tend to mix the nature or essence of a type of uncertainty with how or where it might occur in computational analysis. For example, some taxonomies would have randomness as one type and model-form uncertainty as another type. A sound taxonomy would *only* categorize uncertainty types according to their fundamental essence, and then discuss how that essence could be embodied in different aspects of a simulation. For types of uncertainty that can be identified and characterized in some way, the computational analysis that incorporates these uncertainties will result in nondeterministic outcomes. By *nondeterministic* outcomes we mean those that explicitly acknowledge uncertainty in some way. Although these outcomes may be more difficult to interpret and deal with than deterministic outcomes, the goal of nondeterministic simulations is to improve the understanding of the processes in complex systems, as well as to improve the design and decision making related to these systems.

During the last 25 years, the risk assessment community, primarily the nuclear reactor safety community, has developed the most workable and effective categorization of uncertainties: aleatory and epistemic uncertainties. Some of the key developers of this categorization were Kaplan and Garrick (1981); Parry and Winter (1981); Bogen and Spear (1987); Parry (1988); Apostolakis (1990); Morgan and Henrion (1990); Hoffman and Hammonds (1994); Ferson and Ginzburg (1996); and Paté-Cornell (1996). See the following texts for a detailed discussion of aleatory and epistemic uncertainties: Casti (1990); Morgan and Henrion (1990); Cullen and Frey (1999); Ayyub and Klir (2006); Vose (2008); and Haimes (2009). The benefits of distinguishing between aleatory and epistemic uncertainty include improved interpretation of simulation results by analysts and decision makers, and improved strategies on how to decrease system response uncertainty when both are present. As will be discussed, the fundamental nature of each is different. As a result, different approaches are required to characterize and reduce each type of uncertainty.

2.3.1 Aleatory uncertainty

Consistent with the references just given, *aleatory uncertainty* is defined as:

Aleatory uncertainty: uncertainty due to inherent randomness.

Aleatory uncertainty is also referred to as stochastic uncertainty, variability, inherent uncertainty, uncertainty due to chance, and Type A uncertainty. The fundamental nature of aleatory uncertainty is randomness, e.g., from a stochastic process. Randomness can, in principle, be reduced, e.g., by improved control of a random process, but if it is removed, for example, by assumption, then you have fundamentally changed the nature of the analysis. Aleatory uncertainty can exist due to inter-individual differences, such as random heterogeneity in a population, and it can exist spatially or temporally. Sources of aleatory uncertainty can commonly be singled out from other contributors to uncertainty by their representation as randomly distributed quantities that may take on values in a known range, but for which the exact value will vary by chance from unit to unit, point to point in space, or time to time. The mathematical representation, or characterization, most commonly used for aleatory uncertainty is a probability distribution.

Aleatory uncertainty can be embodied in two ways in computational analyses: in the model form itself and in parameters of the model. If the model is given by a differential operator, then aleatory uncertainty in the model form can be expressed as a stochastic differential operator. Although there have been some applications of stochastic differential operators to actual engineering systems, this type of modeling is in its very early stages (Taylor and Karlin, 1998; Kloeden and Platen, 2000; Serrano, 2001; Oksendal, 2003). Aleatory uncertainty in parameters is, by far, a much more common situation in computational analyses. Aleatory uncertainty in parameters can occur in the mathematical description of the system and its characteristics, initial conditions, boundary conditions, or excitation function. Typically, aleatory uncertainty occurs in a scalar quantity appearing in the PDE, but it can also appear as a vector or a field quantity. Some examples of scalar parameters having random variability are: variability in geometric dimensions of manufactured parts; variability of the gross takeoff weight of a commercial airliner; and variability of the atmospheric temperature on a given day, at a given location on earth.

Consider a simple example of a scalar variability in a heat conduction analysis. Suppose one were interested in heat conduction through a homogenous material whose thermal conductivity varied from unit to unit due to a manufacturing process. Assume that a large number of samples have been drawn from the material population produced by the manufacturing process and the thermal conductivity has been measured on each of these samples. Figure 2.13 shows both the probability density function (PDF) and the cumulative distribution function (CDF) representing the thermal conductivity as a continuous random variable. The PDF and the CDF both represent the variability of the thermal conductivity of the population, but each show it in a different way. The variability of the population could also be shown as a histogram. The PDF (Figure 2.13a) shows the probability density of any chosen value of conductivity x. Stated differently, it shows the probability per unit variation in conductivity for any value x. The CDF (Figure 2.13b) shows the fraction of the population that would have a conductivity less than or equal to the particular value of conductivity chosen x. For example, the probability is 0.87 that all possible thermal conductivity values will be 0.7 or lower.

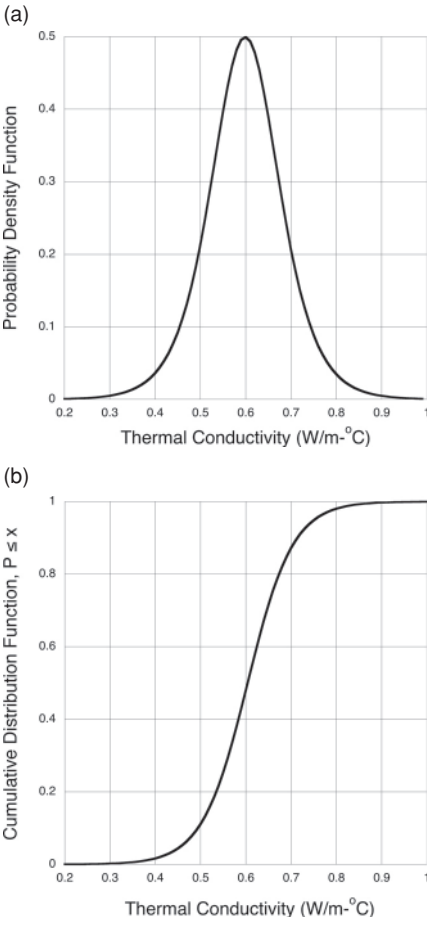

Figure 2.13 Examples of PDF and CDF for variability of thermal conductivity: (a) probability density function and (b) cumulative distribution function.

2.3.2 *Epistemic uncertainty*

Consistent with references cited above, *epistemic uncertainty* is defined as:

Epistemic uncertainty: uncertainty due to lack of knowledge.

Epistemic uncertainty is also referred to as reducible uncertainty, knowledge uncertainty, and subjective uncertainty. In the risk assessment community, it is common to refer to epistemic uncertainty simply as *uncertainty* and aleatory uncertainty as *variability*. The fundamental source of epistemic uncertainty is incomplete information or incomplete knowledge of any type that is related to the system of interest or its simulation. Epistemic uncertainty is a property of the modeler or observer, whereas aleatory uncertainty is a property of the system being modeled or observed. The lack of knowledge can be related to modeling issues for

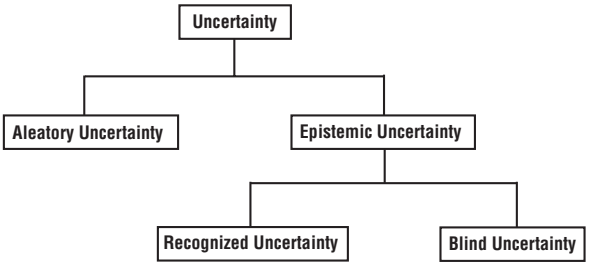

Figure 2.14 Classification of uncertainties.

the system, computational issues of the model, or experimental data needed for validation. Modeling issues include lack of knowledge of characteristics or processes in the system, the initial state of the system, or the surroundings or environment of the system. Computational issues include programming mistakes, estimation of numerical solution errors, and numerical approximations in algorithms. Experimental data issues include incomplete knowledge of experimental information that is needed for simulation of the experiment and approximations or corrections that are made in the processing of the experimental data. An increase in knowledge or information can lead to a reduction in epistemic uncertainty and thereby a reduction in the uncertainty of the response of the system, given that no other changes are made.

Taking a complementary perspective, it is seen that the fundamental characteristic of epistemic uncertainty is ignorance; specifically ignorance by an individual, or group of individuals, conducting a computational analysis. Smithson (1989) points out that ignorance is a social construction, analogous to the creation of knowledge. Ignorance can only be discussed by referring to the viewpoint of one individual (or group) with respect to another. Smithson gives the following working definition of ignorance:

> A is *ignorant* from B's viewpoint if A fails to agree with or show awareness of ideas which B defines as actually or potentially valid.

This definition avoids the absolutist problem by placing the onus on B to define what he/she means by ignorance. It also permits self-attributed ignorance, since A and B can be the same person.

Ayyub (2001), following Smithson, divides ignorance into two types: conscious ignorance and blind ignorance. Conscious ignorance is defined as a self-ignorance recognized through reflection. Conscious ignorance would include, for example, any assumptions or approximations made in modeling, the use of expert opinion, and numerical solution errors. For conscious ignorance, we will use the term *recognized uncertainty* to mean any epistemic uncertainty that has been recognized in some way. Blind ignorance is defined as ignorance of self-ignorance or unknown unknowns. For blind ignorance, we will use the term *blind uncertainty* to mean any epistemic uncertainty that has *not* been recognized in some way. Figure 2.14 shows the categorization of uncertainty that will be used. Recognized

uncertainty and blind uncertainty will now be discussed in detail, along with where and how they occur in computational analyses.

2.3.2.1 Recognized uncertainty

Although introduced above, we define *recognized uncertainty* more formally as:

Recognized uncertainty: an epistemic uncertainty for which a conscious decision has been made to either characterize or deal with it in some way, or to ignore it for practical reasons.

For example, in making decisions concerning the modeling of a system, one makes assumptions concerning what physics will be included in the model and what will be ignored. Whether a certain type of physical phenomenon is included or ignored, or a specific type of conceptual model is chosen, these are recognized uncertainties. Assumptions such as these are usually referred to as model form uncertainties, i.e., uncertainties due to the assumptions made in the modeling of the physics. Depending on the complexity of the physics involved, a modeler could, in concept, change the assumptions or the model and possibly estimate the magnitude of the effect on system response quantities of interest. Regardless of what level of physics modeling fidelity is chosen there are *always* spatial and temporal scales of physics, as well as coupled physics, that are ignored. A balance must be decided between what physics should be included in the modeling and the time and effort (both computational and experimental resources) needed to simulate the desired system responses. Whether the magnitude of the effect of an assumption or approximation is estimated or not, it is still a recognized uncertainty.

Another example of a recognized uncertainty is obtaining opinions from experts when experimental data is not available. For example, suppose an expert is asked to provide an opinion on his/her belief of a scalar parameter in the system that is a fixed quantity, but the value of the quantity is not known. The expert may provide an opinion in the form of a single number, but more likely the opinion would be given as an interval in which the true value is believed to be. Similarly, suppose an expert is asked to provide an opinion on a parameter that is characterized by a random variable. They would probably provide a named family of distributions for the characterization, along with estimated fixed values for the parameters of the family. Alternately, they could also provide interval values for the parameters of the family. In either case, the scalar parameter in the system would be a mixture of aleatory and epistemic uncertainty, because it represents expert opinion for a random variable.

Since the root cause of a recognized uncertainty is incomplete knowledge, increasing the knowledge base can reduce the epistemic uncertainty. Epistemic uncertainty can be reduced by an action that generates relevant information, such as allowing for a stronger level of physics coupling in a model, accounting for a newly recognized failure mode of a system, changing a calculation from single precision to double precision arithmetic, and performing an experiment to obtain knowledge of system parameters or boundary conditions imposed on the system. Epistemic uncertainty can also be reduced by eliminating the possibility of the existence of certain states, conditions, or values of a quantity. By reducing the collection

(or sample space) of possible events, one is reducing the magnitude of uncertainty due to ignorance. For example, suppose a system failure mode or dangerous system state has been identified such that it could occur if the system is incorrectly assembled. If the system is redesigned such that the system cannot be improperly assembled, then the epistemic uncertainty in the system response has been reduced.

The amount of information produced by an action could be measured by the resulting reduction in the uncertainty of either an input or output quantity. Treating uncertainty as an aspect of information theory or considering more general representations of uncertainty has led to the development of a number of new, or expanded, mathematical theories during the last three decades. Examples of the newer theories are (a) fuzzy set theory (Klir *et al.*, 1997; Cox, 1999; Dubois and Prade, 2000); (b) interval analysis (Moore, 1979; Kearfott and Kreinovich, 1996); (c) probability bounds analysis, which is closely related to second order probability, two-dimensional Monte Carlo sampling, and nested Monte Carlo sampling (Bogen and Spear, 1987; Helton, 1994; Hoffman and Hammonds, 1994; Ferson and Ginzburg, 1996; Helton, 1997; Cullen and Frey, 1999; Ferson and Hajagos, 2004; Suter, 2007; Vose, 2008); (d) evidence theory, also called Dempster–Shafer theory (Guan and Bell, 1991; Krause and Clark, 1993; Almond, 1995; Kohlas and Monney, 1995; Klir and Wierman, 1998; Fetz *et al.*, 2000; Helton *et al.*, 2005; Oberkampf and Helton, 2005; Bae *et al.*, 2006); (e) possibility theory (Dubois and Prade, 1988; de Cooman *et al.*, 1995); and (f) theory of upper and lower previsions (Walley, 1991; Kozine, 1999). Some of these theories only deal with epistemic uncertainty, but most deal with both epistemic and aleatory uncertainty. In addition, some deal with other varieties of uncertainty, e.g., nonclassical logics appropriate for artificial intelligence and vagueness due to language (Klir and Yuan, 1995).

2.3.2.2 Blind Uncertainty

Our formal definition of *blind uncertainty* is:

Blind uncertainty: an epistemic uncertainty for which it is not recognized that the knowledge is incomplete and that the knowledge is relevant to modeling the system of interest.

Adding knowledge can reduce blind uncertainty, just as with recognized uncertainty. However, the approach and the procedures are quite different because one is attempting to identify unknown unknowns. The most common causes of blind uncertainty are human errors, blunders, or mistakes in judgment. Some examples are: programming errors made in software used in the simulation, mistakes made in the preparation of input data or post-processing of output data, blunders made in recording or processing experimental data used for validation, and not recognizing how a system could be easily misused or damaged so that the system could be very dangerous to operate. Blind uncertainty can also be caused by inadequate communication between individuals contributing to the M&S, for example: (a) between those providing expert opinion and those interpreting and characterizing the information for input to the modeling, and (b) between computational analysts and experimentalists working on validation activities. In experimental activities, some additional examples of blind uncertainty are unrecognized bias errors in diagnostic techniques

or experimental facilities and improper procedures in using a reference standard in the calibration of experimental equipment.

There are *no* reliable methods for estimating or bounding the magnitude of blind uncertainties, their impact on a model, its simulation, or on the system's response. As a result, the primary approach for dealing with blind uncertainties is to try to identify them through such techniques as: (a) redundant procedures and protocols for operations or analyses, (b) various software and hardware testing procedures, (c) use of different experimental facilities, (d) use of a variety of expert opinions, and (e) use of broader sampling procedures to try to detect a blind uncertainty. Once blind uncertainties are identified or a hint of their existence is recognized, then they can be pursued or dealt with in some way or the impact of their effect could possibly be estimated or removed. For example, as discussed earlier in code verification, testing of numerical algorithms and SQA practices have proven effective in finding algorithm deficiencies and code bugs. Methods have been developed to estimate the frequency of coding errors, e.g., average number of static or dynamic faults per hundred lines of code. However, these measures do not address the possible *impact* of undetected coding errors. Human mistakes made in input preparation for simulations and mistakes in processing of output data are most commonly detected by having separate individuals check the data or by having completely separate teams conduct the same simulation, using the same modeling assumptions, and possibly even the same computer code, to detect any differences in results. For centuries, experimental science has been built on the crucial importance of independent reproducibility of experimental results and measurements. Scientific computing has a great deal to learn from this venerable tradition.

To stress the personal or social aspect of blind uncertainty, (Ayyub, 2001) gives several thought-provoking examples of root causes of blind uncertainty: knowledge that is dismissed as irrelevant (yet it is relevant); knowledge or experience that is ignored (yet it should not be ignored); and knowledge or questioning that is avoided or shunned because it is socially, culturally, or politically considered taboo. This personal aspect of blind uncertainty, and some of those mentioned earlier, can be countered, to some extent, by independent and/or external peer reviews of a computational effort. The effectiveness of an external review depends to a great extent on the independence, creativeness, expertise, and authority of the external reviewers. If an external review is focused on finding weaknesses, errors, or deficiencies, they are commonly referred to as *Red Team* reviews. Sometimes Red Teams have such high enthusiasm and zeal, one wonders if they are friends or enemies.

2.4 Error in a quantity

There are many situations in scientific computing and in experimental measurements where the concept of *error* proves to be quite useful. We will use the common dictionary definition of error.

Error in a quantity: a deviation from the true value of the quantity.

This definition is also used in a number of metrology texts (Grabe, 2005; Rabinovich, 2005; Drosg, 2007). To be more specific, let y_T be the true value of the quantity y, and let

y_{obtained} be the obtained value of the quantity y. *Obtained value* means that the result can be derived from any source, e.g., numerical solution, computational simulation, experimental measurement, or expert opinion. It is assumed that y_{T} and y_{obtained} are fixed numbers, as opposed to random quantities, i.e., realizations of a random variable. Then the error in y_{obtained} is defined as

$$\varepsilon_{\text{obtained}} = y_{\text{obtained}} - y_{\text{T}}. \tag{2.2}$$

Many texts and technical articles use the terms *error* and *uncertainty* interchangeably. We believe, however, that this produces a great deal of confusion and misinterpretation of the fundamental concepts. In addition, interchangeable use of the terms error and uncertainty can lead to a misrepresentation of results, causing misguided efforts directed at reduction or elimination of the source of the error or uncertainty.

As was discussed in Section 2.3, the concept of uncertainty fundamentally deals with whether the source is either stochastic in nature or its nature is lack of knowledge. The concept of error *does not* address the nature of the source, but concentrates on the identification of the true value. The true value can be defined in a number of different ways. For example, the true value of a physical constant, such as the gravitational constant or the speed of light in a vacuum, can be defined in multiple ways depending on the accuracy needed for the situation. The true value can also be defined as a reference standard, for example, the reference standards for length, mass, and time are set by the International System of Units. In scientific computing, it is sometimes convenient to define a true value as a floating-point number with specified precision in a computer. However, in most simulations the true value in not known or is not representable with finite precision, and in experimental measurements of engineering and scientific quantities the true value is *never* known.

As a result, the usefulness of the concept of error in practical applications depends on the definition and accuracy of the true value. If the accuracy of the true value is known, or the true value is given by an appropriate definition, and the accuracy of the true value is much higher than the y_{obtained} value, then the concept of error is quite useful, both conceptually and practically. For example, consider the case in code verification where an analytical solution to the PDEs can be computed with high accuracy, e.g., known to the double precision accuracy of the computer being used. One could define the error as the difference between the particular solution obtained by a code and the highly accurate analytical solution. If, however, the accuracy of the analytical solution is not computed very accurately, e.g., using an insufficient number of terms in an infinite series expansion, then the concepts and terminology associated with uncertainties are more useful in practice.

For example, consider the case of two computer codes solving the same physics models, but using different numerical solution procedures. Suppose that one of the codes has been traditionally accepted as producing the "correct" result and the other code is relatively new. One could define the error as the difference between the new code result and the traditional code result. Unless the traditional code result has been thoroughly investigated and the accuracy carefully documented over the range of input parameters, it would be foolish to consider it as producing the true value. A more appropriate

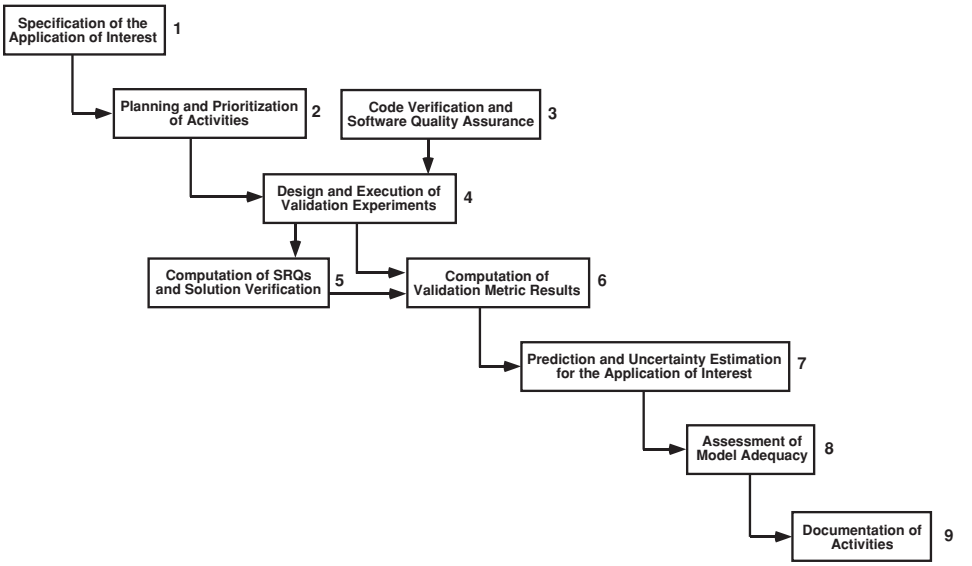

Figure 2.15 Integrated view of the elements of verification, validation, and prediction (adapted from Trucano *et al.*, 2002).

approach would be to characterize the accuracy of the result from each code as epistemically uncertain.

2.5 Integration of verification, validation, and prediction

As suggested earlier in this chapter, V&V are contributing elements in the development of a computational predictive capability. Researchers and code developers in M&S often stress the generality and capability of their models and codes. In some applications of M&S, however, the focus is on (a) quantitative assessment of confidence in the predictions made for a particular application of interest, and (b) how the predictions can be effectively used in a risk-informed decision-making process. By *risk-informed decision-making* we mean decision making that is guided and aided by both the estimated uncertainty in possible future outcomes, as well as the risk associated with those outcomes. Since uncertainty and risk can be very difficult to quantify, in addition to the issue of personal or organizational tolerance for risk, we use the term risk-informed decision-making. Many chapters will discuss uncertainty, and Chapter 3 will define and discuss risk in more detail.

Figure 2.15 depicts an integrated view of the nine elements of the verification, validation, and prediction process we discuss in detail in this book. This integrated view is similar to the ASME Guide diagram (Figure 2.5), but Figure 2.15 stresses the sequential nature of all of the elements of the M&S process. Each of these elements will be discussed in detail in various chapters of this book; however, a brief description of each element will be given here. Many of the concepts described in each element are based on Trucano *et al.* (2002).

Note that although the activities in Figure 2.15 are shown sequentially, the process is commonly an iterative one in practice. For example, when the computational results are compared with the experimental measurements in Element 6, it may be found that the computational results are not as good as expected. One may have several options for iterative adjustment of previously completed elements: (a) alteration of the modeling assumptions made in Element 1; (b) alteration of the application domain specified in Element 1; (c) reprioritization of certain V&V activities in Element 2 so as to better address the cause of the problem; (d) performance of additional code verification activities in Element 3 if the coding is suspect; (e) conducting additional experimental measurements in Element 4 if experimental measurements are suspect; and (f) computing solutions on finer mesh resolutions in Element 5 if the solution error is suspected of causing a problem.

2.5.1 Specification of the application of interest

The first element of the M&S process describes what physical process, engineering system, or event is of interest. One should define in some detail the specific purpose for which the M&S process is being undertaken. If different environments are of interest, such as accident and hostile environments, then it is appropriate to construct completely separate diagrams such as Figure 2.15; one for each environment. (System environments and scenarios will be discussed in detail in Chapter 3.) V&V can be accomplished without the focus on a specific application of interest, such as in software development and research into physical processes. Throughout this text, however, the discussion will usually focus on an application-driven V&V processes. In this first element, the customer for the computational results should be specified, along with how the customer intends to use the results, such as design, optimization, or policy making.

It may seem obvious that the specifications are given for the application of interest at the beginning of a M&S process. However, in both large and small-scale M&S activities there is commonly inadequate communication concerning the primary and secondary goals of the complete activity. The key participants in this discussion are the ultimate users of the computational information generated (referred to as customers of the M&S effort), the stakeholders in the activities, and the computational analysts conducting the work. Each group usually has similar ideas concerning the goals of the effort, but each group always brings a different perspective, priorities, and agenda to the activities. It is common that only after significant effort has been expended, or there are difficulties with the M&S activity, that these groups discover that each has surprisingly different goals for the activities. These types of miscommunication, or lack of communication, are particularly likely if the funding source of the M&S activity is *not* the user of the resulting capability, but some third party. For example, suppose the funding source is a governmental agency that is interested in developing a M&S capability in some application area under their regulatory control. Suppose the intended user of the capability is a contractor to the governmental agency, and the developer of the capability is a different contractor. This triad is especially susceptible to failure due to miscommunication.

As part of the first element, a description should be given for the application domain for the intended use of the model. This would include, for example, specification of the environments and scenarios that the system could be exposed to and which the M&S is suppose to address. This could also include general specification of all of the initial conditions, boundary conditions, and excitation conditions that the system might be exposed to. The anticipated validation domain needed for the application domain should also be described in general terms. At this early stage, the validation domain may only be vaguely anticipated because either the system is in the early design phase or the predictive capability of the model is poorly understood. However, unless the application domain is extremely different from the experience base on similar systems, some existing experimental data will be pertinent to the application domain. Note that an application domain and a validation domain can be specified at multiple tiers of the validation hierarchy discussed earlier in Figure 2.4.

An important part of the first element is the specification of all of the SRQs that are needed from the computational analysis. Some examples of SRQs that may be of interest are (a) temperature distribution inside or on the surface of a solid, (b) maximum stress level within a component or group of components, (c) maximum acceleration as a function of time at any point in or on a structural system, and (d) concentration level of a contaminant or toxic waste along a specified boundary. Closely associated with specification of the SRQs of interest is the specification of the predictive accuracy requirements that are needed from the M&S effort. It is the customer of the effort who should define the predictive accuracy requirements. Sometimes, however, the customer either (a) fails to do this, or (b) is overly demanding of the accuracy requirements. This results in a constructive dialogue and negotiation between the customer and the analysts concerning the cost and schedule required for certain levels of predictive accuracy. Although estimates for costs and schedule, as well as estimated predictive accuracy, are often very poorly known early in an effort, these discussions are critical early on so that all parties, including stakeholders and experimentalists that may need to provide additional validation data, have some feel for the trade-offs involved. Too often, understanding of trade-offs between cost, schedule, and achieved predictive accuracy occur very late in the effort after significant resources, time, and modeling effort have been expended or wasted.

Many of the activities in this element are discussed in Chapters 3, 10, and 14.

2.5.2 Planning and prioritization of activities

Formal planning and prioritization of M&S, V&V, and prediction activities are conducted in the second element. The planning and prioritization should attempt to address all of the activities that are conducted in the remaining seven elements shown in Figure 2.15, given the specifications made in Element 1. On a large M&S project, this requires significant resources and effort from a wide variety of individuals and, sometimes, a variety of organizations. On a large project, the effort should also include documentation of the planning and prioritization in a V&V Plan. Preparation of a V&V Plan is also discussed and recommended in the ASME

Guide (ASME, 2006). The Plan should be appropriate to the magnitude of the project and to the consequences of the decisions made based on the computational results.

The focus of the planning and prioritization effort should always be:

Given the resources available (people, time, money, computational facilities, experimental facilities, etc), what is the appropriate level of effort in each activity needed to achieve the goals of the M&S effort identified in Element 1?

Some examples of the types of question addressed in the planning and prioritization element are the following.

- What physical phenomena are important and what level of coupling of the various phenomena are appropriate for the goals of the analysis?
- What are the anticipated application domains and validation domains?
- What are the SRQs of interest and what are the prediction accuracy requirements expected by the customer?
- What code verification and SQA activities are appropriate for the application of interest?
- Are existing numerical error estimation techniques adequate?
- Are new mesh generation capabilities needed for the analysis?
- Are new experimental diagnostics or facilities needed for validation activities?
- Do new validation metric operators need to be developed?
- Are adequate methods available for propagating input uncertainties through the model to obtain output uncertainties?
- If model accuracy or experimental measurements are found to be lacking, what alternatives or contingency plans should be considered?

In our experience, the most commonly used approach for planning and prioritization is the Phenomena Identification and Ranking Table (PIRT) (Boyack *et al.*, 1990; Wilson *et al.*, 1990; Wulff *et al.*, 1990; Wilson and Boyack, 1998; Zuber *et al.*, 1998). PIRT was originally developed to identify physical phenomena, and the coupling of physical phenomena, that could affect nuclear power plant safety in accident scenarios. PIRT should be viewed as a process as well as a collection of information. As stressed by Boyack *et al.* (1990), the PIRT is most certainly not set in stone once it is formulated and documented. While a given formulation of a PIRT guides M&S activities, it must also adapt to reflect the information gathered during the conduct of those activities.

An additional planning and prioritization process was developed by Pilch *et al.*, (2001); Tieszen *et al.* (2002); Trucano *et al.* (2002); and Boughton *et al.* (2003). This process, referred to as a gap analysis, begins with the results of the PIRT process and attempts to answer the question:

Where does the M&S effort presently stand relative to the phenomena and SRQs that have been identified as important?

In the gap analysis portion of the process, the emphasis shifts from improving the understanding of the environments, scenarios, system, and physical phenomena, to an understanding the possible gap between the *present* capabilities and required capabilities of M&S tools.

Answers to this question with regard to modeling, computer codes, verification, validation, and uncertainty quantification can directly aid in planning and prioritization.

The PIRT and gap analysis processes are discussed in detail in Chapter 14, Planning and prioritization in modeling and simulation.

2.5.3 Code verification and software quality assurance activities

Code verification and software quality assurance (SQA) activities are conducted in the third element. Both of these activities can be viewed as the accumulation of evidence to support the belief that: (a) the numerical algorithms are functioning as intended, (b) the source code is implemented correctly and it is functioning as intended, and (c) the computer system hardware and software environment is functioning as intended. It is well known, although rarely stated, that these three essentials (source code, hardware, and system software) *cannot* be proven to be correct and functioning as intended. In fact, experience on any computer system shows that application source codes have programming errors, and hardware and system software have limitations and flaws (sometimes known, i.e., recognized uncertainties, and sometimes unknown, i.e., blind uncertainties). As a result, computer users tend to develop the mind set that the potential for software and hardware errors is ignored up to some level. Individual tolerance levels primarily depend on two very different factors. First, how averse is the individual to the severity and frequency of errors and unreliability in the software and hardware? Second, what are the individual's options for using other software and hardware to accomplish their job? For example, if a user has a low tolerance for software bugs, *and* they have the option to change software, then they may be motivated to make a change. On the other hand, a user: (a) may tolerate buggy, unreliable, software if they have no options, e.g., if there is near monopoly in the software market, or (b) may be forced to use certain system or application software because of corporate or organizational mandates (Platt, 2007).

Given the perspective of the balance between individual tolerance of errors and unreliability on the one hand, and the availability of software options on the other, it is our observation that computer users of computational software show a high tolerance for errors and lack of robustness, as long as the features and capabilities they need for their simulations are perceived to be met. Stated differently, computational software users place little value on the accumulation of evidence for code verification and SQA, as opposed to the value they place on the software having the features and capabilities they need to do their job. This commonly held value-system is, we believe, at the root of why Element 3 in the M&S process receives considerable lip service, but minimal effort when it competes for code development resources. Code development groups, whether they are in-house groups or commercial software companies, understand this value system and they respond accordingly.

The integrated view of V&V and prediction shown in Figure 2.15 does not solve the problem of the competition of resources between code verification and SQA activities versus implementation of features and capabilities needed to complete the M&S goals at hand.

However, Figure 2.15 does call attention to the critical foundation that code verification and SQA activities play in assessing the credibility of a computational result. For example, if a code bug is found later on in the M&S process, say in Element 7, all of the effort devoted to Elements 5 through 7 must be rechecked to see if the bug affected the previous work. If the bug did have an effect on the earlier results, then much of the work is wasted and most has to be redone. The far more dangerous situation is if a relevant code bug was *not found* and trust was placed in the computational results. For example, if good agreement was found between computational and experimental results in Element 6, little interest, energy, and resources may be found to conduct code verification and SQA activities. By using misleading computational results, decision makers may then have unknowingly made erroneous decisions on system safety, performance, or reliability.

Code verification and SQA will be discussed in detail in Chapters 4–6.

2.5.4 Design and execution of validation experiments

Element 4 deals with the design and execution of validation experiments, as well as the more common situation of using existing experimental data in validation activities. Before briefly discussing the design and execution of validation experiments, a few comments should be made concerning how a validation experiment is different from traditional types of experiments. Traditional experiments can generally be grouped into three broad categories (Oberkampf *et al.*, 1995; Aeschliman and Oberkampf, 1998; Oberkampf and Blottner, 1998; Oberkampf and Trucano, 2002). The first category comprises experiments that are conducted primarily for the purpose of improving the fundamental understanding of some physical process. Sometimes these are referred to as physical-discovery or phenomena-discovery experiments. Examples are experiments that investigate (a) turbulent reacting flow, (b) decomposition of materials as they decompose, (c) micromechanics processes underlying crack growth in solids, and (d) properties of materials undergoing phase change at extreme pressure and temperature.

The second category of traditional experiments consists of those conducted primarily for constructing or improving mathematical models of fairly well-understood physical processes. Sometimes these are called model development or model calibration experiments. For these types of experiment, the range of applicability of the model or the level of detail of the physics in the model is not usually important. Examples are experiments to (a) measure the reaction-rate parameters in a model for reacting flows, (b) determine the joint-attachment damping and the aerodynamic damping parameters in the vibration of a built-up structure, (c) determine the parameters in a model for crack propagation in a certain class of composite materials, and (d) calibrate the constitutive parameters in a material model for reinforced concrete.

The third category of traditional experiments includes those that determine the reliability, performance, or safety of components or subsystems, as well as complete engineering systems. Sometimes these are called reliability tests, performance tests, safety tests, certification tests, or qualification tests. Examples are (a) tests of a new compressor or

combustor design in a gas turbine engine, (b) tests of a new propellant formulation for a solid rocket motor, (c) tests of the crash worthiness of a new automobile design, and (d) qualification tests of a modified submarine design submerging to maximum operational depth.

A validation experiment, on the other hand, is conducted for the primary purpose of assessing the accuracy of a mathematical model. In other words, a validation experiment is designed, executed, and analyzed for the purpose of quantitatively determining the ability of a mathematical model expressed in computer software to simulate a well-characterized physical process. Thus, in a validation experiment one could state that *the computational analyst is the customer* or *the code is the customer* for the experiment as opposed to, for example, a physical phenomena researcher, a model builder, or a system project manager. Only during the last few decades has M&S matured to the point where it could even be considered as a viable customer. As modern technology increasingly moves toward engineering systems that are designed, certified, and possibly even fielded, based on M&S, then M&S itself will increasingly become the customer of validation experiments.

Since a validation experiment, as defined here, is a relatively new concept, most experimental data generated in Element 4 will be from different types of traditional experiment. Use of experimental data from traditional experiments when used for validation assessment must, as a result, deal with a number of technical and practical difficulties. These difficulties are discussed in Chapter 10, Model validation fundamentals. Here a brief discussion will be given of some of the important aspects of the design and execution of validation experiments. A more detailed discussion will be given in Chapter 11, Design and execution of validation experiments.

Validation experiments in the present context should be designed specifically for the purpose of evaluating the computational predictive capability that is directed toward the application of interest identified in Element 1. Validation experiments can, of course, be designed and executed without a specific application of interest in mind. However, our focus here is on validation experiments directed toward a specific application driver. The planning and prioritization of validation experiments should be a product of Element 2, not only for experimental activities, but also across the entire M&S project. For example, referring back to Figure 2.10c, the issue should be raised in Element 2 concerning resources required for conducting a new validation experiment within the application domain, versus resources expended on additional modeling activities. The approach to these trade-off studies should be: for a given quantity of resources expended, which option most reduces the estimated uncertainty in the predicted SRQs of interest? Even though it is constructive to frame the question as an optimization problem, it must be realized that it is still a very difficult question to answer. Some of these reasons are (a) the resources needed to achieve a certain goal or capability are poorly known; (b) it is only vaguely known what is needed to decrease the uncertainty in an input parameter in order to achieve a decrease in uncertainty in SRQs of interest; (c) the number of parameters in the trade-off space is extremely high; and (d) there are commonly unknown dependencies between some of the parameters in

trade-off space, i.e., all of the coordinates in the space are not orthogonal. These issues will be discussed in Chapter 14.

The primary guidelines for the design and execution of validation experiments have been formulated by Aeschliman and Oberkampf (1998); Oberkampf and Blottner (1998); and Oberkampf and Trucano (2002). These six guidelines are:

1 A validation experiment should be jointly designed by experimentalists, model developers, code developers, and code users working closely together throughout the program, from inception to documentation, with complete candor about the strengths and weaknesses of each approach.
2 A validation experiment should be designed to capture the essential physics of interest, and measure all relevant physical modeling data, initial and boundary conditions, and system excitation information required by the model.
3 A validation experiment should strive to emphasize the inherent synergism attainable between computational and experimental approaches.
4 Although the experimental design should be developed cooperatively, independence must be maintained in obtaining the computational and experimental system response results.
5 Experimental measurements should be made of a hierarchy of system response quantities; for example, from globally integrated quantities to local quantities.
6 The experimental design should be constructed to analyze and estimate the components of random (precision) and systematic (bias) experimental uncertainties.

These guidelines will be discussed in detail in Chapter 11, along with a high quality validation experiment example that demonstrates each guideline.

2.5.5 Computation of the system response quantities and solution verification

Element 5 deals with obtaining simulations for the validation experiments conducted, as well as assessing the numerical accuracy of those solutions. In Figure 2.15, the arrow drawn from Element 4 to Element 5 indicates that information from the validation experiment must be provided to the analyst to compute the SRQs that were measured in the validation experiment. Examples of the information needed for the simulation are the boundary conditions, initial conditions, geometric details, material properties, and system excitation. The information provided by the experimentalist should be accompanied by uncertainty estimates for each quantity provided. Here we stress estimates of uncertainties for both the SRQs, as well as the input quantities needed for the simulation. This is one of the important characteristics of high quality validation experiments. The uncertainty estimates provided could be characterized in several different ways, e.g., either probability density functions (PDFs) or equivalently cumulative distribution functions (CDFs), or simply interval-valued quantities with no likelihood information provided. However the uncertainty is characterized, this same characterization should be used when these uncertainties are propagated through the model to obtain SRQs with similarly characterized uncertainty.

As pointed out earlier in this chapter, and by several authors in the literature, the SRQs measured in the experiment should *not* be provided to the computational analysts before the simulations are completed. The optimum situation is for the analysts to make a blind

prediction of the validation experiment results, provided only with the input quantities needed for their simulation. For well-known experiments in the validation database, however, this is not possible. There are varying opinions on how damaging it is to the value or credibility of the comparisons of computational and experimental results if the analysts know the measured SRQs. We are of the belief that it is *very* damaging to the usefulness of validation and the credibility of predictive capability. Stated differently, it has been our experience, and the experience of many others, that when the analysts know the measured responses they are influenced in many ways, some obvious and some not so obvious. Some examples of influence are (a) modification in modeling assumptions, (b) choice of numerical algorithm parameters, (c) mesh or temporal convergence resolution, and (d) adjustment of free parameters in the model or poorly known physical parameters from the experiment.

Solution verification activities are conducted on the solutions that are used to compare results with experimental data. Two very different types of verification are conducted: verification of the input and output processing and verification of the numerical solution accuracy. Most of the formal solution verification effort is typically directed toward estimating numerical convergence errors (space, time, and iterative). *A posteriori* methods are, generally, the most accurate and effective approach for estimating numerical solution error in the SRQs of interest. If the SRQs of interest are field quantities, such as local pressure and temperature over the domain of the PDE, then numerical error must be estimated directly in terms of these quantities. It is well known that error estimation in local or field quantities is much more demanding in term of discretization and iterative convergence than error estimation of norms of quantities over the entire domain of the PDE.

If a relatively large number of validation experiments are to be simulated, then numerical solution error estimates are usually computed for representative conditions of various classes or groups of similar conditions. This procedure, if it can be properly justified, can greatly reduce the computational effort needed compared to estimating the numerical solution error for each experiment simulated. For example, a solution class could be defined for conditions that have similar geometries, similar nondimensional parameters occurring in the PDEs, similar interactions of physical processes, and similar material properties. After a solution class is defined, then one should choose either a representative condition from the entire class or, if it can be physically justified, the most computationally demanding condition from the class. For example, the most demanding in terms of mesh resolution and iterative convergence may be one that has the highest gradient solutions, the highest sensitivity to certain physical characteristics occurring in the field, or the highest interaction of coupled physics.

Computation of SRQs will be discussed in detail in Chapters 3 and 13, and solution verification will be discussed in Chapters 7 and 8.

2.5.6 Computation of validation metric results

Element 6 of Figure 2.15 deals with the quantitative comparison of computational and experimental results by using validation metric operators. It is common practice in all fields

of engineering and science to compare computational results and experimental data using graphs. Graphical comparisons are usually made by plotting a computational SRQ along with the experimentally measured SRQ over a range of some parameter. Common practice has been that if the computational results generally agree with the experimental data over the range of measurements, the model is commonly declared "validated." Comparing computational results and experimental data on a graph, however, is only incrementally better than making a subjective comparison. In a graphical comparison, one rarely sees quantification of numerical solution error or quantification of uncertainties in the experiment or the simulation. Uncertainties arise from experimental measurement uncertainty, uncertainty due to variability in experimental conditions, initial conditions or boundary conditions not reported by the experimentalist, or poorly known boundary conditions in the experiment. The experimental condition uncertainties, or those that are unreported from the experiment, are commonly considered as free parameters in the computational analysis and, as a result, they are adjusted to obtain better agreement with the experimental measurements.

The topic of validation metrics has received a great deal of attention during the last decade, primarily by researchers associated with Sandia National Laboratories. For some of the early work in this field, see Coleman and Stern (1997); Hills and Trucano (1999); Oberkampf and Trucano (2000); Dowding (2001); Easterling (2001a), (2001b); Hills and Trucano (2001); Paez and Urbina (2001); Stern *et al.* (2001); Trucano *et al.* (2001); Urbina and Paez (2001); Hills and Trucano (2002); and Oberkampf and Trucano (2002). A validation metric operator can be viewed as a difference operator between computational and experimental results for the same SRQ. The validation metric operator could also be referred to as a mismatch operator. The output from the difference operator is called the *validation metric result* and it is a measure of the model-form bias error for the specific conditions of the validation experiment. The validation metric result is a *quantitative* statement of the difference between the model predictions and the experimental measurements. The validation metric result is of significant practical value because it is an objective measure, as opposed to subjective personal opinions as to "good" or "bad" agreement. If the validation domain encompasses the application domain, as shown in Figure 2.10a, then the validation metric result can be directly compared with the model accuracy requirements specified in Element 1. In addition, the aggregation of validation metric results over the entire validation domain can be used to form the basis for characterization of model-form uncertainty for extrapolations outside of the validation domain. Stated differently, validation metric results are based on *observed* performance of the model that can be used to estimate model-form uncertainty for extrapolations to other conditions of interest.

The construction of validation metric operators is relatively new and there are different opinions as to what they should include, and exclude, and how they should be constructed. The following recommendations give one perspective on a constructive approach to formulation of validation metrics (Oberkampf and Trucano, 2002; Oberkampf and Barone, 2004, 2006; Oberkampf and Ferson, 2007).

1 A metric should either: (a) explicitly include an estimate of the numerical error in the SRQ of interest resulting from the computational simulation or (b) exclude the numerical error in the SRQ of interest, but only if the numerical error was previously estimated, by some reasonable means, to be small.
2 A metric should be a quantitative evaluation of predictive accuracy of the SRQ of interest, including all of the combined modeling assumptions, physics approximations, and previously obtained physical parameters embodied in the model.
3 A metric should include, either implicitly or explicitly, an estimate of the error resulting from post-processing of the experimental data to obtain the same SRQ that results from the model.
4 A metric should incorporate, or include in some explicit way, an estimate of the measurement errors in the experimental data for the SRQ that is compared with the model.
5 A metric should generalize, in a mathematically rigorous way, the concept of a difference between scalar quantities that have no uncertainty and quantities that can have both aleatory and epistemic uncertainty.
6 A metric should exclude any indications, either explicit or implicit, of the level of adequacy in agreement, or satisfaction of accuracy requirements, between computational and experimental results.
7 A validation metric should be a true metric in the mathematical sense, i.e., retaining essential features of a true distance measure.

A detailed discussion of the construction and use of validation metrics will be given in Chapter 12.

2.5.7 Prediction and uncertainty estimation for the application of interest

In the analysis of the performance, safety, and reliability of many systems, predictions are viewed strictly as deterministic, i.e., uncertainties in the modeling of the physical process or system are considered small or they are simply ignored. To account for any uncertainties that exist, a safety factor is then added to the various design features of the system (Elishakoff, 2004). A second approach that is sometimes used is to try and identify the worst condition, or most demanding operational condition, under which the system might be required to operate. The system is then designed to operate successfully and safely under those conditions. Depending on the needs of the analysis and the systems involved, either approach can be appropriate and cost effective.

During the last three decades, the fields of nuclear reactor safety (Morgan and Henrion, 1990; NRC, 1990; Modarres, 1993; Kafka, 1994; Kumamoto and Henley, 1996) and underground storage of toxic and radioactive materials (LeGore, 1990; Helton, 1993; 1999; Stockman *et al.*, 2000) have pioneered modern approaches to risk assessment. The performance and risk analysis of high-consequence systems such as these required the development of new and more credible nondeterministic methods. The mathematical model of the system, which includes the influence of the surroundings on the system, is considered nondeterministic in the sense that: (a) the model can produce nonunique system responses because of the existence of uncertainty in the input data for the model, (b) the analysis may consider multiple possible environments and scenarios that the system may experience, and

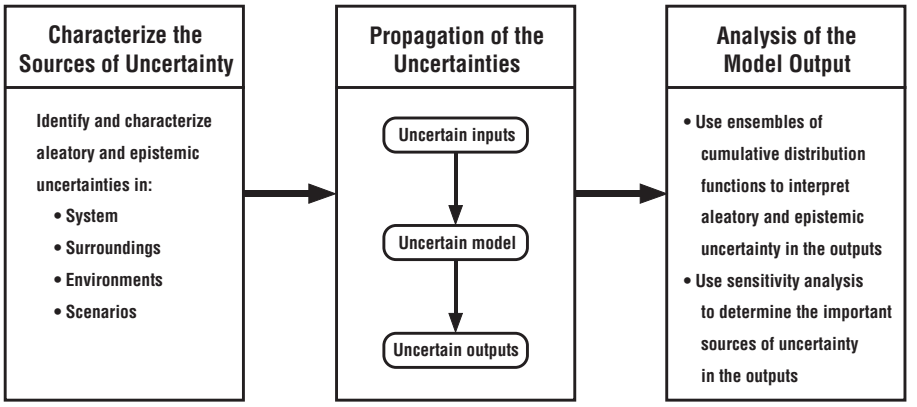

Figure 2.16 Basic steps in an uncertainty analysis.

(c) there may be multiple alternative mathematical models for the same system of interest. The term *nondeterministic* is used instead of *stochastic* because the nondeterminism can be due to either aleatory or epistemic uncertainty or, more commonly, a combination of both. The mathematical models, however, are assumed to be deterministic in the sense that when all necessary input data for a designated model are specified, the model produces only one value for every output quantity. That is, there is a one-to-one correspondence from input to output of the model. To predict the nondeterministic response of the system, it is necessary to evaluate the mathematical model, or alternative mathematical models, of the system multiple times using different input data and under possibly different environments and scenarios.

Element 7 deals with the nondeterministic prediction of the SRQs for the application of interest by incorporating into the mathematical model any uncertainties that have been identified. The most common strategy for incorporating uncertainty directly into the computational analysis involves three basic steps (Figure 2.16). The first step is called *characterizing the sources of uncertainty*. The uncertainties can be characterized as either an aleatory uncertainty or as a recognized epistemic uncertainty (Figure 2.13). If an uncertainty is purely aleatory, then it is characterized as a PDF or a CDF. If it is characterized as purely epistemic, then it is characterized as an interval-valued quantity with no likelihood information specified. An uncertainty can also be characterized as mixture of aleatory and epistemic uncertainty, i.e., some portions of the characterization are probability distributions and some are given as intervals.

In the second step, called *propagation of the uncertainty*, values from the uncertain input quantities specified in the previous step are propagated through the model to obtain uncertain output quantities. There are a number of propagation methods available to compute the mapping of input to output quantities. (For a detailed discussion of propagation methods, see the following texts: Morgan and Henrion, 1990; Cullen and Frey, 1999; Melchers, 1999; Haldar and Mahadevan, 2000; Ang and Tang, 2007; Choi *et al.*, 2007; Suter, 2007;

Rubinstein and Kroese, 2008; Vose, 2008.) In this text, we will concentrate on using statistical sampling procedures, such as Monte Carlo or Latin Hypercube Sampling, for two primary reasons. First, sampling methods are conceptually straightforward to understand and apply in practice. They can be used as a pre- and post-processor to any type of mathematical model, in the sense that they can be used as an *outer loop* or *wrapper* to the simulation code. Second, sampling methods can easily accommodate aleatory and epistemic uncertainties. Samples are taken from both the aleatory and epistemic uncertainties, but each type of uncertainty is treated separately and kept segregated in the analysis and in the interpretation of the results (Helton, 1994; Hoffman and Hammonds, 1994; Ferson and Ginzburg, 1996; Cullen and Frey, 1999; Ferson *et al.*, 2004; Suter, 2007; Vose, 2008). Samples from aleatory uncertainties represent stochastic uncertainty or variability and, as a result, these samples represent aleatory uncertainty in the SRQs. Samples from the epistemic uncertainties represent lack of knowledge uncertainty and, therefore, these samples represent *possible* realizations in the SRQs. That is, no probability or likelihood is associated with any samples taken from epistemically uncertain input quantities. Note that if alternative mathematical models are used to estimate the model form uncertainty, then the results from each model are also considered as epistemic uncertainties. The complete set of all samples for the SRQs is sometimes called an *ensemble of calculations*. Where once one might have performed a single calculation for a deterministic result, now one must perform a potentially large number of calculations for a nondeterministic simulation.

After the set of calculations has been generated, the third step, *analysis of the model output*, is performed. This step involves interpretation of the ensemble of calculations produced by sampling for the SRQs of interest. The general form of the ensemble of calculations is a family of CDFs or, equivalently, a family of complementary cumulative distribution functions. Multiple probability distributions are produced because of the existence of epistemic uncertainty. Each probability distribution represents the results from sampling all of the aleatory uncertainties, from *one* sample from all of the epistemically uncertainty quantities. Each one of the probability distributions represents a *possible* probability distribution of the SRQs.

The analysis of the output should also include a sensitivity analysis of the results (Cacuci, 2003; Saltelli *et al.*, 2004, 2008). Sensitivity analysis is the study of how the variation in the model outputs can be apportioned to different sources of variation in the model inputs (Saltelli *et al.*, 2008). Sensitivity analyses are commonly grouped into local and global analyses. Local sensitivity analyses deal with the question: how do uncertain outputs change as a function of uncertainty inputs? Global sensitivity analyses deal with the broader question: how does the uncertainty structure of the inputs, including multiple models, map to the uncertainty structure of the outputs? Answering these types of question can be extremely important from a design optimization, project management, or decision-making perspective because one can begin to focus on the causes of large uncertainties in system performance, safety, and reliability.

Each of the activities discussed in this element will be discussed in detail in Chapter 13.

2.5.8 Assessment of model adequacy

The assessment of model adequacy conducted in Element 8 primarily deals with assessment of estimated model accuracy as compared to required model accuracy specified in Element 1. As mentioned earlier, many other practical and programmatic issues enter into the decision of model adequacy for an intended application. In validation activities, we are only concerned with the estimate of model accuracy relative to the required accuracy. If the model accuracy requirements are not specified, the underlying philosophy of validation is put at risk. Accuracy requirements should be given over the entire application domain for all of the SRQs of interest. Since there is commonly a broad range in SRQs of interest in an analysis, and their importance to the system performance varies widely, the accuracy requirements can vary from one SRQ to another. In addition, the accuracy requirements for a given SRQ typically vary considerably over the application domain. For example, in regions of the application domain that are unimportant from a system performance or risk perspective, the accuracy requirements may be relatively low.

If there is sufficient validation data, the estimate of model accuracy can be built directly on the validation metric results obtained in Element 6. As mentioned earlier, the validation metric result is a direct measure of the model-form bias error over the validation domain. A validation metric result can be computed using a multi-dimensional interpolation procedure to compute the difference (mismatch) between the computational results and experimental measurements. If the application domain is completely enclosed by the validation domain (Figure 2.10a), the interpolated mismatch can then be compared with the model accuracy requirements to determine the adequacy of the model. If the model accuracy is adequate, then the model, along with the mismatch represented as an epistemic uncertainty, can be used in predictions for the system of interest. If the model accuracy is inadequate, then improvements to the model can be made in two ways. First, adjustable parameters in the model can be calibrated to obtain better agreement with the experimental data. Second, assumptions made in the conceptual model can be updated so as to improve the model form. In this latter case, however, one may need to repeat many of the elements in the entire process shown in Figure 2.15. Alternatively, the computational approach may be abandoned, such as the case shown in Figure 2.10a, and the system performance estimated using only the available experimental data and the judgment of the decision makers.

If any portion of the application domain is outside of the validation domain (Figure 2.10b), then the validation metric results must be extrapolated to the conditions of the application of interest. If the application domain is far from the validation domain (Figure 2.10c), then the extrapolation procedure can introduce large uncertainty in the estimated model-form bias error. In an attempt to try and address this issue, various extrapolation procedures could be used to estimate the uncertainty due to the extrapolation. The results of each of these extrapolation procedures could be compared to the model accuracy requirements to determine the adequacy of the model. Note that this extrapolation is completely separate from the model prediction that relies on the physics-based assumptions in the model and

the conditions at the application of interest. With model-form uncertainty we are dealing with an extrapolation of the estimated error in the model.

Conceptually, the model adequacy assessment approach outlined is logically well founded and, most importantly, it directly ties model accuracy assessment to the application-specific requirements. However, there are severe technical and practical difficulties that can arise in using the procedure when large extrapolation of the validation metric result is required (Figure 2.10c). Here we mention one, but a more complete discussion is given in Chapters 12 and 13. Suppose one is dealing with a situation where there is a system or environmental parameter that cannot be considered as a continuous variable. For example, suppose that experimental facilities can only produce relevant physical conditions on components or subsystems, but the complete system *cannot* be tested. As a result, all of the validation domain data exists at lower tiers in the validation hierarchy (Figure 2.4). Then one is dealing with the vague concept of increasing system complexity and its impact on the credibility of the model predictions. One could simply ignore the additional uncertainty in the model that is due to coupling of the models from the tested level of the hierarchy to the untested level of the hierarchy. This is certainly not appealing. Another approach is to use alternative plausible models at the tested and untested levels of the hierarchy so as to obtain multiple model predictions at the untested level. Considering the difference between each model prediction as an epistemic uncertainty, one could begin to estimate the uncertainty due to coupling of the models. This approach will not necessarily bound the prediction uncertainty, but it will give the decision maker a rough indication of the magnitude of the uncertainty.

2.5.9 Documentation of M&S activities

Although the topic of documentation in M&S usually generates little enthusiasm (at best) among analysts, some level of documentation is always needed. The magnitude of the documentation effort usually depends on the size and goals of the M&S project, as well as the consequences of the risk-informed decision-making that is based on the simulation results. For example, at the minimal end of the documentation spectrum would be quick response in-house studies or informal questions asked by a design group. In the middle part of the spectrum would be documentation requirements for a corporate product that had either performance guaranties associated with the product, or some aspect of legal liability. At the extreme end of the spectrum would be documentation requirements for high-consequence systems that could affect large portions of the population or the environment if a failure occurred. Our comments here are directed at the middle to high end of the documentation spectrum.

The goals of documentation are usually discussed in terms of the need for reproducibility, traceability, and transparency of the M&S activity. By *transparency* we mean that all aspects of the M&S activity can be examined and probed by technically qualified individuals. When proprietary models or software are used, however, transparency suffers greatly. These documentation goals are important in any simulation effort, particularly those that

support certification or regulatory approval of the safety and reliability of high-consequence systems. Some examples are the performance and risk assessment for nuclear reactor safety, large scale public structures such skyscrapers and dams, and long-term underground storage of nuclear or toxic wastes, such as the Waste Isolation Pilot Plant (WIPP) and the Yucca Mountain Project. The US DoD has stressed documentation in all of its V&V and accreditation activities and their recommendations for the structure and information content are given in DoD (2008).

Since the documentation goals of reproducibility, traceability, and transparency seem rather aloof and uninteresting to most personnel involved in M&S activities, we give the following examples that may be more motivational to various individual perspectives. These examples are listed in the order of the elements shown in Figure 2.15.

- Clear documentation of the application of interest (including system environments and scenarios), assumptions made in the modeling, and the prediction requirements expected of the M&S capability.
- Documented justification for the planning and prioritization of the M&S and V&V activities, not only at the beginning of the project, but also any changes that are needed during the project.
- Documentation of code verification and SQA activities that have (and have not) been conducted, as well as the ability to reproduce the activities during the project.
- Documentation of the design and execution of validation experiments not only for use in the present project, but also for use in future projects.
- Detailed documentation of simulations computed and numerical methods used so that the results can be explained and justified to the customer of the effort, delivered as part of a contractual agreement, or reproduced for training new staff members, an investigation board, or a regulatory or legal authority.
- Documentation of the validation domain, particularly its relationship to the application domain, as well as model accuracy assessment for various SRQs over the validation domain.
- Documentation of when and how model calibration was conducted so that changing model predictions can be traceable to specific calibration activities.
- Documentation of the predicted SRQs and their uncertainties for the conditions of the application of interest.
- Documentation of model adequacy (and any identified inadequacies) relative to the prediction accuracy requirements, for the application of interest.

Whatever the level or type of documentation generated, an electronic records management system (RMS) should be used. Some commercial RMS software is available, but on large-scale projects it is common to construct a tailor-made system. The RMS could be organized in a tree or folder/subfolder structure so that at the base level would be the particular M&S project, then it would divide into the eight elements shown in Figure 2.15, and then any further appropriate subdivisions. The RMS should be searchable by key words within any portion of any information element. The search engine could operate much like that found in Google or Wikipedia. Functionality could be expanded to include a relevancy-ranking feature that would further improve the search-and-retrieval capability. The overall system design could include searchable elements such as design configuration, computer code,

experimental facility, system safety features, and personnel involved. After the results were retrieved, they could be sorted according to their relevance to the words input to the search. The high-level results retrieved should be embedded with hyperlinks. One could then select the hyperlinks to pursue more detailed information of interest, including photographic images and audio/video records of experiments.

2.6 References

Aeschliman, D. P. and W. L. Oberkampf (1998). Experimental methodology for computational fluid dynamics code validation. *AIAA Journal*. **36**(5), 733–741.

AIAA (1998). Guide for the verification and validation of computational fluid dynamics simulations. *American Institute of Aeronautics and Astronautics*, *AIAA-G-077–1998*, Reston, VA.

Almond, R. G. (1995). *Graphical Belief Modeling*. 1st edn., London, Chapman & Hall.

Anderson, M. G. and P. D. Bates, eds. (2001). *Model Validation: Perspectives in Hydrological Science*. New York, NY, John Wiley & Sons Ltd.

Ang, A. H.-S. and W. H. Tang (2007). *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering*. 2nd edn., New York, Wiley.

ANS (1987). Guidelines for the verification and validation of scientific and engineering computer programs for the nuclear industry. *American Nuclear Society, ANSI/ANS-10.4–1987*, La Grange Park, IL.

Apostolakis, G. (1990). The concept of probability in safety assessments of technological systems. *Science*. **250**(4986), 1359–1364.

Arthur, J. D. and R. E. Nance (1996). Independent verification and validation: a missing link in simulation methodology? 1996 *Winter Simulation Conference*, Coronado, CA, 229–236.

ASME (2006). Guide for verification and validation in computational solid mechanics. *American Society of Mechanical Engineers*, ASME Standard V&V 10–2006, New York, NY.

Ayyub, B. M. (2001). *Elicitation of Expert Opinions for Uncertainty and Risks*, Boca Raton, FL, CRC Press.

Ayyub, B. M. and G. J. Klir (2006). *Uncertainty Modeling and Analysis in Engineering and the Sciences*, Boca Raton, FL, Chapman & Hall.

Bae, H.-R., R. V. Grandhi, and R. A. Canfield (2006). Sensitivity analysis of structural response uncertainty propagation using evidence theory. *Structural and Multidisciplinary Optimization*. **31**(4), 270–279.

Beck, M. B. (1987). Water quality modeling: a review of the analysis of uncertainty. *Water Resources Research*. **23**(8), 1393–1442.

Beven, K. (2002). Towards a coherent philosophy of modelling the environment. *Proceedings of the Royal Society of London, Series A*. **458**(2026), 2465–2484.

Blottner, F. G. (1990). Accurate Navier–Stokes results for the hypersonic flow over a spherical nosetip. *Journal of Spacecraft and Rockets*. **27**(2), 113–122.

Bogen, K. T. and R. C. Spear (1987). Integrating uncertainty and interindividual variability in environmental risk assessment. *Risk Analysis*. **7**(4), 427–436.

Bossel, H. (1994). *Modeling and Simulation*. 1st edn., Wellesley, MA, A. K. Peters.

Boughton, B., V. J. Romero, S. R. Tieszen, and K. B. Sobolik (2003). Integrated modeling and simulation validation plan for W80–3 abnormal thermal environment

qualification – Version 1.0 (OUO). *Sandia National Laboratories, SAND2003–4152 (OUO)*, Albuquerque, NM.
Boyack, B. E., I. Catton, R. B. Duffey, P. Griffith, K. R. Katsma, G. S. Lellouche, S. Levy, U. S. Rohatgi, G. E. Wilson, W. Wulff, and N. Zuber (1990). Quantifying reactor safety margins, Part 1: An overview of the code scaling, applicability, and uncertainty evaluation methodology. *Nuclear Engineering and Design*. **119**, 1–15.
Bradley, R. G. (1988). CFD validation philosophy. *Fluid Dynamics Panel Symposium: Validation of Computational Fluid Dynamics, AGARD-CP-437*, Lisbon, Portugal, North Atlantic Treaty Organization.
Cacuci, D. G. (2003). *Sensitivity and Uncertainty Analysis: Theory*, Boca Raton, FL, Chapman & Hall/CRC.
Carnap, R. (1963). Testability and meaning. *Philosophy of Science*. **3**(4), 419–471.
Casti, J. L. (1990). *Searching for Certainty: What Scientists Can Know About the Future*, New York, William Morrow.
Chiles, J.-P. and P. Delfiner (1999). *Geostatistics: Modeling Spatial Uncertainty*, New York, John Wiley.
Choi, S.-K., R. V. Grandhi, and R. A. Canfield (2007). *Reliability-based Structural Design*, London, Springer-Verlag.
Churchman, C. W. (1968). *The Systems Approach*, New York, Dell.
Coleman, H. W. and F. Stern (1997). Uncertainties and CFD code validation. *Journal of Fluids Engineering*. **119**, 795–803.
Cosner, R. R. (1995). CFD validation requirements for technology transition. *26th AIAA Fluid Dynamics Conference, AIAA Paper 95–2227*, San Diego, CA, American Institute of Aeronautics and Astronautics.
Cox, E. (1999). *The Fuzzy Systems Handbook: a Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*. 2nd edn., San Diego, CA, AP Professional.
Cullen, A. C. and H. C. Frey (1999). *Probabilistic Techniques in Exposure Assessment: a Handbook for Dealing with Variability and Uncertainty in Models and Inputs*, New York, Plenum Press.
Davis, P. A., N. E. Olague, and M. T. Goodrich (1991). Approaches for the validation of models used for performance assessment of high-level nuclear waste repositories. *Sandia National Laboratories, NUREG/CR-5537; SAND90–0575*, Albuquerque, NM.
Davis, P. K. (1992). Generalizing concepts and methods of verification, validation, and accreditation (VV&A) for military simulations. *RAND, R-4249-ACQ*, Santa Monica, CA.
de Cooman, G., D. Ruan, and E. E. Kerre, eds. (1995). *Foundations and Applications of Possibility Theory*. Singapore, World Scientific Publishing Co.
DoD (1994). *DoD Directive No. 5000.59: Modeling and Simulation (M&S) Management*. from www.msco.mil.
DoD (1996). *DoD Instruction 5000.61: Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*, Defense Modeling and Simulation Office, Office of the Director of Defense Research and Engineering.
DoD (1997). *DoD Modeling and Simulation Glossary*. from www.msco.mil.
DoD (2008). *Department of Defense Standard Practice: Documentation of Verification, Validation, and Accreditation (VV&A) for Models and Simulations*. US Washington, DC, Department of Defense.
Dowding, K. (2001). Quantitative validation of mathematical models. *ASME International Mechanical Engineering Congress Exposition*, New York, American Society of Mechanical Engineers.

Drosg, M. (2007). *Dealing with Uncertainties: a Guide to Error Analysis*, Berlin, Springer.
Dubois, D. and H. Prade (1988). *Possibility Theory: an Approach to Computerized Processing of Uncertainty*, New York, Plenum Press.
Dubois, D. and H. Prade, eds. (2000). *Fundamentals of Fuzzy Sets*. Boston, MA, Kluwer Academic Publishers.
Easterling, R. G. (2001a). Measuring the predictive capability of computational models: principles and methods, issues and illustrations. *Sandia National Laboratories, SAND2001–0243*, Albuquerque, NM.
Easterling, R. G. (2001b). "Quantifying the Uncertainty of Computational Predictions." *Sandia National Laboratories, SAND2001–0919C*, Albuquerque, NM.
Elishakoff, I. (2004). *Safety Factors and Reliability: Friends or Foes?*, Norwell, MA, Kluwer Academic Publishers.
Ferson, S. and L. R. Ginzburg (1996). Different methods are needed to propagate ignorance and variability. *Reliability Engineering and System Safety*. **54**, 133–144.
Ferson, S. and J. G. Hajagos (2004). Arithmetic with uncertain numbers: rigorous and (often) best possible answers. *Reliability Engineering and System Safety*. **85**(1–3), 135–152.
Ferson, S., R. B. Nelsen, J. Hajagos, D. J. Berleant, J. Zhang, W. T. Tucker, L. R. Ginzburg, and W. L. Oberkampf (2004). Dependence in probabilistic modeling, Dempster – Shafer theory, and probability bounds analysis. *Sandia National Laboratories, SAND2004–3072*, Albuquerque, NM.
Fetz, T., M. Oberguggenberger, and S. Pittschmann (2000). Applications of possibility and evidence theory in civil engineering. *International Journal of Uncertainty*. **8**(3), 295–309.
Gass, S. I. (1993). Model accreditation: a rationale and process for determining a numerical rating. *European Journal of Operational Research*. **66**, 250–258.
Grabe, M. (2005). *Measurement Uncertainties in Science and Technology*, Berlin, Springer.
Guan, J. and D. A. Bell (1991). *Evidence Theory and Its Applications*, Amsterdam, North Holland.
Haimes, Y. Y. (2009). *Risk Modeling, Assessment, and Management*. 3rd edn., New York, John Wiley.
Haldar, A. and S. Mahadevan (2000). *Probability, Reliability, and Statistical Methods in Engineering Design*, New York, John Wiley.
Helton, J. C. (1993). Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliability Engineering and System Safety*. **42**(2–3), 327–367.
Helton, J. C. (1994). Treatment of uncertainty in performance assessments for complex systems. *Risk Analysis*. **14**(4), 483–511.
Helton, J. C. (1997). Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *Journal of Statistical Computation and Simulation*. **57**, 3–76.
Helton, J. C. (1999). Uncertainty and sensitivity analysis in performance assessment for the waste isolation pilot plant. *Computer Physics Communications*. **117**(1–2), 156–180.
Helton, J. C., W. L. Oberkampf, and J. D. Johnson (2005). Competing failure risk analysis using evidence theory. *Risk Analysis*. **25**(4), 973–995.
Hills, R. G. and T. G. Trucano (1999). Statistical validation of engineering and scientific models: background. *Sandia National Laboratories, SAND99–1256*, Albuquerque, NM.

Hills, R. G. and T. G. Trucano (2001). Statistical validation of engineering and scientific models with application to CTH. *Sandia National Laboratories, SAND2001–0312*, Albuquerque, NM.

Hills, R. G. and T. G. Trucano (2002). Statistical validation of engineering and scientific models: a maximum likelihood based metric. *Sandia National Laboratories, SAND2001–1783*, Albuquerque, NM.

Hodges, J. S. and J. A. Dewar (1992). Is it you or your model talking? A framework for model validation. *RAND, R-4114-AF/A/OSD*, Santa Monica, CA.

Hoffman, F. O. and J. S. Hammonds (1994). Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. *Risk Analysis*. **14**(5), 707–712.

IEEE (1984). *IEEE Standard Dictionary of Electrical and Electronics Terms*. ANSI/IEEE Std 100–1984, New York.

IEEE (1991). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12–1990, New York.

ISO (1991). *ISO 9000–3: Quality Management and Quality Assurance Standards – Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software*. Geneva, Switzerland, International Organization for Standardization.

Kafka, P. (1994). Important issues using PSA technology for design of new systems and plants. *Reliability Engineering and System Safety*. **45**(1–2), 205–213.

Kaplan, S. and B. J. Garrick (1981). On the quantitative definition of risk. *Risk Analysis*. **1**(1), 11–27.

Kearfott, R. B. and V. Kreinovich, eds. (1996). *Applications of Interval Computations*. Boston, MA, Kluwer Academic Publishers.

Kleindorfer, G. B., L. O'Neill, and R. Ganeshan (1998). Validation in simulation: various positions in the philosophy of science. *Management Science*. **44**(8), 1087–1099.

Klir, G. J. (1969). *An Approach to General Systems Theory*, New York, NY, Van Nostrand Reinhold.

Klir, G. J. and M. J. Wierman (1998). *Uncertainty-Based Information: Elements of Generalized Information Theory*, Heidelberg, Physica-Verlag.

Klir, G. J. and B. Yuan (1995). *Fuzzy Sets and Fuzzy Logic*, Saddle River, NJ, Prentice Hall.

Klir, G. J., U. St. Clair, and B. Yuan (1997). *Fuzzy Set Theory: Foundations and Applications*, Upper Saddle River, NJ, Prentice Hall PTR.

Kloeden, P. E. and E. Platen (2000). *Numerical Solution of Stochastic Differential Equations*, New York, Springer.

Kohlas, J. and P.-A. Monney (1995). *A Mathematical Theory of Hints – an Approach to the Dempster – Shafer Theory of Evidence*, Berlin, Springer.

Konikow, L. F. and J. D. Bredehoeft (1992). Ground-water models cannot be validated. *Advances in Water Resources*. **15**, 75–83.

Kozine, I. (1999). Imprecise probabilities relating to prior reliability assessments. *1st International Symposium on Imprecise Probabilities and Their Applications*, Ghent, Belgium.

Krause, P. and D. Clark (1993). *Representing Uncertain Knowledge: an Artificial Intelligence Approach*, Dordrecht, The Netherlands, Kluwer Academic Publishers.

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. 3rd edn., Chicago and London, University of Chicago Press.

Kumamoto, H. and E. J. Henley (1996). *Probabilistic Risk Assessment and Management for Engineers and Scientists*. 2nd edn., New York, IEEE Press.

Law, A. M. (2006). *Simulation Modeling and Analysis*. 4th edn., New York, McGraw-Hill.
LeGore, T. (1990). Predictive software validation methodology for use with experiments having limited replicability. In *Benchmark Test Cases for Computational Fluid Dynamics*. I. Celik and C. J. Freitas (eds.). New York, American Society of Mechanical Engineers. FED-Vol. 93: 21–27.
Lewis, R. O. (1992). *Independent Verification and Validation*. 1st edn., New York, John Wiley.
Lin, S. J., S. L. Barson, and M. M. Sindir (1992). Development of evaluation criteria and a procedure for assessing predictive capability and code performance. *Advanced Earth-to-Orbit Propulsion Technology Conference*, Huntsville, AL, Marshall Space Flight Center.
Lipton, P. (2005). Testing hypotheses: prediction and prejudice. *Science*. **307**, 219–221.
Marvin, J. G. (1988). Accuracy requirements and benchmark experiments for CFD validation. *Fluid Dynamics Panel Symposium: Validation of Computational Fluid Dynamics, AGARD-CP-437*, Lisbon, Portugal, AGARD.
Marvin, J. G. (1995). Perspective on computational fluid dynamics validation. *AIAA Journal*. **33**(10), 1778–1787.
Mehta, U. B. (1990). The aerospace plane design challenge: credible computational fluid dynamics results. Moffett Field, NASA, TM 102887.
Melchers, R. E. (1999). *Structural Reliability Analysis and Prediction*. 2nd Edn., New York, John Wiley.
Modarres, M. (1993). *What Every Engineer Should Know about Reliability and Risk Analysis*, New York, Marcel Dekker.
Modarres, M., M. Kaminskiy, and V. Krivtsov (1999). *Reliability Engineering and Risk Analysis; a Practical Guide*, Boca Raton, FL, CRC Press.
Moore, R. E. (1979). *Methods and Applications of Interval Analysis*, Philadelphia, PA, SIAM.
Morgan, M. G. and M. Henrion (1990). *Uncertainty: a Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. 1st edn., Cambridge, UK, Cambridge University Press.
Naylor, T. H. and J. M. Finger (1967). Verification of computer simulation models. *Management Science*. **14**(2), 92–101.
NRC (1990). *Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants*. U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Division of Systems Research, NUREG-1150, Washington, DC.
Oberkampf, W. L. and D. P. Aeschliman (1992). Joint computational/experimental aerodynamics research on a hypersonic vehicle: Part 1, experimental results. *AIAA Journal*. **30**(8), 2000–2009.
Oberkampf, W. L. and M. F. Barone (2004). Measures of agreement between computation and experiment: validation metrics. *34th AIAA Fluid Dynamics Conference, AIAA Paper 2004–2626*, Portland, OR, American Institute of Aeronautics and Astronautics.
Oberkampf, W. L. and M. F. Barone (2006). Measures of agreement between computation and experiment: validation metrics. *Journal of Computational Physics*. **217**(1), 5–36.
Oberkampf, W. L. and F. G. Blottner (1998). Issues in computational fluid dynamics code verification and validation. *AIAA Journal*. **36**(5), 687–695.
Oberkampf, W. L. and S. Ferson (2007). Model validation under both aleatory and epistemic uncertainty. *NATO/RTO Symposium on Computational Uncertainty in Military Vehicle Design, AVT-147/RSY-022*, Athens, Greece, NATO.

Oberkampf, W. L. and J. C. Helton (2005). Evidence theory for engineering applications. In *Engineering Design Reliability Handbook*. E. Nikolaidis, D. M. Ghiocel, and S. Singhal, eds. New York, NY, CRC Press: 29.

Oberkampf, W. L. and T. G. Trucano (2000). Validation methodology in computational fluid dynamics. *Fluids 2000 Conference, AIAA Paper 2000–2549*, Denver, CO, American Institute of Aeronautics and Astronautics.

Oberkampf, W. L. and T. G. Trucano (2002). Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences*. **38**(3), 209–272.

Oberkampf, W. L. and T. G. Trucano (2007). *Verification and Validation Benchmarks*. Albuquerque, NM, Sandia National Laboratories, SAND2007–0853.

Oberkampf, W. L. and T. G. Trucano (2008). Verification and validation benchmarks. *Nuclear Engineering and Design*. **238**(3), 716–743.

Oberkampf, W. L., F. G. Blottner, and D. P. Aeschliman (1995). Methodology for computational fluid dynamics code verification/validation. *26th AIAA Fluid Dynamics Conference, AIAA Paper 95–2226*, San Diego, CA, American Institute of Aeronautics and Astronautics.

Oberkampf, W. L., T. G. Trucano, and C. Hirsch (2004). Verification, validation, and predictive capability in computational engineering and physics. *Applied Mechanics Reviews*. **57**(5), 345–384.

Oksendal, B. (2003). *Stochastic Differential Equations: an Introduction with Applications*. 6th edn., Berlin, Springer.

Oren, T. I. and B. P. Zeigler (1979). Concepts for advanced simulation methodologies. *Simulation*, 69–82.

Oreskes, N., K. Shrader-Frechette, and K. Belitz (1994). Verification, validation, and confirmation of numerical models in the Earth Sciences. *Science*. **263**, 641–646.

Paez, T. and A. Urbina (2001). Validation of structural dynamics models via hypothesis testing. *Society of Experimental Mechanics Annual Conference*, Portland, OR, Society of Experimental Mechanics.

Parry, G. W. (1988). On the meaning of probability in probabilistic safety assessment. *Reliability Engineering and System Safety*. **23**, 309–314.

Parry, G. W. and P. W. Winter (1981). Characterization and evaluation of uncertainty in probabilistic risk analysis. *Nuclear Safety*. **22**(1), 28–41.

Paté-Cornell, M. E. (1996). Uncertainties in risk analysis: six levels of treatment. *Reliability Engineering and System Safety*. **54**, 95–111.

Pilch, M., T. G. Trucano, J. L. Moya, G. K. Froehlich, A. L. Hodges, and D. E. Peercy (2001). *Guidelines for Sandia ASCI Verification and Validation Plans – Content and Format: Version 2*. Albuquerque, NM, Sandia National Laboratories, SAND2000–3101.

Platt, D. S. (2007). *Why Software Sucks . . . and what you can do about it*, Upper Saddle River, NJ, Addison-Wesley.

Popper, K. R. (1959). *The Logic of Scientific Discovery*, New York, Basic Books.

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

Rabinovich, S. G. (2005). *Measurement Errors and Uncertainties: Theory and Practice*. 3rd edn., New York, Springer-Verlag.

Raczynski, S. (2006). *Modeling and Simulation: the Computer Science of Illusion*, New York, Wiley.

Refsgaard, J. C. and H. J. Henriksen (2004). Modelling guidelines – terminology and guiding principles. *Advances in Water Resources*. **27**, 71–82.

Roache, P. J. (1990). Need for control of numerical accuracy. *Journal of Spacecraft and Rockets*. **27**(2), 98–102.
Roache, P. J. (1995). Verification of codes and calculations. *26th AIAA Fluid Dynamics Conference, AIAA Paper 95–2224*, San Diego, CA, American Institute of Aeronautics and Astronautics.
Roache, P. J. (1998). *Verification and Validation in Computational Science and Engineering*, Albuquerque, NM, Hermosa Publishers.
Roza, Z. C. (2004). *Simulation Fidelity, Theory and Practice: a Unified Approach to Defining, Specifying and Measuring the Realism of Simulations*, Delft, The Netherlands, Delft University Press.
Rubinstein, R. Y. and D. P. Kroese (2008). *Simulation and the Monte Carlo Method*. 2nd edn., Hoboken, NJ, John Wiley.
Rykiel, E. J. (1996). Testing ecological models: the meaning of validation. *Ecological Modelling*. **90**(3), 229–244.
Saltelli, A., S. Tarantola, F. Campolongo, and M. Ratto (2004). *Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models*, Chichester, England, John Wiley & Sons, Ltd.
Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008). *Global Sensitivity Analysis: the Primer*, Hoboken, NJ, Wiley.
Sargent, R. G. (1979). Validation of simulation models. *1979 Winter Simulation Conference*, San Diego, CA, 497–503.
Schlesinger, S. (1979). Terminology for model credibility. *Simulation*. **32**(3), 103–104.
Serrano, S. E. (2001). *Engineering Uncertainty and Risk Analysis: a Balanced Approach to Probability, Statistics, Stochastic Modeling, and Stochastic Differential Equations*, Lexington, KY, HydroScience Inc.
Shannon, R. E. (1975). *Systems Simulation: the Art and Science*, Englewood Cliffs, NJ, Prentice-Hall.
Sheng, G., M. S. Elzas, T. I. Oren, and B. T. Cronhjort (1993). Model validation: a systemic and systematic approach. *Reliability Engineering and System Safety*. **42**, 247–259.
Sindir, M. M., S. L. Barson, D. C. Chan, and W. H. Lin (1996). On the development and demonstration of a code validation process for industrial applications. *27th AIAA Fluid Dynamics Conference, AIAA Paper 96–2032*, New Orleans, LA, American Institute of Aeronautics and Astronautics.
Smithson, M. (1989). *Ignorance and Uncertainty: Emerging Paradigms*, New York, Springer-Verlag.
Stern, F., R. V. Wilson, H. W. Coleman, and E. G. Paterson (2001). Comprehensive approach to verification and validation of CFD simulations – Part 1: Methodology and procedures. *Journal of Fluids Engineering*. **123**(4), 793–802.
Stockman, C. T., J. W. Garner, J. C. Helton, J. D. Johnson, A. Shinta, and L. N. Smith (2000). Radionuclide transport in the vicinity of the repository and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliability Engineering and System Safety*. **69**(1–3), 369–396.
Suter, G. W. (2007). *Ecological Risk Assessment*. 2nd edn., Boca Raton, FL, CRC Press.
Taylor, H. M. and S. Karlin (1998). *An Introduction to Stochastic Modeling*. 3rd edn., Boston, Academic Press.
Tieszen, S. R., T. Y. Chu, D. Dobranich, V. J. Romero, T. G. Trucano, J. T. Nakos, W. C. Moffatt, T. F. Hendrickson, K. B. Sobolik, S. N. Kempka, and M. Pilch (2002).

Integrated Modeling and Simulation Validation Plan for W76–1 Abnormal Thermal Environment Qualification – Version 1.0 (OUO). Sandia National Laboratories, SAND2002–1740 (OUO), Albuquerque.

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

Trucano, T. G., M. Pilch, and W. L. Oberkampf (2002). *General Concepts for Experimental Validation of ASCI Code Applications*. Albuquerque, NM, Sandia National Laboratories, SAND2002–0341.

Tsang, C.-F. (1989). A broad view of model validation. *Proceedings of the Symposium on Safety Assessment of Radioactive Waste Repositories, Paris, France*, Paris, France, OECD, 707–716.

Urbina, A. and T. L. Paez (2001). Statistical validation of structural dynamics models. *Annual Technical Meeting & Exposition of the Institute of Environmental Sciences and Technology*, Phoenix, AZ.

Vose, D. (2008). *Risk Analysis: a Quantitative Guide*. 3rd edn., New York, Wiley.

Walley, P. (1991). *Statistical Reasoning with Imprecise Probabilities*, London, Chapman & Hall.

Wilson, G. E. and B. E. Boyack (1998). The role of the PIRT in experiments, code development and code applications associated with reactor safety assessment. *Nuclear Engineering and Design*. **186**, 23–37.

Wilson, G. E., B. E. Boyack, I. Catton, R. B. Duffey, P. Griffith, K. R. Katsma, G. S. Lellouche, S. Levy, U. S. Rohatgi, W. Wulff, and N. Zuber (1990). Quantifying reactor safety margins, Part 2: Characterization of important contributors to uncertainty. *Nuclear Engineering and Design*. **119**, 17–31.

Wulff, W., B. E. Boyack, I. Catton, R. B. Duffey, P. Griffith, K. R. Katsma, G. S. Lellouche, S. Levy, U. S. Rohatgi, G. E. Wilson, and N. Zuber (1990). Quantifying reactor safety margins, Part 3: Assessment and ranging of parameters. *Nuclear Engineering and Design*. **119**, 33–65.

Zeigler, B. P. (1976). *Theory of Modelling and Simulation*. 1st edn., New York, John Wiley.

Zeigler, B. P., H. Praehofer, and T. G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2nd edn., San Diego, CA, Academic Press.

Zuber, N., G. E. Wilson, M. Ishii, W. Wulff, B. E. Boyack, A. E. Dukler, P. Griffith, J. M. Healzer, R. E. Henry, J. R. Lehner, S. Levy, and F. J. Moody (1998). An integrated structure and scaling methodology for severe accident technical issue resolution: development of methodology. *Nuclear Engineering and Design*. **186**(1–2), 1–21.