

# **Nuclear Energy – Knowledge Base for Advanced Modeling and Simulation (NE-KAMS) Code Verification and Validation Data Standards and Requirements: Fluid Dynamics Version 1.0**

V. Gregory Weirs  
Kimberlyn C. Mousseau  
Richard W. Johnson  
Hyung Lee  
William L. Oberkampf  
Bart Smith  
James Fort  
William J. Rider

September 2011

The INL is a U.S. Department of Energy National Laboratory  
operated by Battelle Energy Alliance



**Nuclear Energy – Knowledge Base for Advanced  
Modeling and Simulation (NE-KAMS)  
Code Verification and Validation  
Data Standards and Requirements: Fluid Dynamics  
Version 1.0**

**V. Gregory Weirs<sup>1</sup>  
Kimberlyn C. Mousseau  
Richard W. Johnson  
Hyung Lee<sup>2</sup>  
William L. Oberkampf<sup>3</sup>  
Bart Smith<sup>4</sup>  
James Fort<sup>5</sup>  
William J. Rider<sup>1</sup>**

<sup>1</sup>Sandia National Laboratory

<sup>2</sup>Bettis Laboratory

<sup>3</sup>WLO Consulting

<sup>4</sup>Utah State University

<sup>5</sup>Pacific Northwest National Laboratory

**September 2011**

**Idaho National Laboratory  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

**Prepared for the  
U.S. Department of Energy  
Office of Nuclear Energy  
Under DOE Idaho Operations Office  
Contract DE-AC07-05ID14517**



## EXECUTIVE SUMMARY

The U.S. Department of Energy, Office of Nuclear Energy (DOE-NE), has been tasked with the important mission of ensuring that nuclear energy remains a compelling and viable energy source in the U.S. The motivations behind this mission include cost-effectively meeting the expected increases in the power needs of the country, reducing carbon emissions, and reducing dependence on foreign energy sources. In the near term, to ensure that nuclear power remains a key element of U.S. energy strategy, the DOE-NE will be working with the nuclear industry to support safe and efficient operations of existing nuclear power plants. In the long term, the DOE-NE will be investing in research and development (R&D) and working in concert with the nuclear industry to build new, safer and more efficient nuclear power plants. The safe and efficient operations of existing nuclear power plants and designing and licensing new reactor designs, however, will require the extensive use of advanced modeling and simulation (M&S). Thus, it is expected that M&S will play a key role in ensuring the safe and efficient operation of existing and new nuclear reactors.

The complexity of nuclear reactor power plants, as well as the cost and difficulty associated with testing nuclear reactor systems makes the use of M&S a desirable tool for nuclear reactor design, analysis and licensing. Thus, engineering analysis and performance characterization of existing and new reactor designs will employ advanced M&S tools, such as computational fluid dynamics (CFD) and computational structural mechanics (CSM), in addition to the traditional thermal hydraulics (T/H) and systems analysis codes. The DOE-NE, in fact, has been actively developing and promoting the use of advanced M&S in reactor design and analysis through its R&D programs, *e.g.*, the Nuclear Energy Advanced Modeling and Simulation (NEAMS) and Consortium for Advanced Simulation of Light Water Reactors (CASL) programs. Also, nuclear reactor vendors are already using CFD and CSM, for design, analysis, and licensing. However, these M&S tools cannot be used with confidence for nuclear reactor applications unless supported by verification and validation (V&V) and uncertainty quantification (UQ) which provide quantitative measures of uncertainty for specific applications.

V&V and UQ are the primary means to assess the accuracy and reliability of M&S and, hence, to establish confidence in M&S. Though the nuclear industry has established standards and processes for carrying out V&V and UQ for systems analysis codes and simulations, at present, similar standards and processes for high fidelity M&S tools such as CFD have not reached the same level of maturity. However, the nuclear industry recognizes that such standards and processes are needed and that the resources required to support V&V and UQ for CFD for nuclear applications is significant. In fact, no single organization, whether a commercial company or government laboratory, has the resources required to organize, develop and maintain the needed V&V and UQ program. What is needed is a standardized program for V&V and UQ at a national or even international level, with a consortium of partners from government, academia and industry. Specifically, what is needed is a structured knowledge base for V&V and UQ that collects, evaluates and maintains V&V and UQ data and information, and provides guidance and resources for performing V&V and UQ assessments. This kind of a knowledge base can promote collaboration and provide for sharing of resources needed to support the use of M&S for engineering and licensing applications.

The Nuclear Energy Knowledge base for Advanced Modeling and Simulation (NE-KAMS) is being developed at the Idaho National Laboratory in conjunction with Bettis Laboratory, Sandia National Laboratories, Argonne National Laboratory, Utah State University and others. The objective of this consortium is to establish a comprehensive and web-accessible knowledge base to provide V&V and UQ resources for M&S for nuclear reactor design, analysis and licensing. The knowledge base will serve as an important resource for technical exchange that will enable credible computational models and simulations for application to nuclear power. NE-KAMS will serve as a valuable resource for the nuclear industry, academia, the national laboratories, the U.S. Nuclear Regulatory Commission (NRC), and the public, and will help ensure the safe, economical and reliable operation of existing and future nuclear reactors.

Existing V&V databases, such as the European Research Community on Flow, Turbulence and Combustion (ERCOFTAC), the European QNET-CFD, and the U.S. NPARC Alliance databases, serve only as repositories for V&V cases. In addition to its V&V benchmark databases, the NE-KAMS knowledge base will provide guidance for progressive levels of V&V completeness, assessment of V&V benchmarks, and improved best practices for V&V. Specifically, the NE-KAMS knowledge base will assist analysts, model developers, experimentalists, designers, and regulators by:

- Establishing accepted standards, requirements and best practices for V&V and UQ of computational models and simulations,
- Establishing accepted standards and procedures for assessing experimental and numerical benchmark data,
- Providing readily accessible databases with long-term storage for code verification and validation benchmark data that can be used in V&V assessments and methods development,
- Providing a searchable knowledge base of information, documents and data on V&V and UQ, and
- Providing web-enabled tools and utilities for V&V and UQ activities, data assessment and processing, and information and data searches.

From its inception, NE-KAMS will directly support nuclear energy R&D programs within the U.S. Department of Energy (DOE), including CASL, NEAMS, Light Water Reactor Sustainability (LWRS), Small Modular Reactors (SMR), and Next Generation Nuclear Power Plant (NGNP). These programs all involve M&S for nuclear reactor systems and it is envisioned that NE-KAMS will facilitate collaboration and sharing of resources and expertise across these programs. The initial focus for the NE-KAMS effort will be on supporting the use of CFD for simulation of nuclear reactor systems, but will eventually expand to include, materials, fuel system performance and other areas of M&S as time and funding allow.

## **Verification and Validation**

Code verification addresses two issues: (a) the mathematical correctness of the numerical methods to obtain numerical solutions to the mathematical model describing the system of interest, and (b) the correctness of the computer code that implements the numerical solution of the mathematical model. A key part of code verification is to find exact solutions to the mathematical model so that the numerical solution software can be tested. This report defines six attributes of exact and numerical solutions that should be used to assess the accuracy, quality,

and usefulness of a benchmark solution that could be used for code verification. Four levels of information completeness are described for each of these attributes in order to assess benchmark solutions.

Validation addresses the ability of a mathematical model to accurately predict system response quantities for an intended use. The primary procedure for estimating model predictive accuracy is comparison of simulation results with experimental measurements. To provide a definitive assessment of model accuracy for the specific conditions of the experiment, the experiment must attain many of the features of a high quality validation experiment. This report defines six attributes of experimental activities that should be used to assess the accuracy, quality, and usefulness of experiments used for model validation. Four levels of information completeness are described for each of these attributes in order to assess validation experiments.

The verification and validation attributes and the completeness levels are focused on CFD, but the fundamental framework can be applied to other fields, such as materials and fuel system performance. It is believed that this document represents the first time that carefully defined attributes and completeness levels for verification and validation benchmarks have been defined.

# Table of Contents

1.	Introduction .....	1
1.1	Background.....	1
1.2	Purpose.....	2
1.3	Scope.....	3
2.	Verification and Validation: Definitions and Processes .....	4
2.1	The Context of Scientific Simulations.....	5
2.2	Taxonomy .....	6
2.3	The Relationship Between Different Methods of Assessment.....	9
2.4	Code Verification .....	11
2.5	Validation.....	13
3.	Code Verification Benchmark Data Standards and Requirements .....	15
3.1	Introduction.....	15
3.2	Description of Verification Benchmark Data and Related Artifacts .....	15
3.3	Completeness Classification of Code Verification Benchmark Data.....	16
3.4	Connection to NE-KAMS V&V Database .....	19
4.	Validation Benchmark Data Standards and Requirements.....	25
4.1	Context.....	25
4.2	Description of Validation Benchmark Data and Metadata .....	26
4.3	Fundamentals of Sensors and Transducers .....	27
4.4	Completeness Classification of Validation Benchmark Data.....	28
5.	NE-KAMS Support for Nuclear Reactor Design, Safety Analysis and Licensing .....	35
5.1	Computational Fluid Dynamics for Nuclear Reactor Safety Analysis .....	35
5.2	Nuclear Reactor Safety Problems for Which CFD Could Bring Real Benefits.....	36
6.	Concluding Remarks .....	41
7.	References .....	42
8.	APPENDIX A: Examples of Required Information for Each Level of the Verification Benchmark Completeness Table .....	44
8.1	Conceptual Description.....	44
8.2	Mathematical Description.....	44
8.3	Boundary and/or Initial Conditions .....	45
8.4	Computer Hardware and System Software .....	45
8.5	Software Quality Assurance (SQA) and Code Verification .....	46
8.6	Solution Accuracy Assessment .....	47
9.	APPENDIX B: Examples of Required Information for the Validation Benchmark Completeness Table .....	49
9.1	Experimental Facility.....	49
9.2	Analog Instrumentation and Signal Processing.....	51
9.3	Boundary and Initial Conditions .....	52
9.4	Fluid and Material Properties of the Walls.....	54
9.5	Test Conditions.....	55
9.6	Measurement of Experimental Responses.....	56

## ACRONYMS

ANL	Argonne National Laboratory
CASL	Consortium for Advanced Simulation of Light Water Reactors
CFD	computational fluid dynamics
CSAU	Code Scaling, Applicability and Uncertainty
CSM	computational structural mechanics
DOE	Department of Energy
DOE-NE	Department of Energy – Office of Nuclear Energy
EPRI	Electric Power Research Institute
INL	Idaho National Laboratory
LWRS	Light Water Reactor Sustainability
M&S	modeling and simulation
NE-KAMS	Nuclear Energy Knowledge base for Advanced Modeling and Simulation
NEAMS	Nuclear Energy Advanced Modeling and Simulation
NGNP	Next Generation Nuclear Plant
NNSA	National Nuclear Security Administration
NRC	Nuclear Regulatory Commission
R&D	Research and Development
SMR	Small Modular Reactors
SNL	Sandia National Laboratories
UQ	uncertainty quantification
USU	Utah State University
V&V	verification and validation

## 1. Introduction

One of the key components of the Nuclear Energy Knowledge base for Advanced Modeling and Simulation (NE-KAMS) is its verification and validation (V&V) benchmark databases. The NE-KAMS Code Verification and Validation Data Standards and Requirements document describes the V&V data standards and requirements that will serve as the technical basis for the development and use of these databases. It defines attributes that should be used to assess the accuracy, quality, and usefulness of benchmark data, and standards of completeness, or completeness levels, for these attributes. These attributes and completeness levels provide a framework to evaluate the quality of exact and numerical solutions for code verification and experimental data for model validation. In this document, the V&V attributes and the completeness levels are focused on computational fluid dynamics (CFD), but the fundamental framework can be applied to other fields, such as materials and fuel system performance. It is believed that this document represents the first time that carefully defined attributes and completeness levels for V&V benchmarks have been defined.

### 1.1 Background

The U.S. Department of Energy, Office of Nuclear Energy (DOE-NE), has been tasked with the important mission of ensuring that nuclear energy remains a compelling and viable energy source in the U.S. The motivations behind this mission include cost-effectively meeting the expected increases in the power needs of the country, reducing carbon emissions, and reducing dependence on foreign energy sources. In the near term, to ensure that nuclear power remains a key element of U.S. energy strategy, the DOE-NE will be working with the nuclear industry to support safe and efficient operations of existing nuclear power plants. In the long term, the DOE-NE will be investing in research and development (R&D) and working in concert with the nuclear industry to build new, safer and more efficient nuclear power plants. The safe and efficient operations of existing nuclear power plants and designing and licensing new reactor designs, however, will require the extensive use of advanced modeling and simulation (M&S). Thus, it is expected that M&S will play a key role in ensuring the safe and efficient operation of existing and new nuclear reactors.

The complexity of nuclear reactor power plants, as well as the cost and difficulty associated with testing nuclear reactor systems makes the use of M&S a desirable tool for nuclear reactor design, analysis and licensing. Thus, engineering analysis and performance characterization of existing and new reactor designs will employ advanced M&S tools, such as CFD and computational structural mechanics (CSM), in addition to the traditional thermal hydraulics (T/H) and systems analysis codes. The DOE-NE, in fact, has been actively developing and promoting the use of advanced M&S in reactor design and analysis through its R&D programs, *e.g.*, the Nuclear Energy Advanced Modeling and Simulation (NEAMS) and Consortium for Advanced Simulation of Light Water Reactors (CASL) programs. Also, nuclear reactor vendors are already using CFD and CSM, for design, analysis, and licensing. However, these M&S tools cannot be used with confidence for nuclear reactor applications unless supported by verification and validation (V&V) and uncertainty quantification (UQ) which provide quantitative measures of uncertainty for specific applications.

V&V and UQ are the primary means to assess the accuracy and reliability of M&S and, hence, to establish confidence in M&S. Though the nuclear industry has established standards and processes for carrying out V&V and UQ for systems analysis codes and simulations, at present, similar standards and processes for high fidelity M&S tools such as CFD have not reached the same level of maturity. However, the nuclear industry recognizes that such standards and processes are needed and that the resources required to support V&V and UQ for CFD for nuclear applications is significant. In fact, no single organization, whether a commercial company or government laboratory, has the resources required to organize, develop and maintain the needed V&V and UQ program. What is needed is a standardized program for V&V and UQ at a national or even international level, with a consortium of partners from government, academia and industry. Specifically, what is needed is a structured knowledge base for V&V and UQ that collects, evaluates and maintains V&V and UQ data and information, and provides guidance and resources for performing V&V and UQ assessments. This kind of a knowledge base can promote collaboration and provide for sharing of resources needed to support the use of M&S for engineering and licensing applications.

## **1.2 Purpose**

The NE-KAMS knowledge base is being developed at the Idaho National Laboratory (INL) in conjunction with Bettis Laboratory, Sandia National Laboratories (SNL), Argonne National Laboratory (ANL), Utah State University and others. The objective of this consortium is to establish a comprehensive and web-accessible knowledge base to provide V&V and UQ resources for M&S for nuclear reactor design, analysis and licensing. The knowledge base will serve as an important resource for technical exchange that will enable credible computational models and simulations for application to nuclear power. NE-KAMS will serve as a valuable resource for the nuclear industry, academia, the national laboratories, the U.S. Nuclear Regulatory Commission (NRC), and the public, and will help ensure the safe, economical and reliable operation of existing and future nuclear reactors.

NE-KAMS will provide clear and specific guidance and recommendations for performing V&V and UQ for modeling and simulation in support of nuclear reactor system design, safety analysis and licensing. A critical component of V&V is to make quantitative comparisons of computational results to numerical or experimental benchmarks. It is a primary objective for NE-KAMS to provide numerical and experimental benchmarks for use in such V&V assessments. While it is usually desired to use high quality benchmark data, it is well known that not all data are of high accuracy or are documented completely enough to establish all of the necessary boundary and auxiliary conditions for simulations of the problem in question. Therefore, it is of great interest to particular stakeholders to have a measure of these characteristics for a particular benchmark dataset.

Existing V&V databases, such as the European Research Community on Flow, Turbulence and Combustion (ERCOFTAC), the European QNET-CFD database, and the U.S. NPARC Alliance database, serve only as repositories for V&V cases. In addition to its V&V benchmark databases, the NE-KAMS knowledge base will provide guidance for progressive levels of V&V completeness, assessment of V&V benchmarks, and improved best practices for V&V. Specifically, the NE-KAMS knowledge base will assist analysts, model developers, experimentalists, designers, and regulators by:

- Establishing accepted standards, requirements and best practices for V&V and UQ of computational models and simulations,
- Establishing accepted standards and procedures for assessing numerical and experimental benchmark data,
- Providing readily accessible databases with long-term storage for code verification and validation benchmark data that can be used in V&V assessments and methods development,
- Providing a searchable knowledge base of information, documents and data on V&V and UQ, and
- Providing web-enabled tools and utilities for V&V and UQ activities, data assessment and processing, and information and data searches.

From its inception, NE-KAMS will directly support nuclear energy R&D programs within the U.S. Department of Energy (DOE), including CASL, NEAMS, Light Water Reactor Sustainability (LWRS), Small Modular Reactors (SMR), and Next Generation Nuclear Power Plant (NGNP). These programs all involve M&S for nuclear reactor systems and it is envisioned that NE-KAMS will facilitate collaboration and sharing of resources and expertise across these programs. The initial focus for the NE-KAMS effort will be on supporting the use of CFD for simulation of nuclear reactor systems, but will eventually expand to include, materials, fuel system performance and other areas of M&S as time and funding allow.

### **1.3 Scope**

The initial focus for the NE-KAMS effort will be on supporting the use of CFD for modeling and simulation of nuclear reactor systems, but will eventually expand to include materials, fuel system performance and other areas of M&S as time and funding allow. The present document opens with an outline of V&V assessment techniques, with particular emphasis on strong-sense benchmarks (Oberkampf and Trucano (2002)). The main part of the document presents a benchmark evaluation processes for V&V benchmarks in CFD for nuclear reactor design and safety analysis applications. Standards of completeness, or completeness levels, are defined for various attributes benchmarks. Using these standards, NE-KAMS intends to assess contributed benchmarks for their value in conducting code verification or validation (depending on the type of benchmark). Finally, a number of specific nuclear reactor safety problems for which CFD (with appropriate V&V and UQ) can have an immediate impact are identified. As reflected in the title, this document is a living document. The evaluation procedures for V&V benchmarks will be updated and improved as experience is gained and new insights are unveiled.

## 2. Verification and Validation: Definitions and Processes

Nuclear reactor power plants increasingly rely on modeling and simulation for design, analysis, and a variety of other purposes, including analyzing scenarios that are far from normal operating conditions. To establish trust in M&S commensurate with the reliance on simulation results, simulation codes must demonstrate adequate accuracy for their intended use. Therefore, accuracy assessments of simulation codes are the critical foundation on which trust in simulations rests.

The only way to assess the accuracy of simulation codes is to test them against accurate data. Code verification and validation are widely accepted processes for assessing codes and simulations, but clear statements about the properties of the V&V data are less common and less well established. In the case of code verification for fluid dynamics, the data represent a solution to a test problem; a particular flow situation defined by an appropriate set of equations, some geometrical and fluid property descriptions, and initial and boundary conditions. For validation, the data are provided by measurements, possibly post-processed, from a physical experiment with a similar set of definitions. Particular test problems and experimental flow datasets are often called benchmarks.

The focus of this document is to define standards of completeness, or completeness levels, for attributes of benchmarks. Attributes are categories of information about benchmarks, and completeness refers to how thoroughly or extensively this information is investigated and documented. We consider two types of benchmarks: numerical solutions to the mathematical model for use in code verification, and physical experiments for use in validation. These standards are needed. Many popular numerical benchmarks do not have highly accurate solutions, so they cannot be used to determine if a code is truly performing as intended. Existing experimental datasets are generally not characterized well enough to draw strong conclusions from validation activities. Most existing experimental datasets were designed for purposes other than validation, and validation efforts that use them are less informative as a result.

The concept of *strong-sense benchmarks* (SSBs) was formulated in Oberkampf and Trucano (2002). SSBs should be of a high quality such that they can be viewed as engineering reference standards. SSBs are test problems that have the following four characteristics: (1) the purpose of the benchmark is clearly understood, (2) the definition and description of the benchmark is precisely stated, (3) specific requirements are stated for how comparisons are to be made with the results of the benchmark, and (4) acceptance criteria for comparison with the benchmark are defined. This report is an initial attempt at constructing benchmark evaluation processes for verification and validation benchmarks in CFD, with the highest evaluations levels corresponding to SSBs.

NE-KAMS intends to assess contributed benchmarks for their value in conducting code verification or validation (depending on the type of benchmark). Contributed benchmarks will be assessed against the completeness levels for the different attributes, with the highest completeness levels corresponding closely to SSBs. One key motivation for this work is to establish stronger communications and interactions between analysts conducting simulations, mathematical model developers, code developers, and experimentalists, through benchmarks and their assessment. While NE-KAMS will provide guidance and best practices for using benchmarks, it will not assess or archive verification or validation activities.

Before introducing the completeness levels for benchmark attributes, the technical context in which they are to be used must be described. Section 2.1 provides this context and begins with an overview of the conceptual steps that produce a simulation code. A number of assessment

techniques are defined in Section 2.2, and then a framework for combining them in a complementary way for conducting verification and validation is presented in Section 2.3. Finally there are two sections motivating high quality benchmarks for code verification and validation. These sections explain why SSBs are more effective than other benchmarks for V&V, and how ultimately, high-quality benchmarks allow more definitive statements about the accuracy of models and their suitability for particular applications.

## 2.1 The Context of Scientific Simulations

### Model hierarchies, equations as models, and approximate solutions

For scientific simulations, one starts with governing equations. These equations describe the system of interest, and encapsulate knowledge of physics, chemistry, and mathematics. The equations constitute a *mathematical model* of the physical world, and may be made up of submodels, or accept other submodels as input to the problem to be solved. The equations are often field equations – partial differential equations or integro-differential equations – that describe a time-varying solution over some volume of space or material, subject to initial and boundary conditions. Submodels could describe the particular response of a material, of interactions between materials such as chemical or nuclear reactions, or some aspect of the governing equations that is approximated or makes the equations easier to solve. Models, including the governing equations, have a range or regime of validity, outside of which the model is not expected to be valid; a regime boundary may be sharp, but more often the loss of validity is gradual or not defined at all. “Valid”, in the generic sense, means that the model is expected to give an accurate prediction for a particular use.

The problem, then, is to find a solution to the mathematical model given the initial and boundary conditions and the appropriate submodels. For problems of interest in nuclear reactors, the equations are often difficult to solve, the initial and boundary conditions may be inaccurate or, in the worst case, entirely unknown, and submodels may have known and unknown deficiencies.

One approach to obtaining solutions is through numerical simulations. The governing equations are discretized by *algorithms*, or *numerical methods*, and as a result the mathematical model is transformed to a discrete model, or a *numerical model*. Solutions of the numerical model (numerical solutions) are *approximate* solutions of the mathematical model. To obtain numerical solutions, the numerical model is implemented in software on a computer; this implementation is often called a computer *code*, and an execution of the code is called a *simulation*. The numerical methods considered here have one or more parameters that specify the number of discrete degrees of freedom, such as the number of elements, the number of particles, or the number of discrete wave numbers, and these parameters control the accuracy and the computational cost of the simulation.

Sometimes the purpose of numerical simulations is to provide insight into models and descriptions of reality. In particular, in most research contexts, simulations are used as “discovery” experiments. For this use, the precision and even the correctness of the simulations may be less important than the ideas and thinking they inspire or the trends they confirm. In this document, however, we are focused on the use of numerical simulations to provide predictions about the behavior of devices and systems, with sufficient precision that decisions of consequence can be made based on the predictions. In this latter context, the prediction is much more valuable if it comes with an estimate of the quality of the prediction – the accuracy of the prediction, or the uncertainty in the prediction.

## **Different assessments for different sources of error**

Given the context as explained above, if one provides a numerical prediction, can an estimate of the accuracy of that prediction be made? What are the sources of error in the prediction? That is, why might the predictive result of a simulation be different from the actual behavior of a device or system?

As anyone that has used a computer knows, software is not perfect. Programming mistakes, or bugs, are the main reason that software does not produce the results intended by the code developer. Programmers have developed a number of practices to identify, reduce, and eliminate bugs, but scientific simulation software presents a unique challenge: scientific simulations are conducted to provide approximate solutions to problems for which the answers are not known. As a result, special testing techniques are required, and *code verification* is the process accepted by the scientific simulation community to find programming errors in numerical methods that are undetected by general software testing strategies.

Once code verification has established that a code correctly implements the numerical methods as intended, there remain three sources of error in predictions from numerical simulations. The first is that the governing equations do not adequately describe the physics; the real device or system is affected by some phenomena that are not included in the model or submodels. Numerical methods are the second source of error. Recall that numerical simulations provide approximate solutions to the governing equations. A third source of error or uncertainty is that the inputs needed for the simulation are not known precisely. For example, the device to be simulated can only be constructed to within manufacturing tolerances; parameters in a model or submodel may not be known precisely, or may be subjective estimates from experts. Different assessment techniques – *validation*, *solution verification*, and *uncertainty quantification* – are used to estimate the errors from these three different sources. In any simulation, all three sources of error are present and contribute to the uncertainty in the prediction. From this it should be clear that estimating errors from just a single source, e.g., performing uncertainty quantification without validation or any form of verification, cannot provide the total uncertainty in a prediction.

Eliminating uncertainty in a prediction is neither practical nor necessary. It is sufficient to control, or manage, the uncertainty to an acceptable level for making the decisions that motivated the simulations. If the level of uncertainty is too high, the different assessments identify the largest sources of uncertainty and indicate where effort should be spent to reduce the uncertainty, e.g., reducing discretization error, improving physics modeling, or more precisely defining the problem.

## **2.2 Taxonomy**

In this section definitions are given for a number of key terms used in this document. Various types of assessments of numerical simulations are identified, as are other activities that are sometimes confused with assessments. More detail is provided on code verification and validation in following sections.

### **Code verification**

Code verification is the process of testing whether or not a computer code implements the numerical methods or algorithms as intended. The failure of a code to function as intended is

most commonly caused by (a) a programming mistake, or (b) an algorithm, though implemented correctly, provides inaccurate solutions or behaves poorly.

Code verification is primarily accomplished through benchmarks with exact solutions; numerical solutions produced by the code are compared to the independently computed exact solutions. The value of a code verification test is directly related to the completeness of the description of the particular exact solution used and the accuracy to which the exact solution can be computed. Factors that define the completeness and accuracy of exact solutions are the topic of Section 3.

### **Software verification**

Software quality assurance (SQA), software quality engineering (SQE), and software engineering are names for related processes that focus on high-quality software development that can be considered *software* verification. As opposed to the more specific testing of numerical methods that is the goal of code verification, software verification includes version control, regression testing, and other techniques of code development in team environments. While software verification is a process recommended by NE-KAMS as necessary for producing high-quality codes, it is not sufficient for testing scientific simulation software.

### **Solution Verification**

Solution verification, also called calculation verification, is a process for estimating numerical error, which arises when a solution technique provides an approximate solution. The numerical error is the difference between the exact solution of a model and the approximate solution provided by the solution technique. Numerical error does *not* include physics modeling error or approximations, and does *not* involve comparisons to physical experiments.

In scientific simulations, there are three types of numerical errors: round off error, iterative solver convergence error, and discretization error. Round off error is the result of the finite precision representation of quantitative values, and finite precision arithmetic on computers. Some numerical methods use iterative techniques to simultaneously solve systems of equations, and such techniques end the iteration procedure when some pre-set error or convergence tolerance has been reached. By not iterating to machine precision, the solution to the system of equations retains some error, called iterative solver error. The source of discretization error is the numerical method itself; for a well-designed method, as the number of discrete degrees of freedom is increased, the numerical error is reduced. For mesh-based methods, such as finite element, finite volume, and finite difference methods, the degrees of freedom increase as the size of the elements or cells decrease. Solution verification is principally concerned with discretization error, which is usually the dominant numerical error, and iterative error. Discretization error is estimated by comparing the simulation results obtained at different spatial and temporal resolutions.

Note that code verification and solution verification are two distinct activities with quite different purposes. Code verification requires exact solutions; this allows the examination of the exact numerical error, for the purpose of finding mistakes made while programming the algorithms. Solution verification is used to estimate the numerical error of simulations for which the exact solution is not known.

### **Validation**

Validation is the process of determining that a mathematical model is an accurate representation of the real world for the intended use of the model. The model is made up of the governing

equations (usually partial differential or integro-differential equations), boundary conditions, initial conditions, and any constitutive or closure relations. Validation requires the quantitative comparison of results from numerical simulations to measurements from physical experiments. The goal of validation is an assessment of the mathematical model that the computer code implements, accounting for numerical error, uncertainties in the input quantities, and errors and uncertainties in the experimental results.

There are three distinct phases of determining the adequacy of a model for an intended purpose (Oberkampf and Trucano (2008) and references therein). The first is an accuracy assessment, in which numerical simulations are quantitatively compared to experimental measurements. In this document our focus is on this first phase. The second phase is an extrapolation or interpolation of the numerical model from the conditions of the validation experiments to the conditions of the intended use. In the nuclear power field, the Code Scaling, Applicability, and Uncertainty (CSAU) process addresses this phase. The final phase is the determination of whether or not the code is sufficiently accurate for the conditions of the intended use. It is intended that the V&V and UQ processes recommended by NE-KAMS will be an important resource for the decision maker who must make such determinations.

Validation requires measurements from physical experiments (AIAA (1998), Trucano et al (2006)). Comparing simulations from different classes of models, e.g., comparing direct numerical simulations of fluid turbulence to simulations that incorporate turbulence models, is *not* validation. Such comparisons can be valuable and worthwhile, as can calibrating a simpler model to match results from a more sophisticated model, but these activities do not test either model against a measurement of reality (Trucano et al (2006)).

### **Uncertainty quantification**

Consider a process that maps a set of inputs to a set of outputs, that is, given values for the inputs, the set of output values are to be determined. The process can be a computation, a controlled experiment, or simply observations at some time after the inputs are measured. It is common for the values of the inputs to not be known precisely. The question that immediately follows, and that uncertainty quantification tries to answer, is: given a description of the uncertainty of the inputs, what is the uncertainty of the outputs?

There are two main types of uncertainty. Random variability of a quantity is called *aleatory* uncertainty. On the other hand, if a parameter is believed to have a single value, but this value is not known precisely, the uncertainty is called *epistemic*. Aleatory and epistemic uncertainties are treated differently; aleatory uncertainties are described by probability distributions, while epistemic uncertainties are best described as intervals. When properly accounting for input uncertainties, the results of the process under consideration (the outputs) are described probabilistically.

It should be noted that a wide array of terms are often used in place of epistemic and aleatory, some of which are synonymous, and others that have similar meaning, but have a unique connotation. For instance in experimental measurements, systematic (or bias) uncertainties are commonly epistemic uncertainties, but not necessarily. For example, experimentalists attempt to identify systematic uncertainties in their measurements and then they attempt to remove them. Once identified, the uncertainty is no longer an epistemic uncertainty, whether it is removed from the measurements or not, because it has now been discovered. In addition, experimentalists will use the term random (or precision) uncertainty in place of aleatory uncertainty, but these three terms have essentially the same meaning.

In the context of the assessment of numerical simulation results, inputs can include parameters that define the geometry, the material models, and the boundary and initial conditions. For these types of inputs, uncertainty quantification estimates how much the simulation results may vary when these inputs are not known, are known imprecisely, or have inherently variability. Another type of input includes parameters associated with the numerical method, or possibly a choice between several methods.

Experimental measurements can also be dependent on conditions, parameters, or other inputs that are uncertain. Uncertainty quantification can be applied to experimental processes to estimate the variability of experimental results. Validation can be seen as a comparison of two distributions, one from the numerical simulations and one from the experimental measurements, when uncertainties in inputs to numerical simulations and experimental measurements are propagated.

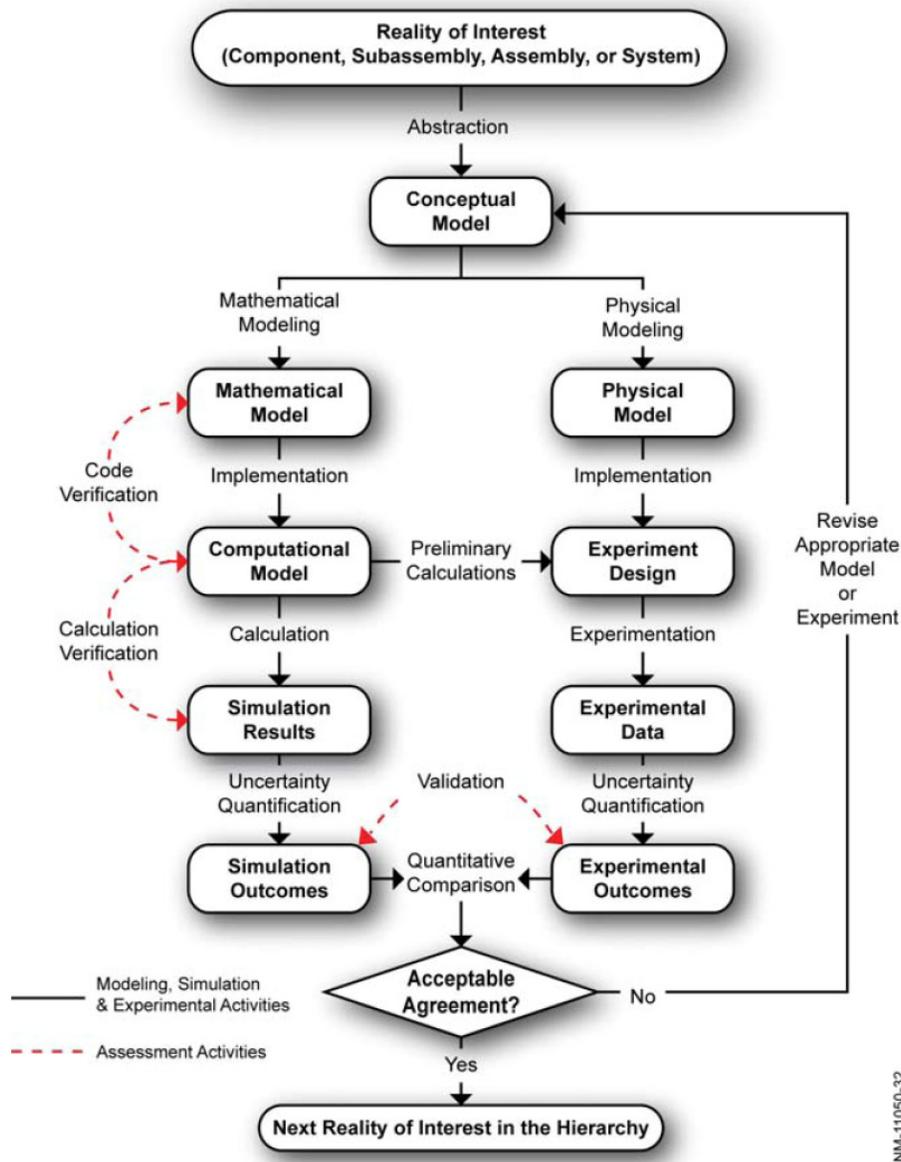
### **Calibration**

Calibration is the process of adjusting input parameters of the model in order to obtain the best match between simulation results and experimental measurements. In a similar vein, model parameters may be adjusted to match results from a more sophisticated model, e.g., adjusting turbulence model parameters to match results from direct numerical simulations (DNS). Depending on the model parameter being calibrated, calibration often restricts the applicability of the model to the domain of the data used to calibrate the parameter.

Calibration is a model development activity, but is often confused or combined with model assessment activities. For example, after calibrating a model parameter to match experimental data, a claim that the code is validated for similar conditions is often (erroneously) made. Another common practice is to refine or coarsen the mesh for a numerical simulation until the results agree with experimental data; this is a calibration of a numerical simulation parameter (the mesh resolution) but is often claimed to be validation or solution verification.

### **2.3 The Relationship Between Different Methods of Assessment**

The various independent assessment techniques described above can be used for different purposes. Figure 1 shows a framework (ASME (2006)) for organizing the different techniques with the goal of assessing numerical simulation results for an intended use. As described below, this framework combines different assessment techniques in a complementary fashion. The approach recognizes that there are experimental errors and uncertainties as well as errors and uncertainties in numerical simulations.



**Figure 1: A framework for validation. (ASME, (2006))**

Validation attempts to quantify the accuracy of the mathematical model through comparisons with experimental measurements. However, a code provides only approximate solutions to the mathematical model, and the approximations made by the numerical method must be accounted for; solution verification is used to estimate this numerical error. Inputs must also be provided to the code to produce a simulation, and these inputs may be uncertain; uncertainty quantification estimates the effects of these input uncertainties on the simulation results. Finally, the experimental measurements also contain errors and uncertainties, and these must be identified and estimated by the experimentalists conducting the experiment. Clearly, any large numerical errors or uncertainties, or large experimental errors or uncertainties, will make it difficult to isolate differences between the mathematical model and the reality it attempts to describe. As shown in Fig. 1, the various assessment techniques are combined so that ultimately, a

probabilistic comparison can be made between the mathematical model that the code seeks to express and the physical model that the experiments attempt to observe.

Code verification is the foundation on the simulation branch of this framework. Without code verification, conclusions drawn from validation, UQ (as a part of validation) and solution verification (as a part of validation) are of dubious value. Code verification provides evidence that the numerical methods are performing as intended; consequently, the results from other assessment techniques can be interpreted as unpolluted by bugs. For example, as the mesh is refined to estimate the numerical error while performing solution verification, all the differences between simulations at different resolutions can be attributed to discretization error; without code verification, there is no evidence to support this attribution.

As applied on the simulation branch of the framework, uncertainty quantification examines the mapping between the inputs to a code and the code's results; this mapping is often *assumed* to be the same for the code as for the mathematical model that the code seeks to approximate. Ideally, code verification provides confidence that the numerical methods are implemented correctly and solution verification demonstrates that numerical errors are small, and this justifies the interpretation of the uncertainty quantification conducted on the code extends to the mathematical model.

Code verification may rely on solution verification to characterize the error of the exact solution. In this role, solution verification is used to estimate the accuracy of the benchmark solution to which the code results will be compared. This use of solution verification is described in Sections 2.4 and Section 3, and in this document, is the main purpose of solution verification; however, in the research literature the solution verification is most often identified with isolating numerical error while conducting validation. Also, note that uncertainty quantification appears on both the simulation and experimental branches of the framework in Fig. 1, and in this document we consider both applications. In the research literature uncertainty quantification is used in other contexts as well.

## 2.4 Code Verification

Code verification is practiced through the quantitative comparison of numerical solutions produced by the code being tested and independently computed exact or highly accurate solutions of the governing equations. As numerical solutions are approximate solutions of the governing equations, differences between numerical solutions and exact solutions are expected. The difference is the numerical error. It is the behavior of this error that code verification examines, with the goals of (a) testing the implementation of the algorithms and numerical methods in the code (i.e., finding any coding mistakes,) and (b) identifying any algorithmic weaknesses.

Numerical analysis is used to show that an algorithm or numerical method has particular properties, e.g., consistency, stability, accuracy, conservation, convergence, symmetry preservation, etc. The properties of an algorithm are in fact statements about the behavior of the numerical error. When a numerical method is developed or modified, it is standard practice to describe the numerical analysis of the method in research literature.

In code verification, one produces numerical solutions using an implementation of a numerical method, and then tests whether expected properties of that numerical method are observed in the

numerical solutions. An implementation of a numerical method cannot be *proven* to be correct; instead, the implementation is examined over a number of tests and properties, and confidence in the implementation is developed by the accumulation of successful test results. Put another way, successful test results provide evidence that the algorithm is implemented correctly.

A key property of numerical methods considered in this document is the convergence rate, broadly defined as the rate at which the error of the numerical solution goes to zero as the degrees of freedom go to infinity. The most common use of code verification is to confirm that a code converges at the design order of accuracy of the algorithm.

The quality of the exact or high-accuracy solution is critical. Successful comparisons against exact or high-accuracy solutions provide strong evidence that the algorithm is implemented and performing correctly. The quality of these solutions is determined by the how clearly the test problem is defined and how accurately the exact solution can be computed. The test problem specification includes all the necessary information to determine the exact solution, e.g., the governing equations, a complete description of any submodels, and the initial and boundary conditions. If any part of this specification is ambiguous or missing, assumptions must be made in the code inputs to run the problem, and differences between the exact solution of the test problem and the numerical solution might be attributable to these assumptions and not solely to numerical error. Similarly, if the exact solution requires, for example, the numerical solution of an ordinary differential equation (ODE), differences with the numerical solution of the code may be partly attributable to the numerical solution of the ODE. In both cases, the evidence that the numerical method is implemented correctly is weakened.

Unfortunately, traditional exact solutions can be difficult or impossible to obtain for many physics regimes. Exact solutions to the governing equations may not be available because of the complexity of the governing equations, boundary conditions, or initial conditions. Exact solutions typically result from simplifying assumptions to the governing equations, boundary conditions, or initial conditions. As a result, simplified exact solutions do not fully test all of the aspects of the code being tested; which is expected to solve a general problem correctly.

When traditional exact solutions are not available, the method of manufactured solutions (MMS) can provide alternatives. (See Roache (1998) and Knupp and Salari (2003).) Manufactured solutions are mathematical devices that can be used to broadly test the coding correctness and implementation of algorithms, and are generally comparable to the highest quality reference solutions. Manufactured solutions require that the code being tested can add and evaluate arbitrary source terms in the governing equations. The flexibility of MMS can also provide access to large regions of the testing domain that traditional exact solutions cannot.

The results of code verification are not always definitive. One possibility is that the conditions under which the property of the algorithm are demonstrated are not met, for example, the numerical simulations are not in the asymptotic regime of convergence. Another is that, the error structure of several algorithms used in a simulation may be convolved in such a way that the individual errors can not be isolated, such as temporal and spatial errors, or errors from algorithms for different physical phenomena. Discontinuous solutions can exist for some models, such as shock waves or multi-material or multi-phase physics; numerical analysis may not provide strong statements for such solutions. Nevertheless, code verification should be pursued in these situations because it identifies weaknesses and barriers in existing numerical methods

and analysis techniques. A great deal can be learned through the systematic, rigorous framework that code verification provides.

## 2.5 Validation

As described above, the combination of the three aspects of validation is used to determine if the mathematical model is an acceptable representation of reality for the intended use of the model. This task is complicated because one does not have direct access to the mathematical model, and one does not precisely know reality. Uncertainty from several sources must be estimated, for simulation results as well as experimental measurements; these estimates connect simulations back to the mathematical model and the experimental measurements back to reality. Ultimately, the comparison of the mathematical model to reality is a comparison of probability distributions, where the probability distributions are the acknowledgement that the mathematical model and reality are not directly accessible.

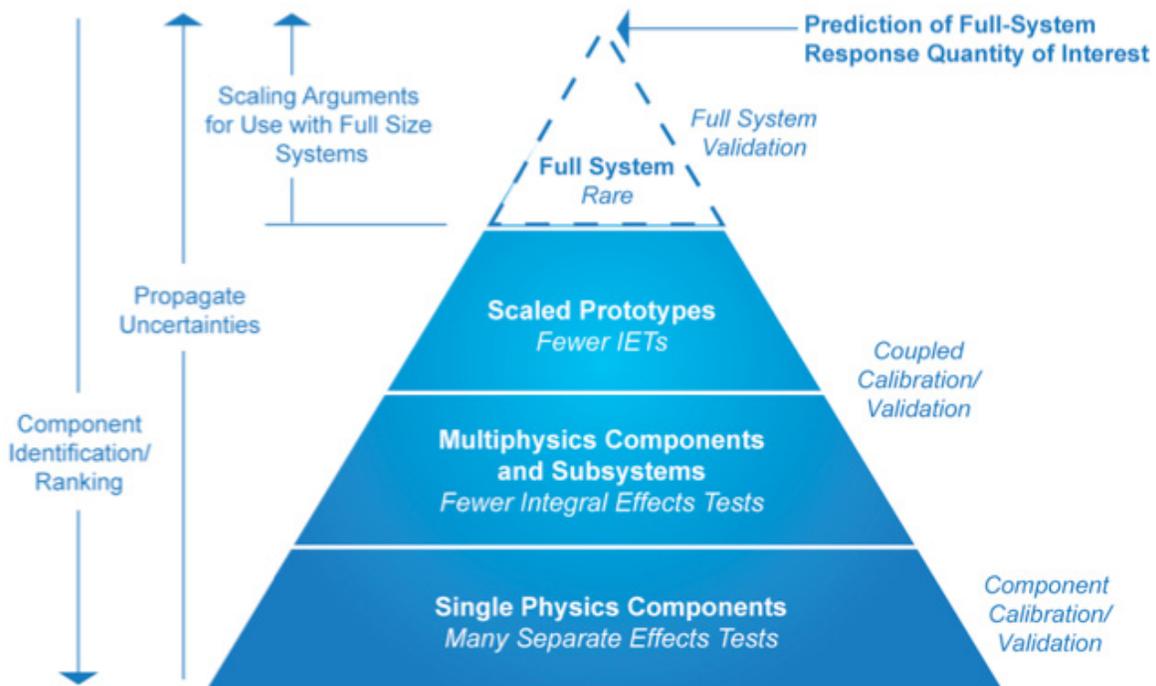
A broad probability density function (PDF) indicates a large uncertainty, and vice versa. While the details of comparing distributions are beyond the scope of this document, the key point is that narrower distributions are needed to make more definitive statements from the comparison, whether positive or negative. So, in order to draw definitive conclusions from validation, (a) the probability distributions from simulation results and experimental results must be made narrow enough, or alternatively, uncertainties must be made small enough, and (b) epistemic (lack of knowledge) uncertainties, either in the simulation or in the experiments, must be minimized.

In Section 4, levels of completeness are defined for the description of various attributes of fluid dynamics experiments and experimental data. The intended effect of achieving higher completeness levels for most of the attributes is to reduce the uncertainties of *simulation* results. In any validation activity, simulations of the experiment must be conducted, and many ambiguities and lack of knowledge deficiencies in the description of the experiment are transferred to the simulations as inputs with large uncertainties. More thorough description of the experimental setup and the characterization of e.g., boundary and initial conditions, clarify the values of the inputs to a simulation.

The completeness levels of other attributes are directed at reducing uncertainties in the experimental results, and concentrate on the characterization of the uncertainties in the data acquisition and processing methods. To further reduce the uncertainties in experimental results, experiments must be designed specifically for this goal. Conceptually, all aspects of an experiment could be very well characterized, but the results might still have large uncertainties – the completeness levels are based on the thoroughness of the description of an experiment, not on the accuracy or precision of the experiment itself. In practice, however, we expect that the experimental benchmarks that achieve high completeness levels will be intended for validation from the outset.

An additional fundamental, as well as practical, aspect of validation in a real engineering environment has been the concept of a validation hierarchy (AIAA (1998); Oberkampf and Trucano (2002); ASME (2006)). A schematic of a validation hierarchy from the CASL program is shown in Fig. 2. Because it is neither feasible nor practical to conduct true validation experiment on the most complex or large-scale systems, the method often employed is to use a building-block approach. This approach decomposes the complex engineering system of interest into three or more progressively simpler levels of complexity (tiers): subsystem cases,

benchmark cases, and unit problems. In the nuclear reactor safety field, a similar concept, referred to as separate effects testing, has been used for a long time and is an integral part of the methods developed for scaling (Zuber (1991)) and quantifying the uncertainty (Boyack, et al. (1990)) in nuclear reactor system thermal hydraulics. The strategy in the tiered approach is to assess how accurately the computational responses compare with experimental responses at multiple levels of physics coupling and geometric complexity. The tiered approach is very useful for several reasons: (1) the hierarchy can represent a large range of complexity in systems, physics, material and geometry; (2) the hierarchy requires a wide range of experienced individuals to construct it, providing opportunities for discovering subsystem or component interactions that had not been previously recognized; (3) the hierarchy supports testing of models, or submodels, at any of the tiers of complexity; and (4) different hierarchies would be constructed for analyzing the system under different environments, *e.g.*, normal, abnormal, and hostile environments. In addition, the tiered approach recognizes that the quantity, accuracy, and cost of information that is obtained from experiments vary radically over the range of tiers.



**Figure 2. Schematic of a validation hierarchy. (CASL (2010))**

At the highest level, there are relatively large uncertainties, and experimental data is difficult and costly to obtain. At the lowest level, physical phenomena are isolated and experiments are less costly to conduct and less difficult to diagnose; consequently uncertainties are expected to be small. By building on these simpler validation tests and moving up the hierarchy, one can identify the point at which uncertainties become too large to make definitive statements about the predictive capability of the code.

### **3. Code Verification Benchmark Data Standards and Requirements**

#### **3.1 Introduction**

The purpose of code verification is to provide unambiguous evidence of the correct implementation of numerical solutions produced by the software. The lack of ambiguity is derived from the quality of the evidence used to establish the correctness. Benchmarks based upon analytical or high quality approximate solutions of the governing equations serve the purpose of the basis of comparison for establishing correctness. The NE-KAMS database provides an appropriately graded approach to establishing the pedigree of test solutions used in code verification. Furthermore, a structure for peer review and assessment of benchmark quality is provided. Taken together, the pedigree and peer review provides a basis for acceptance and confidence in the testing of simulation codes against the benchmarks.

Code verification relies upon solving mathematically well-defined problems. The quality of the benchmark definition and its accurate solution is essential to producing the desired result. The benchmark's quality is a direct determining factor on the quality of the code verification results. A high quality benchmark is a starting point for high quality code verification. The standards described below are designed to provide the structure by which a verification benchmark can be defined with the necessary quality to achieve high quality code verification. The level of rigor and documentation associated with the problem is directly related to the strength of the verification benchmark. For example, a simple closed form benchmark solution is necessarily simpler to describe in its entirety than a discretized solution to a full set of partial differential equations with a computer code similar to the code being verified by the application of the fiducial solution. The NE-KAMS knowledge base is designed to accommodate each approach while recognizing the differences inherent in the benchmark definitions.

#### **3.2 Description of Verification Benchmark Data and Related Artifacts**

Below are listed several types of benchmarks for verification, and types of associated data necessary to include with each. Examples of the benchmark types are included that should be familiar to those working in computational fluid dynamics. Because different types of test problems present different levels of difficulty in solution evaluation and quality, the problems require different supporting data. These data will be used to provide details of the solution and provide users with a level of confidence in the solution. Next, a brief description of types of data is provided so that benchmarks can be fully documented. This is followed by a summary of the expected data to be included for each type of benchmark defined. This section closes with comments regarding benchmark solutions and direct numerical simulations, because the latter are often used as alternatives to true analytical solutions.

##### **Types of code verification benchmarks**

The problem types are categorized by the difficulty of providing a solution coupled with the quality of the solution that can be obtained. These two concepts go hand-in-hand. For example, simple closed form solutions are easy to obtain and evaluate, whereas a numerical solution of partial differential equations is difficult and carries a number of serious issues regarding its quality and trustworthiness. These issues are addressed by an increased level of scrutiny of the evidence provided for the associated data. Not all of the types of benchmarks are analytical in nature, and the solutions for each type may be constructed in different ways with different

expected levels of quality and accompanying data. This necessitates the different levels of required documentation and accompanying supporting material to assure the user of a benchmark's quality and accuracy.

Following is a list of types of code verification benchmarks along with an archetypical example of each. This is intended to be instructive to the experienced reader, who may recognize the example. The list is roughly ordered in increasing level of difficulty and need for greater supporting material.

- Analytical solution with no numerical approximations.
  - Traditional analytical solution: Incompressible, unsteady, 2-D, laminar flow over an oscillating plate (Stokes oscillating plate) given on pp. 266-272 of Panton (1984).
  - Manufactured Solution: Incompressible, steady, 2-D, turbulent, wall-bounded flow for several turbulence models, given in Eça, et al. (2007).
- Analytical solution with significant numerical evaluation.
  - Series solution. Examples include numerous classical problems in Lamb (1932); classical separation of variables solutions to heat conduction problems; and impulsively started incompressible, unsteady, axisymmetric 2-D, laminar flow in a circular tube (Szymanski flow), given on pp. 133-134 of White (1991).
  - Nonlinear algebraic solution: The Riemann shock tube problem, J. Gottleib and Groth (1988).
- Numerical solution of an ordinary differential equation. Example: Incompressible, steady, 2-D, laminar stagnation flow on a flat plate (Hiemenz flow), given on pp. 152-157 of White (1991).
- Numerical benchmark with an accurate numerical solution. Example: Incompressible, steady, 2-D, laminar flow in a driven cavity (with the singularities removed), given in Prabhakar and Reddy (2006).
- Code-to-code comparison data. Example: Incompressible, steady, 2-D, laminar flow over a back-step, given in Gartling (1990).

### **3.3 Completeness Classification of Code Verification Benchmark Data**

Each benchmark will be provided with a variety of metadata that describes the benchmark and related information. This information is related to the description of the problem, its solution and the accuracy of that solution. Each of these artifacts or metadata will be reviewed and this review will assist the user of the NE-KAMS knowledge base in assessing their own use of the test problem. Higher graded tests will encourage the user of NE-KAMS to use a given problem with confidence, while lower grades will provide caution to the user. The intent of the grades will be to inform NE-KAMS knowledge base users, as well as NE-KAMS contributors, regarding the elements of quality associated with code verification submittals.

A table designed for the evaluation of verification benchmarks based on completeness of the description of various attributes is given below (Table 1: Verification Benchmark Completeness Table.) The attributes are supported by the documentation categories discussed in Section 3.4, and the quality of the documentation impacts the assessment of completeness. In Section 3.4, the mapping between the documentation types and the attribute(s) it supports is provided.

<div style="text-align: center;"><b>COMPLETENESS</b></div> <div style="text-align: left;"><b>ATTRIBUTES</b></div>	<b>Completeness Level 0</b>	<b>Completeness Level 1</b>	<b>Completeness Level 2</b>	<b>Completeness Level 3</b>
<b>Conceptual Description</b>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of the physical and modeling aspects tested</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the physical and modeling aspects tested</li> <li>• Some discussion of the mathematical and numerical aspects tested</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed discussion of the mathematical and numerical aspects tested</li> </ul>
<b>Mathematical Description</b>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of the mathematical and solution aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the mathematical and solution aspects</li> <li>• Some description of the existence and uniqueness/non-uniqueness of the solution</li> <li>• Some description of the chaotic and/or random nature of the solution, if applicable</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the existence and uniqueness/non-uniqueness of the solution</li> <li>• Detailed description of the chaotic and/or random nature of the solution, if applicable</li> </ul>
<b>Boundary and/or Initial Conditions</b>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of all boundary and initial conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of discontinuities and/or singularities in the boundary and initial conditions, if applicable</li> <li>• Some description of coordinate singularities, if applicable</li> <li>• Some discussion of discrete implementation details of initial and boundary conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of discontinuities and/or singularities in the boundary and initial conditions, if applicable</li> <li>• Detailed description of coordinate singularities, if applicable</li> <li>• Detailed discussion of discrete implementation details of initial and boundary conditions</li> </ul>
<b>Computer Hardware and System Software</b>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of the hardware and system software used</li> <li>• Some description of the compiler and effect of the options used</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the hardware and system software used</li> <li>• Detailed description of the compiler and the effect of the options used</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of results obtained on alternate hardware and system software</li> <li>• Detailed description of results obtained on alternate compilers</li> </ul>

<p style="text-align: center;"><b>SQA and Code Verification</b></p>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of the software and its history</li> <li>• Some SQA practices are documented and followed</li> <li>• Some code verification conducted and documented</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the software and its history</li> <li>• Detailed SQA practices are documented and followed</li> <li>• Detailed code verification conducted and documented</li> <li>• Observed order of numerical convergence evaluated, if applicable</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of results obtained using alternate symbolic manipulators, if applicable</li> <li>• Observed order of numerical convergence evaluated, including discontinuities, singularities, chaos, and randomness, if applicable</li> </ul>
<p style="text-align: center;"><b>Solution Accuracy Assessment</b></p>	<ul style="list-style-type: none"> <li>• Little or no description of all aspects</li> </ul>	<ul style="list-style-type: none"> <li>• Some description of the effect of iterative error is provided, if applicable</li> <li>• Some description of the effect of discretization error is provided, if applicable</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the effect of iterative error on all quantities of interest is provided, if applicable</li> <li>• Detailed description of the effect of discretization error on all quantities of interest is provided, if applicable</li> <li>• Observed order of numerical convergence estimated on all quantities of interest using multiple solutions, if applicable</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of the effect of numerical algorithm parameters, if applicable</li> <li>• Detailed description of the effect of round-off error</li> <li>• Statistical convergence error estimated, if applicable</li> <li>• Detailed description of the effect of discontinuities, singularities, chaos, and randomness, if applicable</li> </ul>

**Table 1: Verification Benchmark Completeness Table**

### 3.4 Connection to NE-KAMS V&V Database

The database will provide an archive of data either directly associated with the benchmark solution or documentation speaking to the quality of the benchmark solution. These data are “graded” by the completeness table above. In the verification portion of the NE-KAMS database, the types of data described below will be contained as reference material to assist the implementation and evaluation of a benchmark. For example for a test problem of reasonable complexity source code should be available to produce the analytical solution for comparison. In addition, the quality of the solution provided will be available so that the assessment can be fully produced. Further evidence should be available regarding the software quality measures used to assure the continued quality of the source code as develop and testing continues. Furthermore, the test problems setup with initial and boundary conditions are contained within the database.

Each of these data items, both individually and as a whole for a benchmark, will be subjected to a peer review and grading using the verification benchmark as a rubric. This review will be fully available to the users of the database allowing the users to justify their use and reliance upon the test problem for the conduct of code verification.

#### Types of code verification data

Below is a list of the different types of data associated with verification problems defined above. Depending on the nature of the test problem only a subset of these data are necessary. This will be provided below the list of data types. As noted above, benchmarks with well-defined closed form analytical solutions require relatively less data than a benchmark associated with the approximate numerical solution of PDEs.

- Detailed technical description of the problem (report or paper)
- Analysis of the mathematics of the problem (report or paper)
- Computer analysis of solution (input file)
- Computer solution of the mathematical solution
- Computer implementation of the numerical solution
- Error analysis of the “exact” numerical solution
- Derivation of the source term and software implementation or input
- Computer implementation of the source term (manufactured solution)
- Grids for numerical solution
- Convergence and error estimation of approximate numerical solution
- Uncertainty and sensitivity study of numerical solution
- Description and analysis of computational methods
- Numerical analysis theory associated with convergence
- Code description/manuals
- Input files for problems and auxiliary software
- Patch test description, Derivation, input and analysis
- Unusual boundary conditions (inflow, piston, etc.)
- Physics restrictions (boundary layer theory, inviscid,)
- Software quality documents
- Scripts and auxiliary software for verification

- Source code
- Metric descriptions
- Verification results including code version, date, etc.
- Numerical sensitivity studies
- Feature coverage in verification

Below, the characteristics of each type of benchmark documentation (could be called artifacts or metadata) associated with a code verification benchmarks to be contained in the NE-KAMS database are briefly described. These metadata take a number of concrete forms such as a written document, computer code, mathematical solution in document or software form, input files for executable codes, input to automatic computer analysis, output from software quality systems, among others. Each type of documentation will be identified directly with an attribute from Table 1 that it supports. Each form of documentation will be assessed against the criteria listed in Table 1 for completeness. Ultimately, the database itself will contain direct examples of each of these data with completeness evaluations.

- Detailed technical description of the benchmark (report or paper): This can include a technical paper in a journal or conference proceeding describing the benchmark and its solution. Another form would be a formal or informal report from an institution providing the same information. This supports the “Conceptual Description” attribute in Table 1. Peer review by the database is important to assure that the documentation is sufficient.
- Analysis of the mathematics (report or paper): For any solution that is closed form, or requiring a semi-analytical solution, the mathematics must be described in detail. This can be included in the paper (report) discussed previously or in a separate document. This supports the “Mathematical Description” and “Boundary and/or Initial Conditions” attributes in Table 1. Again, the document should be peer-reviewed for completeness and accuracy.
- Computer analysis of solution (input file): If the mathematics or solution is accomplished using a computerized analysis, the program used and the input to the program should be included. Some sort of written documentation such as a manual for the software ideally accompanies this artifact. This supports the “Mathematical Description,” “Boundary and/or Initial Conditions” attributes and “SQA and Code Verification” attributes in Table 1.
- Computer solution of the mathematical solution: The actual computerized solution of the mathematical problem should be included in whatever form the computerized solution takes. This should include any error analysis completed with this solution. This supports the “Mathematical Description” and “SQA and Code Verification” attributes in Table 1.
- Computer implementation of the numerical solution: The analytical solution should be implemented in a computational form to allow the comparison with the numerical solution. This should include some sort of error analysis in the form of a report. This supports the “Mathematical Description” and “SQA and Code Verification” attributes in Table 1.
- Derivation of the source term and software implementation or input: In the case of the method of manufactured solutions, the source term used to drive the numerical method must be derived through a well-defined numerical procedure. This should be

documented, along with the numerical tools used for the derivation and implementation. This supports the “Mathematical Description,” “Boundary and/or Initial Conditions,” and “SQA and Code Verification” attributes in Table 1.

- Computer implementation of the source term (manufactured solution): The source term should be included in a form amenable to direct use in a computer code. The language for the computer code should be clearly defined as well as the compiler and computer system used. This supports the “Mathematical Description,” “Boundary and/or Initial Conditions,” “Computer Hardware and System Software” and “SQA and Code Verification” attributes in Table 1.
- Grids for numerical solution: If a solution is computed using another simulation code, all relevant details on the numerical grid(s) used must be included. This could be direct grid files, or input files to well-defined grid generation software. This supports the “Solution Accuracy Assessment” attribute in Table 1.
- Convergence and error estimation of numerical solution: The numerical solution must include a convergence study and error estimate. These should be detailed in an appropriately peer-reviewed document. This supports the “Solution Accuracy Assessment” attribute in Table 1.
- Uncertainty and sensitivity study of numerical solution: The various modeling options in the code used to provide the numerical solution must be examined vis-a-vis the uncertainty and sensitivity of the solution to these choices. This study should be used to justify the methodology used for the baseline solution. This supports the “Solution Accuracy Assessment” attribute in Table 1.
- Description and analysis of computational methods: The methods used by the code used for the baseline solution must be completely described and analyzed. This can take the form of a complete bibliography of readily available literature. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Numerical analysis theory associated with convergence: The nature of the convergence and the magnitude of error in the numerical solution must be described and demonstrated. This can take the form of a complete bibliography of readily available literature. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Code description/manuals: The code manual and complete description must be included with the analysis and description. This supports the “SQA and Code Verification,” “Computer Hardware and System Software” and “Solution Accuracy Assessment” attributes in Table 1.
- Input files for benchmarks and auxiliary software: The input file used to produce the solution must be included. Any auxiliary software used to produce or analyze the solution must be full described or included. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Unusual boundary conditions (inflow, piston, outflow, Robin, symmetry, etc.): Should the benchmark require unusual or involved boundary or initial conditions, these must be described in additional detail, including the nature of implementation. This supports the “Mathematical Description” and “Boundary and/or Initial Conditions” attributes in Table 1.

- Physics restrictions (boundary layer theory, inviscid, parabolized Navier-Stokes, ...): If the solution requires the solution of a reduced or restricted set of equations, this must be fully described. Examples are boundary layer theory, truly inviscid flow, or various asymptotic limits. This supports the “Mathematical Description” and “Boundary and/or Initial Conditions” attributes in Table 1.
- Software quality documents: Of non-commercial software used to produce solutions, the software quality pedigree should be clearly established by documenting the software quality and steps taken to assure the maintenance of the quality. This supports the “SQA and Code Verification” attribute in Table 1.
- Scripts and auxiliary software for verification: Auxiliary software or scripts used to determine the verification or compute error estimates for a software used to produce solution should be included. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Source code: If possible the actual source code for the software along with instructions for producing an executable (makefile, scripts) should be included with all other documentation. This supports the “SQA and Code Verification” attribute in Table 1.
- A full mathematical or computational description of metrics used in error analysis and evaluation of solution implementation or numerical solution. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Verification results including code version, date, and other identifying characteristics: The verification basis for the code used to produce the baseline solution must be included. This includes any documentation of verification, peer-review, code version, date completed and error estimates. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.
- Feature coverage in verification: The code features covered by verification benchmarks must be documented. Any gaps where the feature used for the baseline solution is not verified must be explicitly documented. This supports the “SQA and Code Verification” and “Solution Accuracy Assessment” attributes in Table 1.

### **Data requirements for each type of benchmark**

Below are the necessary data requirements for each category of benchmark, again arranged in order of increasing level of documentation required. For completeness each data type would be expected to be present in the database to describe a benchmark of a given type.

- Common elements for all types of benchmarks (it is notable that the use of proper verification using an analytical solution results in the most compact set of requirements for data, manufactured solutions also).
  1. Paper or report
  2. Mathematical analysis
  3. Computerized solution and input
  4. Error and uncertainty analysis
  5. Computer implementation of the evaluation of the solution
  6. Restrictions
  7. Boundary or initial conditions

- Closed form analytical solution
  1. Paper or report
  2. Mathematical analysis
  3. Computerized solution and input
  4. Error and uncertainty analysis
  5. Computer implementation of the evaluation of the solution
  6. Restrictions
  7. Boundary or initial conditions
  
- Manufactured Solution
  1. Paper or report
  2. Mathematical analysis
  3. Computational solution and input
  4. Error and uncertainty analysis
  5. Computer implementation of the evaluation of the solution
  6. Derivation and implementation of the source term
  7. Restrictions
  8. Boundary or initial conditions
  
- Analytical solution with significantly complex evaluation (e.g., series solution, nonlinear algebraic solution, nonlinear ODE solution)
  1. Paper or report
  2. Mathematical analysis
  3. Computerized solution and input
  4. Error and uncertainty analysis
  5. Computer implementation of the evaluation of the solution
  6. Input files
  7. Source code
  8. Source code SQA
  9. Method description and manual
  10. Restrictions
  11. Boundary or initial conditions
  
- Highly accurate numerical solution (not analytical), numerical benchmarks or code-to-code comparisons.
  1. Paper or report
  2. Mathematical analysis
  3. Computational solution and input
  4. Error and uncertainty analysis for the solution
  5. Computer implementation of the evaluation of the solution
  6. Input files
  7. Grids
  8. Source code
  9. Source code SQA
  10. Method description and manual
  11. Method analysis
  12. Method verification analysis and coverage

13. Restrictions

14. Boundary or initial conditions

### **Comments on code-to-code comparison and direct numerical simulation**

The use of direct numerical simulation (DNS) requires a similar or even higher level of documentation than analytical solutions. This coincides with the discussion of the last type of verification benchmark where a complex numerical method with significant approximations is utilized to produce the solution. As a numerically computed benchmark, the burden of proof is much larger. Code verification is best served by exact analytical solutions because of the relative ease in assuring benchmark solution accuracy. Nonetheless, it remains a common practice due to its inherent simplicity. It also appeals to those who have a vested interest in the solutions produced by a certain computer code. The credibility of the comparison is predicated on the credibility of the code producing the “benchmark” used as the surrogate for truth. Therefore the documentation of the benchmark must provide the basis for the credibility.

The use of DNS as a surrogate for experimental data has received significant attention. This practice violates the fundamental definition of validation adopted herein because no observation of the physical world is used to define the data. This practice also raises other difficulties. First the DNS code itself requires a rigorous code verification assessment. Second that verification activity must be further augmented by a validation assessment for its application. This includes all the activities that would define a validation study including experimental uncertainty analysis and numerical and physical equation based error analysis. When the DNS serves to provide validation data, the DNS contains approximation errors that must be estimated as part of the “error bars” for the validation data. Furthermore, the code must have documented credibility beyond the details of the calculation used as validation data. This level of documentation again takes the form of the last form of verification benchmark introduced above because of the nature of DNS codes. For this reason DNS is included as a member of this family of benchmarks.

## 4. Validation Benchmark Data Standards and Requirements

### 4.1 Context

A validation experiment should be designed to capture the essential physics of interest, and measure all information required by the mathematical model to simulate the physics, including initial and boundary conditions, material properties, and system excitation. Validation experiments are fundamentally different than traditional experiments. (See Oberkampf and Roy (2010) for a detailed discussion of validation experiments.) A validation experiment is designed for the purpose of determining the ability of a model to simulate a well-characterized physical process. In traditional experiments, the emphasis is typically on discovery or improved understanding of physical phenomena, calibration of existing models, or on assessment of system performance, reliability, or safety. In validation experiments, the emphasis is on detailed characterization of the experiment so that the predictive accuracy of the mathematical model can be critically assessed. In a validation experiment, control and repeatability of the experiment are important, but what is more important is precisely measuring the all of the information that is needed as input to the model. In addition, measurements should be made to determine if any boundary or initial condition assumptions are made that might be inappropriate in attempting to simulate the experiment. For example, 2-D inflow may be assumed in the modeling, but in a validation experiment the experimentalist should carefully measure the inflow to determine if the assumption is appropriate. Suppose that the experimental inflow has significant 3-D aspects and, as a result, the simulation is inaccurate. Then the modeler may be able to use the 3-D inflow data, or at least relax the 2-D assumption, and obtain good agreement with the experiment.

Typical experiments are not documented sufficiently for use in validation, either because a desire for a concise description of the experiment (as is desirable in a journal article), or because the required information was never acquired. When detailed documentation of experimental conditions is not available, a situation can occur that is very detrimental to model validation. If needed input data are not measured and documented by the experimentalist, a computational analyst frequently chooses to adjust these unknown parameters or conditions so that the best agreement is obtained between their results and the measurements. This practice, obviously, degrades the ability to assess the predictive accuracy of a model.

The comparison of the computational results with experiment should occur after both code and solution verification are complete. If these are not adequately completed, one can obtain misleading or incorrect interpretations of the model's predictive accuracy. For example, one could have (a) a situation where a computer coding error cancels a modeling error resulting in good agreement between simulation and experiment, or (b) a situation where a simulation result for a particular spatial mesh resolution agrees with experiment, but finer mesh resolution results disagree with measurements.

Comparisons of computational predictions and experimentally measured responses are typically made by graphing the results. Although useful, this type of comparison has limited quantitative information for either decision makers who want to use simulations, or for predictive uncertainty techniques that use quantitative estimates of model form uncertainty. Mathematical techniques have been recently constructed for estimating model form uncertainty and these are usually referred to as *validation metrics* (see, for example, Oberkampf and Trucano (2002), Oberkampf

and Barone (2006), Ferson, Oberkampf, and Ginzburg (2008)). Validation metrics are quantitative measures of disagreement between computational results and experimental measurements of system response quantities (SRQs) of interest. They can be thought of as difference operators where the input to the operator is computational and experimental SRQs. Each input can be a scalar, a function, or probability distribution, such as a cumulative distribution function. The output of the metric can be a scalar, a function, or a probability distribution. This report does not recommend or prescribe how computational and experimental results should be compared. That decision is best left to user of the validation database and the needs of the project using the database.

## 4.2 Description of Validation Benchmark Data and Metadata

The validation benchmark table is constructed in a manner similar to the code verification benchmark table discussed in Section 3.2. The validation table is based on various completion levels of six attributes of the experiment:

- Experimental facility
- Analogy instrumentation and signal processing
- Boundary and initial conditions
- Fluid and material properties of the walls
- Test conditions
- Measurement of experimental responses

These attributes were chosen because they focus on three features: (a) information needed as input to the model, (b) measurement of experimental responses predicted by the model, and (c) how measurements, either input or responses, were obtained and processed by the experimentalist.

One of the key technical challenges of forming a validation benchmark table is that there is no a priori knowledge of what level of physical modeling fidelity may be used to simulate the experiments in the knowledge base. Specifically, as the CFD analyst employs higher levels of physics modeling fidelity, the analyst requires an increasing variety of information. For instance, a simulation using the  $k-\varepsilon$  turbulence model requires a measure of the turbulent kinetic energy at the inflow plane. The more sophisticated Reynolds stress models would require that all Reynolds stresses be measured at the inflow plane. A Large Eddy Simulation (LES) may require that the inflow data be provided of the turbulence spectrum. Lower levels of modeling fidelity, such as Reynolds-Averaged Navier-Stokes models, cannot use the higher levels of completeness information.

As was done in the code verification benchmark table, there are four levels of information completeness for each of the six attributes. As one increase the completeness information on an attribute, one increases the completeness level for that attribute. For example, the lowest level, Level 0, refers to a limited or non-existent description of the attribute. At the highest level, Level 3, the experimentalist must measure and characterize features of physics in the experiment and the facility that would be needed by the highest levels of physics modeling fidelity that can be foreseen for the prediction of the physical processes in the experiment. To understand the need for extremely high levels of physics characterization, the experimentalist must remove himself or herself from the traditional mindset of believing that they have a priori knowledge of what types of modeling will be used to predict the experimental measurements.

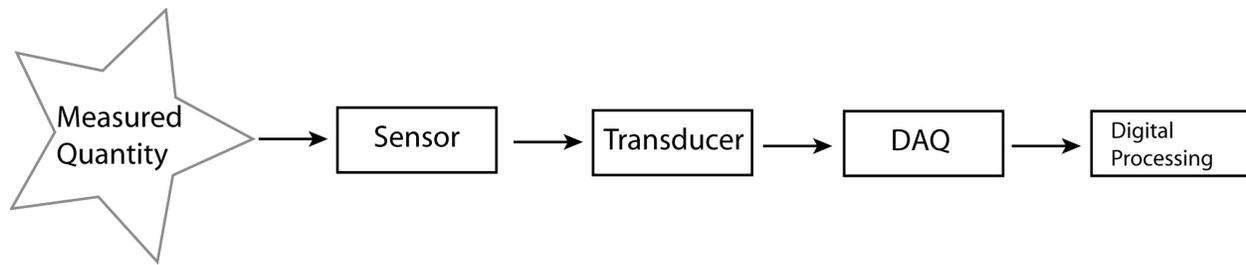
It is important to note that the level of completeness is not an assessment of the quality of the experiment. A low completeness level does not imply a poorly performed experiment, but rather one for which limited information is reported. It is rather a gage of how much information has been provided that *might be used* for very high levels of physics fidelity in the future. Data sets that merit high levels of completeness will be inherently more expensive to generate than data sets with lower completeness levels. In many cases, the extra information provided in a high completion level data set may be completely unnecessary for the validation of the model of interest. The completion level should be used as a guide when balancing the costs to conduct a validation experiment against the goals of the validation exercise directed toward some real-world application of interest. It should also be stressed that each higher level of completion requires higher levels of uncertainty estimates on all reported quantities.

The concept of SSBs was formulated in Oberkampf and Trucano (2002). SSBs should be of a high-quality such that they can be viewed as engineering reference standard. SSBs are test problems that have the following four characteristics: (1) the purpose of the benchmark is clearly understood, (2) the definition and description of the benchmark is precisely stated, (3) specific requirements are stated for how comparisons are to be made with the results of the benchmark, and (4) acceptance criteria for comparison with the benchmark are defined. This report is an initial attempt at constructing SSBs for both verification and validation benchmarks in computational fluid dynamics.

### **4.3 Fundamentals of Sensors and Transducers**

In some cases, the information requested in the validation benchmark completion table is the same information used by the experimentalist to generate other information that is also requested. For example, information on the calibration of the instrumentation and information on the signal conditioning may be requested in one element of the table, but this same information is also used by the experimentalist to report the measured quantities of interest and their estimated uncertainty. As a result, it may appear that some of the requested information is redundant. However, reporting this information will allow users of the database to make more detailed and accurate assessments of the data than simply examining the measured quantities of interest.

Proper interpretation of a measurement requires understanding of the methods used to obtain the measurement. Nearly any measurement involves several steps. A physical phenomenon can be detected, but typically, a sensor's response must be converted to an electrical signal in order to be useful. This signal is often amplified, filtered, or modified in other ways before being converted to a digital quantity. As a result, it is important to understand our use of a few fundamental terms. As shown in Figure 3 below, a measurement of a quantity is performed using a sensor, which converts its output to an analog voltage through a transducer. This voltage, which is normally converted to a digital signal, is recorded using a data acquisition system (DAS). After acquisition, data is typically processed further digitally, whether that be averaging, integration, differentiation, or computation of one variable from another (e.g. computing the pressure field from velocity data).



**Figure 3. Processing is often needed to interpret measurements.**

#### 4.4 Completeness Classification of Validation Benchmark Data

This section discusses the various types of information that are needed for each of the six attributes of the validation benchmark completion table, Table 2. Most examples of information required in the table are relevant to a water or wind tunnel experiment, since these are the most common experimental facilities in fluid dynamics. In the future, we expect to add other examples as we receive input from experimentalists and computational analysts with broader interests. Additional examples of how to achieve the higher levels are discussed for each attribute in Appendix B.

With regard to all of the attributes discussed below, *each element* of information provided for the completeness table should include an *estimate of the measurement uncertainty*. This may seem excessive to most experimentalists, but it should be clear that modern CFD simulations and validation activities are increasing the emphasis on uncertainty quantification. It is common among experimentalists to only think about reporting uncertainties on system response quantities (SRQs) of interest, i.e., measured quantities that would be compared with the predicted, or output, quantities of the CFD simulation. As will be seen in the discussion of the completeness table, five of the six attributes of information describing the experiment deal with *input data* to the CFD simulation. Uncertainties on the simulation input quantities allow the modeler to perform sensitivity studies with these quantities. The modeler may then choose to conduct a nondeterministic simulation using the most important uncertainty input quantities in order to obtain a more properly characterized uncertainty prediction of SRQs. These predicted uncertain SRQs could then be compared more quantitatively with the uncertain measurements from the experiment.

##### Experimental Facility

The required information dealing with the experimental facility is concerned with the functional operation of the facility and its operating procedures. That is, this attribute gathers information on how a facility works, how flow conditions are controlled, and how it was operated for the experiment that is being submitted for inclusion in the knowledge base. As one would expect for a water or wind tunnel facility, the information required would include calibration measurements of the flow inside the test section of the facility, including such characteristics as fine-scale flow features, turbulence spectrum and its distribution over the test section, and acoustic environment. These types of measurements should be obtained inside the test section without a model installed so that the CFD analyst would be able to concentrate on calibrating features in the CFD model, without dealing with the complexity of the model installed in the test section.

As the physical modeling fidelity of the CFD analysis is improved, modelers will begin to model the flow in larger portions of the facility; not just the flow around a model in a test section. As discussed in Oberkampf and Roy (2010), some CFD analyses include the simulation of the flow

in the entire test section of the wind tunnel facility, including the model installed in the test section. In the future, we expect higher fidelity simulations to include computing the flow upstream and downstream of the test section; potentially even the simulation of the entire flow loop of a closed circuit facility. For example, in a water or wind tunnel, it deals with the geometric and equipment features of the facility and how the flow is driven and controlled. This would be information such as; whether the facility was a closed circuit or in-draft facility, what type of pump or blower is used to drive the flow, flow control hardware and its operation, feedback control system for flow control, and any human control of the facility. For a heat transfer experiment, this attribute requires details on how heat is generated in the operation of the facility, information on heat transfer through the walls of the facility, and how the temperature of the flow is controlled.

### **Analog Instrumentation and Signal Processing**

By analog instrumentation and signal processing, we refer to all of the important sensors in question, all of the analog equipment, as well as analog and digital signal processing. In many cases, signal conditioning is performed in the same "package" as the sensor, which does not obviate the need for a description of the signal conditioning, especially if filtering of the analog output signal is performed.

The information provided for this attribute will be used to interpret the quality and accuracy of the experimental data. Without the requested information about the instrumentation and signal processing, it may be impossible to understand, or the CFD analyst may misinterpret the measurement results reported. Two examples of this situation are (a) lack of information dealing with the time correlation of two signals measured with different sensors, and (b) analog or digital time averaging of a sensor output in the experiment, but the CFD analyst does not apply an analogous time filter to the simulation. An additional reason for requiring this information is that subsequent to the NE-KAMS assessment of a dataset, new information may come to light about the temporal or spatial resolution of a certain measurement technique. This new information may alter the interpretation of the performance of the measurement technique.

In the case of optical measurements, lenses and optical filters fall under signal processing. This information may be used to better understand subtle aspects of the data that the may go unnoticed to the experimentalist and uncertainty sources that the experimentalist may be unaware of. For instance, filtering of an analog signal causes the signal's phase to shift. The optics in a PIV measurement directly impact the measurement uncertainty in a manner that is a current research topic.

Any and all digital processing of the data must also be reported. This could range from subtraction of an offset in a pressure measurement, to time averaging, to signal processing in the frequency domain. In order to achieve high completion levels, each digital operation should be described in detail. Ideally, the code used to perform the digital processing is also provided.

In some instances, considerable digital processing occurs, such as PIV measurements. Once a vector field is computed, the velocity data are "post processed" to remove "bad vectors" that can be caused either by bad experimental technique or by factors over which one has no control. Bad vectors are present in any PIV measurement and must be removed since they have no relationship to the actual velocity at that location. Bad vectors can be detected a number of ways, including, but not limited to, checking to see if they fit within an expected range of values, requiring that each vector looks similar to its neighbors, and by requiring that the correlation that produced the vector, which is akin to its signal strength, is sufficiently large. Some data

processing codes search for a reasonable result that fits well with its neighbors and report this value even when it does not represent the strongest signal. In a case where no reasonable answer can be found, some codes replace the missing data with an average of the neighboring values. Such practices can lead to large and unquantifiable errors in the mean and statistics of the data. All details of the scheme used to choose the reported vector and to replace missing values should be provided.

### **Boundary and Initial Conditions**

Obtaining boundary and initial condition data is critically important because these data are directly used as input to the CFD simulation of the experiment. If portions of this information are missing, a computational analyst may adjust assumptions about these conditions until a desirable result is obtained, resulting in a model calibration activity, as opposed to model validation.

If the CFD simulation defines the computational domain as the test section of the water or wind tunnel, then the inflow and wall boundary conditions must be provided by the experimentalist and reported as part of the experiment. What information is needed by the CFD analyst for the outflow boundary conditions is problematic. If the streamwise component of the flow over the entire outflow plane is downstream, then the typical outflow boundary condition imposed by the analyst is to set the streamwise derivative of all dependent variables in the PDEs to zero. To justify this modeling assumption, the experimentalist would be required to measure (at least some) of the flow quantities at the exit of the test section. If reverse flow exists at the exit of the test section, i.e., flow is in the upstream direction, then the information required from the experimentalist is significantly increased. For example, the experimentalist may (a) need to better characterize the flow at the exit of the test section, or (b) provide detailed geometric information of the mount structure holding the model in position (and possibly flow measurements) downstream of the test section so that the analyst can move the outflow boundary aft of the test section.

Since higher fidelity simulations require more input information on a wider variety of variables, higher completeness levels will require more information than lower levels in the table. The highest level of completeness in the table, i.e., Level 3, will include all boundary and inflow information that could reasonably be used in high-fidelity-physics simulation.

### **Fluid and Material Properties of the Walls**

Properties of the fluid are required inputs to any fluid simulation model. For a simulation that includes the energy equation, one also requires the dependence of all fluid properties on temperature and pressure. To provide this information, the composition of the working fluid must be precisely reported. For instance, for the simple case of air, it is necessary to report the fluid properties at the stagnation condition upstream of the test section. This would include, for example, pressure, temperature, density, relative humidity, and possibly air composition, if an unusual composition is expected. For water as the working fluid, information would be required for the amount of dissolved air, chlorine, ph, and possibly any small bubbles or particles in the water.

Concerning information on the test section walls, high-fidelity physics models can require detailed information about the material properties and the surface characteristics. For example, if the CFD simulation models the flow in the entire test section, then the experimentalist must measure this type of information. Material joints, gaps, or mismatches in wall sections should be reported. For high levels of completeness in the table, information should be reported on wall roughness not only in the test section, but also upstream of the test section. More sophisticated

modeling approaches may also require information about the distribution and shape of the roughness elements, including details on fasteners that are exposed to the flow, such as bolt or screw heads. In addition, some models may require information affecting heat transfer through the walls, such as radiative properties and thermal conductivity, including spatial variation of these quantities.

### **Test Conditions**

Information required for Test Conditions is similar to the attribute Boundary and Initial Conditions, but here the emphasis is on reporting detailed temporal variation of conditions during an experiment, or series of experiments. For example, in a closed-circuit water tunnel, control of water temperature and bubble concentration during a long-running data-gathering experiment can be difficult. High levels of completeness for Test Conditions would require reporting the temporal variation in these quantities during a run, as well as reporting these for each of the runs during which experimental measurements were obtained.

A similar example deals with reporting the test conditions for an indraft wind tunnel, a natural convection experiment inside a large laboratory, or an experiment conducted in the open atmosphere. For a long-running experiment, the room conditions can change significantly during the experiment either due to changes in atmospheric conditions or recirculation of heated air. In addition, there could be changes in conditions at the exit of a wind tunnel, such as one exhausting outdoors and the outside wind conditions affect the test conditions of the experiment. As a result, high completion levels require that the ambient conditions be recorded and reported. Lower completion levels require that the band over which the ambient conditions vary be reported.

### **Measurement of Experimental Responses**

A validation data set should include measurements of all system response quantities (SRQs) of interest that could be compared to the output of the numerical simulation. The thoroughness of comparing predicted and measured SRQs has two aspects: spatial coverage of SRQs compared and variety of SRQs compared. Spatial coverage refers to the range of spatial dimensionality over which an SRQ is measured and the ability to compare this result to the CFD simulation. The range can vary from measurements (a) at a point in space, (b) along a line, for example, on a line spanning  $360^\circ$  along the surface of a sphere, (c) in a 2-D plane in space, for example, in a streamwise plane near a model geometry of interest, and (d) in a 3-D volume in space, for example a volume encompassing a geometry of interest. In general, it becomes more difficult experimentally to obtain these increasing quantities of measurements. Some experimental techniques, however, are able to inherently obtain measurements over a plane in space, such as particle image velocimetry (PIV).

The variety of SRQs refers to the wide range of dependent variables that can be used in the formulation of the PDEs in the mathematical model, as well as derivatives, functionals, and combinations of these quantities. The ability to measure and predict this wide range of SRQs is staggering. One method of ordering the general difficulty of measuring and predicting this range of quantities was given by Oberkampf and Roy (2010). They constructed a spectrum of SRQs based on whether the quantity is a derivative or an integral of a dependant variable in the PDEs. For example, an integral of pressure and fluid velocity, such as lift and draft on an object, is (in general) easier to predict and measure than the pressure and velocity of the fluid in the vicinity of the object. Similarly, predicting and measuring fluid velocity is easier than predicting and measuring the derivative of fluid velocity components, such as local heat flux and shear stress.

Although this concept of a spectrum has limited usefulness, it does provide some guidance on what measured quantities will provide higher levels of completion assessment in the table.

The completeness level must also be judged on what measurement science considers as currently feasible. In some cases, two SRQs could have similar prediction difficulty, while one is more important to the intended application than the other. In such cases, the application of interest will be used as a guide for judging completion level.

### **Completeness Levels**

As with the code verification table, the validation table is constructed such that the columns are the various levels of completeness of each attribute. To attain a given level of completeness for a given attribute, not only must the requirements listed in the associated element of the table be completed, but also all of the requirements listed to the left of the given element. Stated differently, the completion requirements are cumulative from left-to-right, as one attains higher levels of completeness in the table.

<b>COMPLETENESS</b> <b>ATTRIBUTES</b>	<b>Completeness Level 0</b>	<b>Completeness Level 1</b>	<b>Completeness Level 2</b>	<b>Completeness Level 3</b>
<b>1. Experimental Facility</b>	<ul style="list-style-type: none"> <li>• Little or no description of the facility or its operation</li> </ul>	<ul style="list-style-type: none"> <li>• Some information on the functional operation of the facility and its operating procedures</li> <li>• Some information on the geometric and equipment features of the facility</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed information on the functional operation of the facility and its operating procedures</li> <li>• Detailed information of the geometric and equipment features of the facility</li> <li>• Some information on the calibration procedures and reference standards for the facility</li> <li>• Some information on the calibration results and characterization of the facility</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed information of the fine-scale flow features/environment inside the test section</li> <li>• Some information of the fine-scale flow features or physical processes upstream and downstream of the test section</li> <li>• Detailed information on the calibration procedures and reference standards for the facility</li> <li>• Detailed information on the calibration results and characterization of the facility</li> <li>• Information on the inspection, maintenance, and repairs of the facility</li> </ul>
<b>2. Analog Instrumentation and Signal Processing</b>	<ul style="list-style-type: none"> <li>• Little or no information on sensors and calibration procedures</li> <li>• Little or no information on instrumentation</li> <li>• Little or no information on signal processing</li> </ul>	<ul style="list-style-type: none"> <li>• Some information on sensors and calibration procedures</li> <li>• Some information on transducers</li> <li>• Some information on signal processing</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed information on sensors and calibration procedures</li> <li>• Detailed information on transducers</li> <li>• Detailed information on signal processing</li> <li>• Some assessment of instrument performance and suitability</li> </ul>	<ul style="list-style-type: none"> <li>• Use of independent sensors and calibration procedures</li> <li>• Use of independent/alternative signal processing procedures</li> <li>• Detailed assessment of instrument performance and suitability</li> </ul>
<b>3. Boundary and Initial Conditions</b>	<ul style="list-style-type: none"> <li>• Little or no information on boundary conditions</li> <li>• Little or no information on initial conditions</li> <li>• Little or no information on</li> </ul>	<ul style="list-style-type: none"> <li>• Some inflow quantities measured</li> <li>• Some wall quantities measured</li> <li>• Some initial conditions measured</li> <li>• Detailed model-design dimensions provided</li> </ul>	<ul style="list-style-type: none"> <li>• Most inflow quantities measured</li> <li>• Most wall quantities measured</li> <li>• Most initial conditions measured</li> <li>• Detailed as-built model dimensions measured</li> <li>• Some outflow and reverse flow quantities measured</li> </ul>	<ul style="list-style-type: none"> <li>• Fine-scale inflow quantities measured</li> <li>• Fine-scale wall quantities measured</li> <li>• Fine-scale outflow quantities measured</li> <li>• Fine-scale initial conditions measured</li> </ul>

	design dimensions			<ul style="list-style-type: none"> <li>• As-tested model dimensions measured</li> <li>• Inflow and outflow quantities measured at multiple streamwise locations</li> </ul>
<b>4. Fluid and Material Properties of the Walls</b>	<ul style="list-style-type: none"> <li>• Little or no information on fluid and material properties</li> <li>• Little or no information on wall properties</li> </ul>	<ul style="list-style-type: none"> <li>• Some thermodynamic state data of fluid(s) provided</li> <li>• Some transport properties of fluid(s) provided</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed information provided to determine thermodynamic state of fluid(s)</li> <li>• Detailed information provided to determine transport properties of fluid(s)</li> <li>• Volume fraction of additional phases is provided, plus size distribution statistics</li> </ul>	<ul style="list-style-type: none"> <li>• All thermodynamic, transport, and optical properties of the fluid(s) are provided, as well as how these are determined</li> <li>• Thermal, mechanical and optical properties of the wall(s) are provided</li> <li>• Detailed description of additional phases is provided, plus size distribution statistics</li> </ul>
<b>5. Test Conditions</b>	<ul style="list-style-type: none"> <li>• Little or no information on test conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Some description provided of the method of control and record of the test conditions</li> <li>• Some description provided for measuring test conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description provided of the method of control and record of the test conditions</li> <li>• Detailed description provided for measuring test conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed description of operational procedures for setting and controlling test conditions</li> <li>• Detailed measurement of time and spatial variation of test conditions</li> </ul>
<b>6. Measurement of Experimental Responses</b>	<ul style="list-style-type: none"> <li>• Little or no information on data acquisition and sampling procedures</li> <li>• Only mean data are provided.</li> <li>• Only data from a single run are provided</li> </ul>	<ul style="list-style-type: none"> <li>• Some information on data acquisition and sampling procedures</li> <li>• Statistical analysis of final experimental responses provided</li> </ul>	<ul style="list-style-type: none"> <li>• Detailed information on data acquisition and sampling procedures</li> <li>• Multiple experiments conducted to determine measurement uncertainty</li> <li>• Statistical analysis of intermediate experimental responses provided</li> </ul>	<ul style="list-style-type: none"> <li>• Use of independent data acquisition procedures</li> <li>• Description of sensitivity of experimental responses to control of test conditions</li> <li>• Video recording of measurement procedures and data acquisition provided</li> <li>• All experimental responses reported with estimated bias and random uncertainties, including correlated uncertainties</li> </ul>

**Table 2: Validation Benchmark Completeness Table**

## **5. NE-KAMS Support for Nuclear Reactor Design, Safety Analysis and Licensing**

From its inception, NE-KAMS will directly support nuclear energy research, development and demonstration programs within the U.S. DOE, including the CASL, NEAMS, LWRS, SMR, and NGNP programs. These programs all involve computational M&S of nuclear reactor systems, components and processes. The NE-KAMS effort will initially focus on supporting the use of CFD and T/H analysis for M&S of nuclear reactor systems, components and processes, and will later expand to include materials, fuel system performance and other areas of M&S as time and funding allow.

### **5.1 Computational Fluid Dynamics for Nuclear Reactor Safety Analysis**

The use of reactor system and containment codes for performing safety analyses of reactor systems has been established for more than 30 years. Very reliable codes have been developed for analyzing the pipework and components of the primary system, and results from these analyses are often used in the safety assessment of nuclear power systems undertaken by the regulatory authorities. Such codes are based on networks of 1-D or even 0-D cells. The application of CFD methods to problems relating to nuclear reactor safety analysis is less well developed but is rapidly accelerating. The need to use CFD arises because the flow in such components as the upper and lower plena, downcomer and core of a reactor pressure vessel is strongly three dimensional. Natural circulation, mixing and stratification in containments is also essentially 3-D in nature, and representing such complex flows by pseudo 1-D approximations may not just be oversimplified but could even be misleading, resulting in erroneous judgments being made.

The use of numerical simulation methods in nuclear reactor safety often has to address regulatory concerns. From this perspective, a common approach to dealing with practical licensing issues is to use simplified modeling, coupled with a conservative approach, to ensure that adequate safety margins are guaranteed. Traditionally, a large number of sensitivity studies are carried out to determine how plant parameters have to be modified in order for the predictions to remain conservative. Sophisticated statistical methods, such as Latin Hypercube Sampling (LHS), have placed this practice on a firm mathematical foundation. However, a key concern is to determine the degree of conservatism needed to compensate for the lack of physics embodied in the simplified models. Information can be obtained from mock-up experiments, but difficult scaling issues have to be faced in order to ensure that the extrapolation of model data to full scale is trustworthy (Yadigaroglu and Zeller (1994)). Moreover, the experiments themselves inevitably involve simplifications, and judging the degree of conservatism associated with introducing the simplifications is itself quite difficult. The only way to ultimately ensure that a conservative approach has been proposed is to increase safety margins, but this practice often places unwelcome constraints on plant efficiency and competitiveness.

The trend being taken by most safety authorities is to gradually replace a conservative approach by a best estimate methodology, coupled with an uncertainty evaluation (D'Auria (2009)). This policy change has already taken place in the context of system analysis with the development of second-generation codes in the 1970s, based on the two-fluid approach as a means of replacing the conservatism of simplified two-phase flow models. The use of CFD codes in nuclear reactor

safety analysis may be viewed similarly in regard to having an appropriate numerical tool to analyze certain situations in which there is a strong 3-D flow component.

To gain acceptance in the licensing world, however, CFD analyses performed need to be underpinned by a comprehensive V&V program to demonstrate the capability of the technology and to provide results reliable enough to be used in licensing procedures. For single-phase applications, CFD is mature enough to complement existing analysis tools currently employed by the regulatory authorities, and it has the potential to reduce conservatism without compromising safety margins. One thing is for certain, however, CFD will enter the safety picture in an increasing way in the near future.

The objective of the NE-KAMS effort is to provide the V&V underpinning required so that CFD can be used with confidence in nuclear reactor safety analysis. To that end, NE-KAMS will provide V&V benchmark databases with strong sense benchmarks for CFD code verification and validation for use in V&V assessments to improve the accuracy and reliability of physics models and CFD codes for nuclear reactor design, safety analysis and licensing applications. In particular, these V&V benchmarks should be of high quality such that they can be viewed as engineering reference standards. Although CFD code verification benchmarks and validation benchmarks have been constructed for a number of years, the comprehensive approach embodied in NE-KAMS with its quality-assessed V&V benchmark databases, V&V standards, requirements, and best practice guidelines, and related support services for the construction and use of SSBs in V&V has not been proposed.

Initially, the NE-KAMS effort will focus on supporting single-phase CFD applications and then will work towards supporting two-phase CFD applications. One of the reasons why the application of CFD methods in nuclear reactor safety has been slow to establish itself is that the transient, often two-phase, phenomena associated with accident events are extremely complex. Traditional approaches using system codes have been successful because a very large database of phasic exchange correlations has been built into them. The correlations have been formulated from 1-D separate-effects experiments and have been well tested. Data on the exchange of mass, momentum, and energy between phases for 3-D flows are very sparse in comparison. Thus, although 1-D formulations may restrict the use of system codes in simulations in which there is geometric complexity and 3-D fluid motions, the physical models are well-established and reliable, provided they are used within their specified ranges of validity, and these days they are often run in real time for full reactor transients. In contrast, to use CFD, the physical models will require considerable further development, especially for two-phase applications.

## **5.2 Nuclear Reactor Safety Problems for Which CFD Could Bring Real Benefits**

A comprehensive listing of nuclear reactor safety problems for which CFD could bring real benefits in terms of better understanding, quantification, and improved safety estimation is provided in the Organization for Economic Cooperation and Development/Nuclear Energy Agency (OECD/NEA) Technical Report, NEA/CSNI/R92207)13 (Smith (2008)). Full details are given in that report, but brief descriptions of some of these nuclear reactor safety problems are provided below.

## **Boron Dilution**

Mechanisms have been identified (Queral and Gonzalez (2004)), such as Small Break Loss of Coolant Accident (SBLOCA) or steam generator tube rupture (SGTR), which could lead to a slug of low boric acid water being injected through one of the coolant loops into the reactor pressure vessel of a Pressurized Water Reactor (PWR). If the slug arrives at the core without mixing significantly with the streams from the other cold legs, a (local) criticality excursion could ensue. The complete phenomenological model requires two steps: (i) knowledge of the concentration of boron at the core entrance, and (ii) thermal-hydraulics/neutronics calculations for the core region. The first step (covered by state-of-the-art CFD) thus provides the initial and boundary conditions for the second. Main CFD inputs to this problem concern the description of the transportation mechanisms to the core: namely, pump start-up or natural circulation after restoration of the water inventory. Relevant parts of the reactor for flow modeling concern at least the downcomer, the lower plenum, and possibly the pipework related to the initial transportation of the slug to the reactor pressure vessel. One-dimensional system codes are not able to simulate these processes realistically; CFD analysis is needed, due to the multi-dimensional, transient nature of the flow, the geometrical complexity of the computational domain, and the requirement of accurately representing the mixing of the different flow streams.

## **Mixing and Thermal Fatigue**

Thermal stratification, cycling and striping phenomena may develop in the major components and piping systems of nuclear plants. The phenomena can occur in safety related lines, such as the pressurizer surge line, the emergency core cooling injection lines, and other lines where hot and cold fluids come into contact with each other. Damage resulting from the associated fluctuating thermal loads to nearby structures has been reported for mixing tees of the feedwater systems, in the reactor cleanup systems, and in residual-heat removal systems. Sometimes, defective valves through which hot (or cold) water leaks into a cold (or hot) water stream are the cause of such fatigue problems. The problem is a serious safety concern in respect to ageing and life management of nuclear plants. Coolant temperature oscillations due to turbulent, thermal mixing effects that pose a risk of wall thermal fatigue are reported to be at frequencies up to several Hz (Wakamatsu et al. (1995)). Significantly higher frequencies than these, however, are considered not to pose a risk, as they are strongly attenuated by the thermal inertia of the pipe material.

In general, the common thermal fatigue issues are well understood and can often be controlled or circumvented (by adding internal static mixers, for example). However, some incidents indicate that certain information on the loading in the mixing zone, and its impact on the structural material, is still missing. One such incident occurred at the Civaux-1 PWR in 1998 (Jungclaus et al. (1998); Shah et al. (1999)), which was worrying in that the plant (1450 MW N4) had only been in operation for 1500 hours, and three similar plants had also been constructed.

## **Hot-Leg Heterogeneities**

For the safe running and control of a PWR, it is essential to have, as precisely as possible, knowledge of the real primary flow rates to ensure that they do not exceed the limiting design-basis values. The upper value is derived from mechanical considerations regarding the assembly holding forces and the control rod falling time while the lower value is associated with the DNB (Departure from Nucleate Boiling) risk protection signal. In the plant, the actual primary flow rates are not measured directly but estimated indirectly from internal temperature measurements and overall heat balances.

By far the main source of uncertainty in this procedure (about 10 times greater than from other sources) is related to estimating the average hot-leg temperature. Despite the mixing processes taking place in the upper plenum, important temperature and flow heterogeneities may still be present at the hot-leg instrumentation location, leading to uncertainties in the estimation of the actual average temperature and by inference the actual coolant flow rate. In order to quantify the error, the average temperature of the hot-leg has to be estimated using scale-model tests, from specific plant data, or from CFD calculations.

Direct extrapolation of experimental results to real plant conditions is very difficult (Yadigaroglu and Zeller (1994)) and can often result in an overestimation of the uncertainty. The use of such overestimated values in the case of actual plant situations (e.g. core loading) can give results which do not satisfy the specified safety criteria. Advanced methodologies based on CFD calculations can reduce the level of uncertainty.

### **BWR/ABWR Lower Plenum Flow**

There are many pipes in the lower plenum of a Boiling Water Reactor (BWR) or Advanced Boiling Water Reactor (ABWR) reactor. Two phenomena are relevant to nuclear reactor safety. One is the stress induced by flow vibration, which may cause these pipes to fret and perhaps break, and the other is a lack of uniformity of flow between the pipes, which may lead to a non-uniform temperature distribution in the reactor core.

Many internal structures are located close together in the lower plenum. At a time of partial pump operation, which is an accepted mode of operation in an ABWR, inverse flow can occur in the leg attached to the inactive pump. CFD codes are effective in evaluating the flow field in such geometrically complex situations, and significant progress has been made in Japan in support of their ABWR program (Shiina et al. (2003)). Note that since the study relates to the lower plenum of the reactor, it is a single phase application.

### **Pipe Break**

Transient pressure forces occur on structures following a large pipe break and are of importance for various reactor types. Inside the reactor pressure vessel, the decompression waves produce dynamic loadings on the surfaces of the vessel internals, such as the core shroud and core grids of a BWR. This issue is an important example of the need to predict accurately the transient three-dimensional pressure fields in order to estimate the resulting dynamic loads on structures. It is also important to realize that modern structural analysis has to include dynamic loads, even for Loss-of-Coolant-Accidents (LOCAs).

The decompression process is a highly 3-D and transient phenomenon, so it can only be realistically simulated using CFD. During the first phase, before flashing of the water in the reactor pressure vessel begins, a single-phase CFD model can be used, but after flashing has started a two-phase model is necessary to describe the decompression process. From the beginning of the flashing of the water, the two-phase phenomena are dominant.

### **Hydrogen Distribution in Containments**

During the course of a severe accident in a Light Water Reactor (LWR), large quantities of hydrogen could accumulate in the containment. Detailed knowledge of the containment thermal hydraulics is necessary to ensure the effectiveness of the hydrogen mitigation procedures.

Condensation and evaporation on walls, pool surfaces and condensers need to be modeled realistically because the related mass and heat transfer processes strongly influence the subsequent pressure and mixture composition in the containment. In addition, there is the issue of pressure loading to the structures. The mixture composition is very important because it strongly affects the burning mode of the hydrogen and the operation of the PARs (Passive Autocatalytic Recombiners).

Containments have very large volumes and have multiple compartments. A too-coarse nodalization will not only lose resolution but will smear the temperature, species concentration and velocity fields through numerical diffusion. From a physical point of view, the flow model must also take into account condensation (in the bulk, or on the surfaces of cold walls) with the associated heat transfer to the structures. Unfortunately, condensation models are not yet standardized in CFD codes.

### **Chemical Reactions/Combustion/Detonation**

Detonation and combustion in containments may lead to pressure rises that exceed the design specifications. There is also the risk of localized overheating of structures in the case of standing flames. Deflagrations, accelerated flames, or even detonations, can all be envisaged for some accident scenarios.

Deflagration is a very complex phenomenon, involving both chemistry and turbulence. CFD, combined with flame speed-based deflagration models, can provide insights into the dynamic loadings on the structures. In contrast, detonation processes are relatively simple to model because the very fast front propagation means there is little feed-back from other slower processes, such as chemistry, fluid flow and structural deformation. However, simulation of shock-wave propagation should also account for multiple reflections and superposition of the waves. In principle, CFD has the capability to follow these phenomena.

### **Aerosol Deposition in Containments**

Following a severe reactor accident, fission products would be released into the containment in the form of aerosols. If there were a subsequent leak in the containment barrier, these aerosols would be released into the environment and pose a health hazard. The most conservative assumption is that all the fission-product aerosols would eventually reach the environment. A more realistic assessment can be made by studying the detailed processes which govern the initial core degradation, fission product release, aerosol-borne transport and retention in the coolant circuitry, and aerosol dynamics and chemical behavior in the containment. CFD can be used to model these processes by employing Lagrangian tracking of the aerosols.

### **Atmospheric Dispersion**

Again, following a severe reactor accident, radioactive release to the atmosphere could ultimately occur, which may represent a health hazard for the installation workers and the surrounding population. Atmospheric release of nuclear materials (in the form of both aerosols and gases) implies air contamination, at first on-site and later off-site. The atmospheric dispersion of such material in complex situations, such as buildings being in close proximity to the reactor, is a difficult problem to analyze, but one that is important for the safety of the people living and working in such areas. Dispersion models, which are used to estimate the levels of radioactivity, require meteorological data as input. Typical examples of such data are atmospheric velocity fields and temperature distributions.

Atmospheric motion and dispersion are 3-D in character, turbulent and unsteady; CFD is the traditional approach in the investigation of such flows. On-site simulations must take account of the proximity of nearby buildings and effects due to the wind and weather (i.e. radiation heating by the sun, precipitation, etc.). Off-site, account has to be taken of the topography of the landscape, night-and-day effects, stratified layers, etc. However, within the realm of the physical phenomena, the major challenge lies in turbulence modeling. The flows are highly 3-D, unsteady and are accompanied by strong streamline curvature, separation, and the generation of vortices. The redeeming feature is that atmospheric dispersion is not unique to releases from nuclear power plants, and much progress has been made in other disciplines, in particular in the release of toxic chemicals from non-nuclear industrial plants. As a consequence, little attention has been paid to the issue by the nuclear community except in integral form. The use of best-estimate methodology, such as CFD, would improve the reality of the predictions should more localized information be required.

### **Flow-Induced Vibrations**

Flow-induced vibrations in steam generators (SGs) have been studied for many years, since tube rupture could have serious consequences due to loss of coolant and because of the risk of direct release of radioactive material to the environment. Tube vibration caused by dynamic forces (as generated in the U-bends of standard PWR SGs) may initiate mechanical damage due to fretting, wear, and fatigue. Similar concerns are being expressed in the context of the radial reflectors of Advanced Pressurized Water Reactors (APWRs), since excessive vibration could result in rupture of the fuel-pin cladding, the first barrier against release of radio-active material.

If the core barrel of the APWR is set into vibration by the turbulent flows in the downcomer, the vibrations would be transmitted to the radial reflector through the water filling the space between them. If the radial reflector vibrates, the grid supporting the outermost fuel bundles may make contact with it, and when the grid vibrates, the fuel cladding could wear through fretting. In order to estimate the level of vibration of the radial reflector with sufficient accuracy, it is necessary to calculate the pressure fluctuations generated by the turbulent fluctuations in the downcomer correctly, since these are the driving forces for the vibration. The following two methods are available for using CFD to evaluate the vibration between fluid and structure:

- The vibration between the fluid and the structure is calculated directly by the coupled use of a CFD code and a structural analysis code, perhaps with the fluid structure interaction described by a Lagrangian moving-mesh technique;
- The vibration between the fluid and the structure is calculated by the structural analysis code directly, modeling the water between the core barrel and the radial reflector simply as an additional mass, and imposing the downcomer pressure fluctuations calculated by the CFD code as load conditions.

The latter method is by far the more practical, but the emergence of corporate links between the structural dynamics FEM (Finite Element Method) code ANSYS (<http://www.ansys.com/>) and the established industrial (Finite Volume) CFD codes FLUENT (<http://www.fluent.com/>) and CFX (<http://www.ansys.com/Products/cfx.asp>) brings the prospect of fully coupled simulations for addressing such problems significantly closer. In addition, the FEM code ABAQUS ([http://www.simulia.com/products/abaqus\\_cfd.html](http://www.simulia.com/products/abaqus_cfd.html)) has just released details of a fully coupled CFD module.

## 6. Concluding Remarks

V&V are the primary means to assess the accuracy and reliability of computational simulations. V&V methods and procedures have fundamentally improved the credibility of simulations in several high-consequence fields, such as nuclear reactor safety, underground nuclear waste storage, and nuclear weapon safety. However, significant improvements in the methodology and practice of V&V are necessary to achieve improved credibility in computational sciences and engineering. One element of needed improvements is the design, construction, and use of highly demanding V&V benchmarks to improve the accuracy and reliability of physics models and codes. In particular, these V&V benchmarks should be of high quality such that they can be viewed as engineering reference standards. As such, one of the key components of the NE-KAMS is its V&V benchmark databases with strong sense benchmarks for code verification and validation. The goal of the NE-KAMS V&V benchmark databases is to improve the reliability of the computer software, the estimation of numerical accuracy, the quality of the physics models used, the quantification of uncertainty, and the training and expertise of users of the codes.

Unlike most V&V and assessment databases which only serve as repositories for V&V cases and related documents, the NE-KAMS knowledge base will provide value-added V&V, UQ and computational methods development technical support services to its users. This will be accomplished by (a) establishing standards, requirements and best practices for V&V and UQ assessments, (b) establishing standards and procedures for qualifying and classifying V&V (numerical and experimental) benchmark data, (c) providing readily accessible databases of highly demanding, strong sense benchmark data for use in V&V assessments, (d) providing a searchable knowledge base of information, documents and data on V&V and UQ, and (e) providing web-enabled tools and utilities for V&V and UQ activities.

Code verification benchmarks and validation benchmarks have been constructed for a number of years in every field of computational simulation. However, the comprehensive approach embodied in NE-KAMS with its quality-assessed V&V benchmark databases, V&V standards, requirements, and best practices, and related support services for the construction and use of strong sense benchmarks has not been proposed. The initial focus of the NE-KAMS effort will be on supporting the use of CFD for modeling and simulation of nuclear reactor systems, components and processes. To that end, standards, requirements and best practices for CFD V&V will be established in support of the application of CFD to nuclear reactor design and safety analysis. It is expected that V&V and UQ approaches developed for CFD will be later applied to other computational methods and problems involving CFD, such as coupling of CFD with reactor systems codes, fluid-structure interaction problems and coupling of CFD to neutronics codes. It is envisioned that over time the NE-KAMS effort will expand to include other areas and disciplines relevant to nuclear reactor design and safety analysis, such as materials and fuel system performance.

## 7. References

AIAA, "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," AIAA G-077-1998, 1998.

ASME, "Guide for Verification and Validation in Computational Solid Mechanics," ASME V&V 10-2006, 2006.

Boyack, B. E. et al., "Quantifying Reactor Safety Margins," papers 1 to 6, Nuc. Eng. Des. 119, 1990.

D'Auria, F., "Overview of Uncertainty Methods in Nuclear Reactor Thermal-Hydraulics", Invited Paper N13P1446, Proc. 13th Int. Top. Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-13), Kanazawa City, Japan, Sept. 27 – Oct. 2, 2009.

Eça, L., M. Hoekstra, M., Hay, A., and Pelletier, D., "On the construction of manufactured solutions for one and two-equation eddy-viscosity models." Int. J. Num. Meth. Fluids 54(2), 119-154, 2007.

Gartling, D. K., "A Test Problem for Outflow Boundary Conditions-Flow Over a Backward-Facing Step." Int. J. Num. Meth. Fluids 11, 953-967, 1990.

Gottlieb, J. and Groth, C., "Assessment of Riemann Solvers for Unsteady One-Dimensional Inviscid Flows of Perfect Gases," J. Comp. Phys., 78(2), 437-458, 1988.

Jungclaus, D., et al., "Common IPSN/GRS Safety Assessment of Primary Coolant Unisolable Leak Incidents caused by Stress Cycling", NEA/CSNI Specialist Meeting on Experience with Thermal Fatigue in LWR Piping caused by Mixing and Stratification, OECD/NEA, Paris France, June 7-12, 1998.

Knupp, P. and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, Chapman & Hall/CRC, 2003.

Lamb, H., *Hydrodynamics*, Dover, 1932.

Oberkampf, W. L. and Barone, M. F., "Measures of Agreement Between Computation and Experiment: Validation Metrics," J. Comp. Phys. 217(1), 5-36, 2006.

Oberkampf, W. L. and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, 2010.

Oberkampf, W. L. and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," Progress in Aerospace Sciences 38(3), 209-272, 2002.

Oberkampf, W. L. and Trucano, T. G., "Verification and Validation Benchmarks," Nuc. Eng. Des. 238, 716–743, 2008.

Oberkampf, W. L., Trucano, T. G., Hirsch, C., “Verification, Validation, and Predictive Capability in Computational Engineering and Physics,” *App. Mech. Rev.* 57(5), 345–384, 2004.

Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, 1998.

Panton, R. L., *Incompressible Flow*, New York, John Wiley, 1984.

Prabhakar, V. and Reddy, J. N., "Spectral/hp Penalty Least-Squares Finite Element Formulation for the Steady Incompressible Navier-Stokes Equations." *J. Comp. Phys.* 215(1), 274-297, 2006.

Queral, C., Gonzalez, I., “Analysis of Heterogeneous Boron Dilution Sequences”, Proc. Int. Conf: Nuclear Energy for New Europe 2004, Portoroz, Slovenia, Sept. 6-9, 2004.

Shah V.N., et al., “Assessment of Field Experience Related to Pressurized Water Reactor Primary System Leaks”, Proc. ASME PVP Conf., Boston, USA, Aug. 1-5, 1999.

Shiina, K. et al., “Evaluation of Flow Characteristics in the Lower Plenum of the ABWR by using CFD Analysis”, Paper 36393, Proc. 11th Int. Conf. on Nuclear Engineering (ICONE-11), Tokyo, Japan, April 20-23, 2003.

Smith, B.L. (ed.), “Assessment of Computational Fluid Dynamics (CFD) for Nuclear Reactor Safety Problems”, OECD Nuclear Energy Agency, Technical Report, NEA/CSNI/R(2007)13, Jan. 2008.

Trucano, T. G., Swiler, L. P., Igusa, T., Oberkampf, W. L. and Pilch, M., “Calibration, Validation, and Sensitivity Analysis: What’s What,” *Reliab. Eng. Syst. Saf.* 91, 1331-1357, 2006.

Wakamatsu, M., Nei, H., Hashiguchi, K., “Attenuation of Temperature Fluctuations in Thermal Striping”, *J. Nucl. Sci. Tech.* 32(8), 752-762 (1995).

White, F. M., *Viscous Fluid Flow*, New York, McGraw Hill, 1991.

Yadigaroglu, G., Zeller, G.M., “Fluid-to-fluid scaling for a gravity- and flashing-driven natural circulation loop”, *Nucl. Eng. Des.* 151, 49–64, 1994.

Zuber, N., “A Hierarchical, Two-Tiered Scaling Analysis,” U.S. NRC, NUREG/CR-5809, Washington, DC., 1991.

## 8. APPENDIX A: Examples of Required Information for Each Level of the Verification Benchmark Completeness Table

### 8.1 Conceptual Description

#### Level 1 Examples:

- *Some description of the physical and modeling aspects tested:* A general description of the physical and modeling aspects associated with the benchmark problem.

#### Level 2 Examples:

- *Detailed description of the physical and modeling aspects tested:* A detailed description of the physical and modeling aspects associated with the benchmark problem (e.g. including notions of the applicability of the problem in more applied settings, and how modeling assumptions impact the problem and its solution).
- *Some discussion of the mathematical and numerical aspects tested:* Provide a general description of the mathematical aspects of the benchmark solution, and the issues associated with the numerical solutions and aspects of the benchmark solution examined.

#### Level 3 Examples:

- *Detailed discussion of the mathematical and numerical aspects tested:* Provide a general description of the mathematical aspects of the benchmark solution (e.g., regularity, existence and any side conditions on the solution), and the issues associated with the numerical solutions and aspects tested to the test problem (e.g. expected order of convergence, method aspects drawn out by the solution, etc.).

### 8.2 Mathematical Description

#### Level 1 Examples:

- *Some description of the mathematical and solution aspects:* Provide a basic mathematical description of the benchmark problem solution.

#### Level 2 Examples:

- *Some description of the existence and uniqueness/non-uniqueness of the solution:* Provide a basic discussion of the conditions for solution existence and uniqueness (basic statements on the nature of the solution with regard to these properties).
- *Detailed description of the mathematical and solution aspects:* Provide a detailed mathematical description of the benchmark problem solution and its mathematical properties.
- *Some description of the chaotic and/or random nature of the solution, if it exists:* As a portion of the mathematical solution discuss any chaotic, or random aspects of the solution including physical instabilities (e.g. Kelvin-Helmholtz instability).

#### Level 3 Examples:

- *Detailed description of the existence and uniqueness/non-uniqueness of the solution:* Provide a detailed discussion of the conditions for solution existence and uniqueness (including rigorous proofs of the solutions existence and uniqueness).

- *Detailed description of the chaotic and/or random nature of the solution, if applicable:* Have a detailed discussion of the any chaotic, or random aspects of the mathematical solution including physical instabilities (e.g. Kelvin-Helmholtz). Provide evidence of the character and statistical nature of these solutions.

### 8.3 Boundary and/or Initial Conditions

#### Level 1 Examples:

- *Detailed description of all boundary and initial conditions:* Provide a complete description of all boundary and initial conditions in terms of both a mathematical and numerical details.

#### Level 2 Examples:

- *Some description of discontinuities and/or singularities in the boundary and initial conditions, if applicable:* Provide a description of discontinuities and/or singularities in the boundary and initial conditions as applicable to the problem. For example boundary conditions at corners of rectangular domains, or discontinuous changes in inflow boundaries.
- *Some description of coordinate singularities, if applicable:* Discuss the presence of any coordinate singularities if appropriate for the given problem.
- *Some discussion of discrete implementation details of initial and boundary conditions:* Provide a discussion of the details of the discrete (numerical) implementation of initial and boundary conditions.

#### Level 3 Examples:

- *Detailed description of discontinuities and/or singularities in the boundary and initial conditions, if applicable:* Provide a detailed description of discontinuities and/or singularities in the boundary and initial conditions as applicable to the problem. For example boundary conditions at corners of rectangular domains, or discontinuous changes in inflow boundaries. For example discuss the impact of these features on the mathematical and numerical solutions.
- *Detailed description of coordinate singularities, if applicable:* Discuss the presence of any coordinate singularities if appropriate for the given problem and the impact of the singularity on the mathematical and numerical solution.
- *Detailed discussion of discrete implementation details of initial and boundary conditions:* Provide a detailed discussion of the details of the discrete (numerical) implementation of initial and boundary conditions, for example are the initial conditions point or integral quantities, and how are the integrals evaluated. Discuss whether boundary conditions are in the strong or weak form of the solution. Provide alternative implementations and discuss solution implications.

### 8.4 Computer Hardware and System Software

#### Level 1 Examples:

- *Some description of the hardware and system software used:* A basic description of the hardware and system software used to implement the benchmark solution.

- *Some description of the compiler and effect of the options used:* Provide a description of the compiler (or equivalent software) used plus options applied to produce the executable for the reported solution.

### **Level 2 Examples**

- *Detailed description of the hardware and system software used:* A detailed description of the hardware and system software used to implement the benchmark solution. For example include the entire system characterization, and system version and lineage for the software.
- *Detailed description of the compiler and the effect of the options used:* Provide a detailed description of the compiler (or equivalent software) and options used by the compiler used to produce the executable for the reported solution. For example include the version number and command line used for compilation.

### **Level 3 Examples**

- *Detailed description of results obtained on alternate hardware and system software:* A detailed description of the main and alternative hardware and system software used to implement the benchmark solution. For example include the entire system characterizations, and system versions and lineage for the software.
- *Detailed description of results obtained on alternate compilers:* Provide a detailed description of the primary and alternative compilers (or equivalent software) and options used by the compiler used to produce the executable for the reported solution. For example include the version number and command line used for compilation.

## **8.5 Software Quality Assurance (SQA) and Code Verification**

### **Level 1 Examples:**

- *Some description of the software and its history:* Provide a basic description of the software used and its lineage.
- *Some SQA practices are documented and followed:* Discuss any SQA practices and processes followed in the development of the benchmark solution software.
- *Some code verification conducted and documented:* Provide a description of any code verification utilized in the development process and supporting documentation.

### **Level 2 Examples:**

- *Detailed description of the software and its history:* Provide a detailed description of the software used and its lineage. For example, provide a record of the software's revision history from the software repository.
- *Detailed SQA practices are documented and followed:* Discuss the SQA practices and processes followed in the development of the benchmark solution software. For example, discuss the basic model followed during the development, review procedures, testing, and planning. Include documentation associated with each aspect.
- *Detailed code verification conducted and documented:* Provide a detailed description of any code verification utilized in the development process and supporting documentation. The discussion should provide the degree of code or feature coverage in the verification, and all problems should be documented.

- *Observed order of numerical convergence evaluated, if applicable:* If the problem contains any approximation or truncation (e.g., truncated series, or numerical solution) provide a documentation of the degree of numerical convergence or error in the solution.

### **Level 3 Examples:**

- *Detailed description of results obtained using alternate symbolic manipulators, if applicable:* If the results are obtained through the use of symbolic manipulation provide complementary results through the analysis of another symbolic manipulation program. This step is particularly applicable to the method of manufactured solutions.
- *Observed order of numerical convergence evaluated, including discontinuities, singularities, chaos, and randomness, if applicable:* If the problem contains any approximation or truncation (e.g., truncated series, or numerical solution) provide a complete documentation and estimation of numerical convergence and error in the solution. If the problem contains any features such as discontinuities or chaos this is described and assessed in the convergence study.

## **8.6 Solution Accuracy Assessment**

### **Level 1 Examples:**

- *Some description of the effect of iterative error is provided, if applicable:* If the solution is determined by an iterative solver (linear algebra or nonlinear systems), described the level of iterative error.
- *Some description of the effect of discretization error is provided, if applicable:* If the solution is determined by a discretized operator provide a description of this error's impact on the solution.

### **Level 2 Examples:**

- *Detailed description of the effect of iterative error on all quantities of interest is provided, if applicable:* If the solution is determined by an iterative solver (linear algebra or nonlinear systems), described the level of iterative error and how it impacts all quantities of interest in the benchmark solution.
- *Detailed description of the effect of discretization error on all quantities of interest is provided, if applicable:* If the solution is determined by a discretized operator provide a detailed description of this error's impact on all quantities of interest in the solution.
- *Observed order of numerical convergence estimated on all quantities of interest using multiple solutions, if applicable:* Provide the numerical convergence of all quantities of interest using multiple solution approximations to the benchmark problem (assuming these are appropriate).

### **Level 3 Examples:**

- *Detailed description of the effect of numerical algorithm parameters, if applicable:* Provide a detailed description of all numerical algorithm parameters used such as artificial viscosity, solver tolerances, numerical order of accuracy, linear or nonlinear solution method.
- *Detailed description of the effect of round-off error:* Provide a detailed description of the impact of round-off error on the solution.
- *Statistical convergence error estimated, if applicable:* If the solution has a statistical character provide a detailed estimate of the convergence of the statistics as applicable.

- *Detailed description of the effect of discontinuities, singularities, chaos, and randomness, if applicable:* If applicable provide a detailed description of the impact of discontinuities, singularities, chaos or randomness on the solution, its convergence and error (including statistical error).

## **9. APPENDIX B: Examples of Required Information for the Validation Benchmark Completeness Table**

In this appendix we will give a wide variety of examples of the types of information that could be provided in each of the attributes at each of the completeness levels shown in Table 2. Since Completeness Level 0 in the table corresponds to “Little or no information” on the attribute, we will not discuss this completeness level.

### **9.1 Experimental Facility**

#### **Level 1 Examples:**

Examples of information on the functional operation of the facility and its operating procedures are:

- A general description of how the facility is operated prior to and during data acquisition.
- A general description of the automatic control system or human control of the test condition of the facility

Examples of information of the geometric and equipment features of the facility are:

- Geometric information about the balance of the facility (beyond the test section).
- For equipment, describing the type of blower or pump used to drive a water or wind tunnel.
- For a heat transfer experiment, the means of driving heat (e.g. an electrical heater with constant flux or a constant temperature heat exchanger) would suffice.

#### **Level 2 Examples:**

Examples of detailed information on the functional operation of the facility and its operating procedures are:

- A detailed description of how the facility is operated prior to data acquisition, e.g. warm-up time and how steady-state is achieved and monitored.
- A detailed description of any factors that influence the control or operation of the facility.

Examples of detailed information of geometric and equipment features of the facility are:

- Providing the fan or pump curve for the driver of the water/wind tunnel for the RPM value used in the experiment.
- For a heat transfer experiment, the means of measuring the heat delivered to the experiment should be reported, along with the measurement uncertainty.
- Detailed geometric features and their uncertainties are provided for such things as (a) drawings of the entire circuit of a closed-circuit wind tunnel/water tunnel, (b) drawings of an indraft wind tunnel where the inflow was either inside or outside the wind tunnel building, and (c) drawings of the model support structure and positioning mechanism in the test section.

Examples of information on the calibration procedures and reference standards for the facility are:

- Description of the calibration of the freestream conditions is provided, along with an uncertainty analysis.
- Description of the reference standards used for calibration of all instruments used in the facility, along with their uncertainties.

Examples of information on the calibration results and characterization of the facility are:

- Information on the non-uniformity of the freestream flow conditions in the test section, along with the estimated uncertainty.
- Mach number and turbulence level in the absence of a wind-tunnel model (these may be used by a modeler as a first step toward simulating the tunnel including the model).
- Pressure distribution over the test section of the water/wind tunnel, along with the estimated uncertainty.
- Thickness of all boundary layers both upstream and in the test section of the water/wind tunnel, along with their uncertainties.

### **Level 3 Examples:**

Examples of detailed information of the fine-scale flow features/environment inside the test section are:

- A complete map of the 3-D time-averaged flow quantities inside the water/wind tunnel, with an empty test section, along with their uncertainties.
- A complete map of the time-averaged and turbulent statistics throughout the test section of an empty water/wind tunnel test section, along with their uncertainties.

Examples of information of the fine-scale flow features or physical processes upstream and downstream of the test section are:

- Measurements of the 3-D time-averaged flow quantities upstream of the test section, along with their uncertainties.
- Measurement of the 3-D time-averaged flow quantities downstream of the test section, along with their uncertainties.

Examples of detailed information on the calibration procedures and reference standards for the facility are:

- Detailed information on the probes and equipment for calibrating a water/wind tunnel test section, plus sections upstream and downstream of the test section, and their uncertainties.
- Detailed information on the reference standards used for calibrating all instruments used in the facility, along with their uncertainties.

Examples of detailed information on the calibration results and characterization of the facility are:

- Detailed information of how freestream flow quantities are determined from calibration information and CFD simulations of the flow around the sensor, along with their uncertainties.
- Use of independent calibration instruments with alternate methods for sensing the flow quantities.

- Measurement of important geometric features of the facility, including uncertainties, including how these may vary over time or during re-assembly or maintenance of the facility.

Examples of information on the inspection, maintenance, and repairs of the facility are:

- Detailed information of how facility calibration results may vary after maintenance and repairs of the facility.
- Detailed information on how facility calibration results may vary after updates to the software for the automatic control system for facility operation.

## 9.2 Analog Instrumentation and Signal Processing

### Level 1 Examples:

Examples of information on sensors and calibration procedures are:

- The sensor type and its operation are described, and the calibration method is described, including the reference standard.
- Sources of degradation of performance of the sensor are described.

Examples of information on transducers are:

- “The diaphragm deflection due to pressure is converted to a voltage through a full strain gage bridge built into the diaphragm.”

Examples of information on signal processing are:

- Amplifier gains and filter cutoffs are provided.
- Lens focal lengths,  $f\#$ , and laser sheet thicknesses provided.

### Level 2 Examples:

Examples of detailed information on sensors and calibration procedures are:

- The model and brand of each sensor is provided along with specifications of performance from the manufacturer.
- Detailed information is provided on how the sensors are calibrated, including the traceability of calibration standards, frequency of calibration, whether the calibration was performed separately or *in situ*, along with uncertainties.

Examples of detailed information on transducers are:

- A detailed description of the strain gage transducer for a pressure sensor as a metallic or semiconductor type, along with uncertainties.
- A detailed description of a camera sensor as CCD or CMOS, and its performance.

Examples of detailed information on signal processing are:

- Amplifier gains, filter cutoffs, and examples of signals before and after signal processing.
- Filter settings, models, manufacture performance specifications, and calibration records, along with uncertainties.

Examples of assessment of instrument performance and suitability are:

- For a PIV measurement, a study of "grid convergence," or if the answer changes depending on the field of view, and the associated uncertainties.

- For a hot-wire probe, a measurement of frequency response, along with the associated uncertainties.

### **Level 3 Examples:**

Examples of use of independent sensors and calibration procedures are:

- Repeating a PIV measurement using a hot wire or LDV. The second measurement must be performed at the same or better level than the first and the comparison of the two techniques should illuminate the level of uncertainty in each measurement technique.
- Flow field instruments calibrated using different procedures or experimental facilities, along with the associated uncertainties.

Examples of use of independent or alternative signal processing procedures are:

- Description of the results of filtering in the analog signal versus the digital signal.
- Use of an independent signal processing algorithm performed on a subset of the data, and analysis of the differences in the results.

Examples detailed assessment of instrument performance and suitability are:

- CFD simulation of the instrument performance, particularly as a function of quantities that cannot be produced in a physical experiment with the available facilities, and the associated uncertainties.
- Investigation of bias uncertainties that could be correlated with signal frequencies, probe design, and analog and digital signal filtering.

## **9.3 Boundary and Initial Conditions**

### **Level 1 Examples:**

Examples of inflow quantities measured are:

- The mean velocity at the center of the facility is reported
- The mean wind velocity is measured upstream of the test region during an open-air experiment in the atmosphere.

Examples of wall quantities measured are:

- The average wall temperature in a water/wind tunnel test section is measured during the experiment.
- The average wall temperature is measured on the surface of the model during the experiment.

Examples of initial conditions measured are:

- The average flow quantities are measured at the center of the facility just before an impulsively started flow field.
- The average temperature on the surface of a model is measured as a function of time for an impulsively started flow field.

Examples of detailed model-design dimensions provided are:

- A complete set of model drawings with all relevant design dimensions is reported, including machining specifications, assembly drawings, and assembly procedures.

- If the model is in motion, or there are significant changes in geometry due to fluid or thermal loading, this information is reported.

### **Level 2 Examples:**

Examples of most inflow quantities measured are:

- The absolute pressure at various locations around the inflow perimeter of the test section are reported, including uncertainty.
- The mean and statistics of velocity are reported over the inflow plane of the test section, including the wall boundary layers.
- The mean and statistics of velocity are reported around the periphery of the test region as a function of time during an open-air experiment in the atmosphere.

Examples of most wall quantities measured are:

- For a heat transfer experiment, the temperature and/or heat flux are measured on the walls of the test sections, including uncertainty.
- The absolute pressure at various locations on the walls of the test section are measured, including uncertainty

Examples of most initial conditions measured are:

- All important flow quantities are measured before the initiation of an impulsively started flow field.
- All important wall boundary conditions are measured before the initiation of an impulsively started flow field.

Examples of detailed as-built model dimensions measured are:

- All geometric features are measured after fabrication and assembly of the model, including uncertainties.
- Measurement of all of the important geometric model features of the support/mounting structure for the model, including uncertainties.

Examples of outflow and reverse flow quantities measured are:

- Measurement of mean flow quantities at the exit of the test section for some model positions in the test section, including uncertainties.
- Measurement and characterization over space and time of reverse flow (entering) the exit plane of the test section, including uncertainties.

### **Level 3 Examples:**

Examples of fine-scale inflow quantities measured are:

- A full time-record is measured of important flow quantities at the beginning of the test section of a water/wind tunnel, including the wall boundary layers, and including uncertainties.
- For a boundary layer experiment, measurement of the acoustic frequencies across the inflow plane at the beginning of the test section.

Examples of fine-scale wall quantities measured are:

- Wall roughness, including microscopic surface profiles, in the test section and over portions upstream of the test section.

- Measurement of joints, gaps, and fasteners on the inside walls of the test section and upstream of the test sections.
- Measurement of the wall static pressure at various locations far downstream of the test section.

Examples of fine-scale outflow quantities measured are:

- Mean velocity and turbulence measurements of flow quantities at the outflow plane of the test section and one other plane near the outflow plane.
- Measurement of the multiphase flow composition at the outflow plane of the test section of a water/wind tunnel.
- Measurement of mean flow velocity far downstream of the test section of a water/wind tunnel.

Examples of fine-scale initial conditions quantities measured are:

- Measurement of all important flow quantities before the start of an impulsively started flow in a water/wind tunnel.
- Measurement of all important flow quantities before the start of an impulsively started flow in an open-air experiment in the atmosphere.

Examples of as-tested model dimensions measured are:

- Measurement of the important geometric features of the model, as they may vary due to fluid and thermal loading, during the experiment.
- Measurement of the motion of the model as a function of time during an experiment of an oscillatory flow field.

Examples of inflow and outflow quantities measured at multiple streamwise locations are:

- Mean and turbulence statistics of flow quantities far upstream of the test section, such as at the beginning of the contraction section of a water/wind tunnel, including boundary layers, and including uncertainties.
- Measurement of the mean flow quantities across a complete plane far downstream of the test section, including uncertainties.

## 9.4 Fluid and Material Properties of the Walls

### Level 1 Examples:

Examples of information provided to determine the thermodynamic state data of the fluid(s) are:

- A property value that allows other property values to be found, but insufficient data to fix the state.
- A gas temperature, which may make it possible to determine quantities such as viscosity or enthalpy, but will not allow one to fix gas density.

Examples of information on transport properties of fluid(s) are:

- Reporting kinematic and/or absolute viscosity.
- Reporting of thermal conductivity.

### Level 2 Examples:

Examples of detailed information provided to determine the thermodynamic state of fluid(s) are:

- For a simple, compressible substance, two independent, intensive properties.
- For two-phase flows, information of the thermodynamic state of each of the fluids, including the uncertainties.

Examples of detailed information provided to determine transport properties of fluid(s) are:

- Sufficient information about the fluid composition that, given the thermodynamic state, there is no ambiguity to all other properties.
- For two-phase flows, information about the transport properties of each of the fluids, including the uncertainties.

Examples volume fraction of additional phases, plus size distribution statistics, are:

- Volume fraction measurements of the additional phases near the wall.
- High-speed video of bubble formation and transport near the wall.

### **Level 3 Examples:**

Examples of all thermodynamic, transport, and optical properties of fluid(s) are provided, as well as how these are determined, are:

- A complete list of fluid property data with uncertainties.
- If an optical technique is used, the list should include optical properties.

Examples thermal, mechanical and optical properties of the wall(s) are:

- Measurement of optical properties of the wall over the range of wavelengths relevant to optical measurement techniques.
- Thermal conductivity and radiation properties for a non-isothermal experiment.

Examples of detailed description of additional phases, plus size distribution statistics, are:

- A detailed description of the additional phases that provide a number count and size distribution of each additional phase near the wall.

## **9.5 Test Conditions**

### **Level 1 Examples:**

Examples of the description provided of the method of control and record of the test conditions are:

- Closed-loop control or frequency drive of a blower.
- Closed-loop control of a heater to control the water temperature in a water tunnel.

Examples of the description provided for measuring the test conditions are:

- Experimental apparatus includes thermal control that maintains the fluid temperature within prescribed bounds.

### **Level 2 Examples:**

Examples of the detailed description provided of the method of control and record of the test conditions are:

- Measurement of the time-dependent fluid temperature at various locations in the test section of a water tunnel, as well as upstream of the test section, including uncertainties.

- Reporting the complete run-log of all test conditions where experimental measurements were made, as well as the flow conditions for each test and variation of conditions during each test.

Examples of the detailed description provided for measuring test conditions are:

- A detailed description of the sensor and transducer for measuring inflow quantities, including uncertainties.
- Measurement of the time history of the inflow quantities to the test section during the experiment.

### **Level 3 Examples:**

Examples of the detailed description of operational procedures for setting and controlling test conditions are:

- A detailed explanation of the manner in which test conditions are controlled, including variations in conditions due to the flow controller.

Examples of the detailed measurement of time and spatial variation of test conditions are:

- Measurement of the time history of the relevant test conditions during the experiment at locations upstream and downstream of the test section of the water/wind tunnel, including uncertainties.

## **9.6 Measurement of Experimental Responses**

### **Level 1 Examples:**

Examples of information on data acquisition and sampling procedures are:

- Provide information on the sequence of measurements of the various system response quantities (SRQs) obtained.
- Provide information on the duration of the acquisition time of various SRQs.
- A complete chronology of acquisition, including timing between channels.

Examples of statistical analysis of final experimental responses provided are:

- Time averaged data and statistics of SRQs are reported, including correlations.
- Statistics provided on the set of runs with the largest uncertainty statistics.

### **Level 2 Examples:**

Examples of detailed information on data acquisition and sampling procedures are:

- Detailed information on data acquisition, sampling procedures, and data filtering.
- A documented log of the procedures for each sequence of measurements made during the entire experiment.
- Number of samples, the sampling rate, the interval between samples on different instruments, including uncertainties.
- Demonstration that statistics are converged and that samples are independent.

Examples of multiple experiments conducted to determine measurement uncertainty are:

- A very detailed uncertainty analysis, with references for the methodology, to determine various sources of uncertainty.

- Determination of the contribution of uncertainty due to sources such as, human operator of a water/wind tunnel, and use of different instrumentation to set flow field, model, or other boundary conditions.

Examples of statistical analysis of intermediate experimental responses are:

- Time averaged values and statistical quantities are reported for quantities that are used to compute final SRQs.
- SRQs are reported whose analog or digital signal have been processed in an alternative method.

### **Level 3 Examples:**

Examples of the use of independent data acquisition procedures are:

- Using separate, independent instrumentation to measure the same, or similar, SRQs, including an uncertainty analysis of each instrument.
- Making independent measurements with a modified sensor, i.e., a 2-component and 3-component PIV measurement, including an uncertainty analysis of each instrument.
- Measurements acquired in different facilities and independent instrumentation, but at the same, or similar, conditions to the primary experiment, including uncertainties.

Examples of the description of sensitivity of experimental responses to control of test conditions are:

- A study reporting how much the SRQ data varies at two extremes of the control band of the experimental facility, including uncertainty.
- Conducting experimental measurements similar to the primary experiment, but at conditions where the CFD simulations have a much higher level of predictive capability, in order to uncover systematic bias errors in the experiment.

Examples of video recording of measurement procedures and data acquisition provided are:

- Complete video log of all data acquisition procedures.
- A detailed video recording of the flow field with flow visualization, with emphasis on experiments where the model position or velocity is varying.

Examples of all experimental responses reported with estimated bias and random uncertainties, including correlated uncertainties, are:

- Use of Design of Experiments methodology to identify various sources of uncertainty, including correlated bias uncertainties, such as freestream flow field non-uniformity.
- Detailed uncertainty analysis of nominally identical experiments conducted in different facilities, with emphasis on reducing or eliminating systematic differences between each facility.